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\textsuperscript{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.

\textsuperscript{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.
Table of Contents

VOLUME I - Methodology and Applications

Preface ........................................................................................................ iii
Acknowledgments ........................................................................................ v
Abstract ........................................................................................................ vi
Table of Contents ........................................................................................ ix
List of Figures ............................................................................................... xvii
List of Tables ................................................................................................. xx

1.0 Introduction .............................................................................................. 1-1
  1.1 Information Available for Flight Readiness Assessment ...................... 1-3
  1.2 Probabilistic Failure Risk Assessment .............................................. 1-7
    1.2.1 Risk Quantification Approach .............................................. 1-7
    1.2.2 The Probabilistic Failure Assessment Methodology ................. 1-8
    1.2.3 Driver Characterization .................................................... 1-10
    1.2.4 Computational Methods .................................................... 1-12
  1.3 Implementing the PFA Methodology ................................................. 1-15
  1.4 Report Organization ............................................................................ 1-17

References ..................................................................................................... 1-20
Appendix 1.A List of Acronyms ..................................................................... 1-23

2.0 Probabilistic Crack Growth Modeling .................................................. 2-1
  2.1 Introduction ......................................................................................... 2-3
  2.2 Crack Growth Methodology ............................................................... 2-5
    2.2.1 Simulation Structure .......................................................... 2-5
    2.2.2 Load Characterization and Stress Analysis ............................. 2-6
      2.2.2.1 Stress Analyses for the HEX Coll .................................. 2-7
      2.2.2.2 Stress Analyses for the EXHEX ................................... 2-9
      2.2.2.3 Stress Summation ...................................................... 2-9
    2.2.3 Stochastic Crack Growth Model ............................................. 2-11
    2.2.4 Crack Growth Calculations .................................................. 2-13
  2.3 Heat Exchanger Coil Case Study ......................................................... 2-17
    2.3.1 Component Description ....................................................... 2-17
      2.3.1.1 Crack Growth Rate Data ........................................... 2-18
    2.3.2 Driver Description ............................................................. 2-20
    2.3.3 Analysis .............................................................................. 2-22
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 - BLDLCD ........................................ 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 ........................................... 4-7
  4.2.4 Mean Stress Effects ............................................. 4-10
  4.2.5 Modeling Multiple Critical Locations ......................... 4-10
  4.2.6 Probabilistic Failure Model Implementation ................. 4-10

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

Appendix 4.A List of Symbols ........................................ 4-15

VOLUME II – Software Documentation

5.0 Analysis Software ................................................ 5-1
  5.1 Crack Growth Analysis Software ................................ 5-3
    5.1.1 Introduction ................................................... 5-3
    5.1.2 PROCRK Program .............................................. 5-3
      5.1.2.1 Main Routine ............................................ 5-3
      5.1.2.2 SETDEF Routine ......................................... 5-5
      5.1.2.3 INPUT Routine ........................................... 5-5
      5.1.2.4 GRODAT Routine ......................................... 5-5
      5.1.2.5 HYPDRW Routine ......................................... 5-7
      5.1.2.6 PARDRW Routine ......................................... 5-7
      5.1.2.7 LIFCAL Routine .......................................... 5-7
      5.1.2.8 STRAN1 and STRAN2 Routines ........................... 5-7
      5.1.2.9 CYCOUN Routine ......................................... 5-11
      5.1.2.10 BLKGRO Routine ........................................ 5-18
      5.1.2.11 STRIF1 and STRIF2 Routines ........................... 5-18

  5.2 Low Cycle Fatigue Analysis Software .......................... 5-21
    5.2.1 Introduction .................................................. 5-21
    5.2.2 BDLDCF Program ............................................... 5-21
      5.2.2.1 Main Routine ............................................. 5-21
      5.2.2.2 BDLIF Routine ........................................... 5-26
      5.2.2.3 RAINF3 Routine ......................................... 5-26
    5.2.3 BDLDCF Program, Nonparametric Materials Model .......... 5-26
      5.2.3.1 Main Routine ............................................. 5-35
      5.2.3.2 INFAGG Routine ......................................... 5-38
      5.2.3.3 SW2SU2 Routine ......................................... 5-45
      5.2.3.4 EXPB Routine ............................................ 5-50
      5.2.3.5 PEB Routine ............................................. 5-52
      5.2.3.6 PICRES Routine ......................................... 5-52
      5.2.3.7 MREGR Routine ......................................... 5-52
5.2.3.8 WORSTN Routine ........................................ 5-57

5.3 High Cycle Fatigue Analysis Software .................................. 5-59
5.3.1 Introduction .................................................................. 5-59
5.3.2 BLDHCF Program ...................................................... 5-59
5.3.2.1 Main Routine ...................................................... 5-59
5.3.2.2 DRVRIN Routine ................................................... 5-63
5.3.2.3 SELECT Routine .................................................. 5-63
5.3.2.4 BLDHLF Routine .................................................. 5-64

References ................................................................. 5-64

Appendix 5.A Program Flowchart Symbols .................................. 5-67

6.0 Software User’s Documentation ......................................... 6-1

6.1 Crack Growth Analysis User's Guide .................................... 6-3
6.1.1 PROCRK Program ...................................................... 6-3
6.1.2 How to Use Program PROCRK ....................................... 6-3
6.1.3 Description of Input Data Files ....................................... 6-4
6.1.3.1 Input File CRKDAT .............................................. 6-4
    Analysis Parameters Block .............................................. 6-11
    Driver Information Block .............................................. 6-13
    Load and Stress Block .................................................. 6-23
    Materials Information Block ......................................... 6-28
6.1.3.2 Reference Time History Files ................................... 6-30
6.1.4 Options and Capabilities ............................................. 6-30
6.1.5 Code Execution Example ............................................. 6-31
    Input File - CRKDAT .................................................... 6-34
    Input File - NBM3 ....................................................... 6-36
    Input File - SIN1 ........................................................ 6-36
    Input File - AERO1 ....................................................... 6-36
    Output File - CRKRES .................................................. 6-37
    Output File - IOUTPR .................................................. 6-42
    Output File - LOWLIF .................................................. 6-43
6.1.6 Error Messages and Possible Remedies ............................ 6-43
6.1.7 Summary of Input/Output Files ..................................... 6-45
    Input Files .................................................................. 6-45
    Output Files .................................................................. 6-45

6.2 Low Cycle Fatigue Analysis User's Guide .............................. 6-47
6.2.1 BLDLCF Program ...................................................... 6-47
6.2.2 How to Use Program BLDLCF ....................................... 6-47
6.2.3 Description of Input Data Files ....................................... 6-48
6.2.3.1 Input File BLDLCD .............................................. 6-48
    Analysis Parameters Block .............................................. 6-53
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.3.2</td>
<td>Input File RELATD</td>
<td>6-67</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Options and Capabilities</td>
<td>6-68</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Code Execution Example</td>
<td>6-69</td>
</tr>
<tr>
<td></td>
<td>Input File - BDLLCD</td>
<td>6-71</td>
</tr>
<tr>
<td></td>
<td>Input File - RELATO</td>
<td>6-72</td>
</tr>
<tr>
<td></td>
<td>Output File - BDLCO</td>
<td>6-72</td>
</tr>
<tr>
<td></td>
<td>Output File - RELATO</td>
<td>6-76</td>
</tr>
<tr>
<td></td>
<td>Output File - DUMP</td>
<td>6-76</td>
</tr>
<tr>
<td></td>
<td>Output File - IOUTPR</td>
<td>6-77</td>
</tr>
<tr>
<td></td>
<td>Output File - LOWLIF</td>
<td>6-77</td>
</tr>
<tr>
<td>6.2.6</td>
<td>Error Messages and Possible Remedies</td>
<td>6-81</td>
</tr>
<tr>
<td>6.2.7</td>
<td>Summary of Input/Output Files</td>
<td>6-86</td>
</tr>
<tr>
<td></td>
<td>Input Files</td>
<td>6-86</td>
</tr>
<tr>
<td></td>
<td>Output Files</td>
<td>6-86</td>
</tr>
</tbody>
</table>

7.0 Structure and Listing of Programs ........................................ 7-1

7.1 Crack Growth Analysis Software PROCRK ................................... 7-3
| 7.1.1 | Program Tree Structure | 7-3 |
| 7.1.2 | List of Subprograms   | 7-3 |
| 7.1.3 | Description of Variables | 7-7 |
| 7.1.4 | Program PROCRK Listing | 7-23 |
|        | Program PROCRK Listing Temporal Order | 7-24 |

7.2 Low Cycle Fatigue Failure Program BDLLCF .................................. 7-77
| 7.2.1 | Program Tree Structure | 7-77 |
| 7.2.2 | List of Subprograms   | 7-77 |
| 7.2.3 | Description of Variables | 7-85 |
| 7.2.4 | Program BDLLCF Listing | 7-97 |
|        | Program BDLLCF Listing Temporal Order, Uniform Distribution | 7-98 |
|        | Program BDLLCF Listing Temporal Order, Truncated Normal Distribution | 7-100 |
| 7.2.5 | Program BDLLCF V3.4B1.3 Listing | 7-176 |
|        | Program BDLLCF V3.4B1.3 Listing Temporal Order | 7-177 |

7.3 High Cycle Fatigue Failure Program BLDHCF .................................. 7-255
| 7.3.1 | Program Tree Structure | 7-255 |
| 7.3.2 | List of Subprograms   | 7-255 |
| 7.3.3 | Description of Variables | 7-261 |
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

Figure 1.2-1  Information Sources for Failure Risk Assessment .................. 1-7
Figure 1.2-2  The Probabilistic Failure Assessment Methodology .................. 1-8
Figure 1.2-3  The Probabilistic Failure Modeling Procedure ....................... 1-10
Figure 1.3-1  Integration of PFA into the Design and Development Process .... 1-15
Figure 2.1-1  Crack Growth Failure Modeling Approach ............................ 2-3
Figure 2.2-1  Crack Growth Failure Simulation Structure .......................... 2-6
Figure 2.2-2  Geometry of Duct .................................................. 2-8
Figure 2.2-3  Description of the Stochastic Crack Growth Equation in Log-Log Space 2-12
Figure 2.2-4  Flowchart for the Crack Growth Calculations ....................... 2-16
Figure 2.3-1  HPOTP Heat Exchanger ............................................. 2-17
Figure 2.3-2  Detail of the HPOTP Heat Exchanger Coil Near Weld 3 ............ 2-18
Figure 2.3-3  Schematic of Approximating the SIF Solution for Duct with a Finite Width Plate Solution ........................................... 2-19
Figure 2.3-4  The Mean Crack Growth Rate Curves for Welded 316L Data ....... 2-19
Figure 2.3-5  Failure Life Distributions and Driver Sensitivities for HEX Coll 2-24
Figure 2.3-6  Impact of Initial Crack Size a_i on Failure Life Distributions for HEX Coll 2-24
Figure 2.3-7  Effect of Initial Crack Size and Crack Growth Threshold on B.1 Life for HEX Coll 2-25
Figure 2.4-1  Proposed External Heat Exchanger Block II Design ................ 2-28
Figure 2.4-2  Arrangement of the Channels in the EXHEX ......................... 2-28
Figure 2.4-3  Crack Configuration Used for EXHEX Channel ...................... 2-29
Figure 2.4-4  Crack Growth Data for C10100 Copper and Mean Curve with λ_KIN = 0 2-29
Figure 2.4-5  EXHEX Failure Life Distribution and Driver Sensitivities ....... 2-32
Figure 2.5-1  Overall Procedure for Crack Growth Failure Mode .................. 2-34
Figure 2.5-2  Steps of the Probability of Failure Curve Parameter Estimation 2-48
Figure 3.1-1  Low Cycle Fatigue Failure Modeling Approach ....................... 3-3
Figure 3.1-2  Calculation Procedure Used in the Low Cycle Fatigue Model .... 3-4
Figure 3.2-1  Axial Cross Section of the ATD-HPFTP Turbine Showing the Monolithic Disk and Both Stages of Blades ............................... 3-8
Figure 3.2-2  ATD-HPFTP First Stage Turbine Blade ................................ 3-8
Figure 3.2-3  Illustration of the Strain-Time History for the ATD Blade ......... 3-9
Figure 3.2-4  Schematic of Temperature Profile During Shutdown ................. 3-10
Figure 3.2-5  The Simulation Structure for the Parametric Representation of Parameter Uncertainty ...................................................... 3-17
Figure 3.2-6  The Simulation Structure for the Bootstrapping Representation of Parameter Uncertainty .................................................. 3-18
Figure 3.2-7  Plot of Eight Specimen Failure Points .................................. 3-19
Figure 3.2-8  Structure of the Probabilistic Failure Model ........................... 3-23
Figure 3.2-9  Structure of the Turbine Blade LCF Failure Simulation .................................. 3-24  
Figure 3.3-1  PWA 1480 Fatigue Data .............................................................................. 3-30  
Figure 3.3-2  Impact of Materials Variation on Failure Life Distribution ...................... 3-31  
Figure 3.3-3  Driver Sensitivities for Weibull Materials Variation .............................. 3-33  
Figure 3.3-4  Driver Sensitivities for Lognormal Materials Variation ......................... 3-33  
Figure 3.3-5  Impact of Materials Variation Parameterization on Failure Life Distribution 3-34  
Figure 3.4-1  Steps of the Probability of Failure Curve Parameter Estimation ............ 3-39  
Figure 4.1-1  High Cycle Fatigue Failure Modeling Approach ........................................ 4-3  
Figure 4.2-1  Calculation Procedure for the Turbine Blade HCF Model ..................... 4-6  
Figure 4.2-2  Structure of the Driver Transformation for the Turbine Blade HCF Model 4-8  
Figure 4.2-3  Structure of the Turbine Blade HCF Probabilistic Failure Model .......... 4-11  
Figure 4.2-4  Turbine Blade HCF Failure Simulation Used in the Probabilistic Failure Model ................................................................. 4-13  
Figure 5.1-1  Main Flowchart for Crack Growth Analysis Program PROCRK .............. 5-4  
Figure 5.1-2  Flowchart for Subprogram GRODAT .......................................................... 5-6  
Figure 5.1-3  Flowchart for Subprogram LIFCAL ........................................................... 5-8  
Figure 5.1-4  Flowchart for Subprogram STRAN1 ......................................................... 5-9  
Figure 5.1-5  Flowchart for Subprogram STRAN2 .......................................................... 5-12  
Figure 5.1-6  Flowchart for Subprogram CYCOUN ......................................................... 5-13  
Figure 5.1-7  Flowchart for Subprogram BLKGRO .......................................................... 5-19  
Figure 5.2-1  Structure of the Turbine Blade LCF Probabilistic Failure Model ............ 5-22  
Figure 5.2-2  Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF .... 5-23  
Figure 5.2-3  Flowchart for Subprogram BLDLIF ............................................................ 5-27  
Figure 5.2-4  Flowchart for Subprogram RAINF3 ........................................................... 5-28  
Figure 5.2-5  Structure of the Turbine Blade LCF Probabilistic Failure Model Using the Nonparametric Materials Model .............................. 5-34  
Figure 5.2-6  Main Flowchart for the ATD Blade LCF Analysis Program BLDLCF Using the Nonparametric Materials Model ................................................................. 5-36  
Figure 5.2-7  Flowchart for Subprogram INFAGG ........................................................... 5-39  
Figure 5.2-8  Flowchart for Subprogram SW2SU2 .......................................................... 5-46  
Figure 5.2-9  Flowchart for Subprogram EXPB ............................................................... 5-51  
Figure 5.2-10 Flowchart for Subprogram PEB ............................................................... 5-53  
Figure 5.2-11 Flowchart for Subprogram MREGR .......................................................... 5-55  
Figure 5.3-1  Structure of the Turbine Blade HCF Probabilistic Failure Model .......... 5-60  
Figure 5.3-2  Main Flowchart for the ATD Blade HCF Analysis Program BLDHCF .... 5-61  
Figure 5.3-3  Structure of the Driver Transformation for the Turbine Blade HCF Model 5-65  
Figure 5.3-4  Structure of the Failure Life Calculation for Turbine Blade HCF ............. 5-66  
Figure 5.A-1 Program Flowchart Symbols ................................................................. 5-67  
Figure 6.1-1  Format for File CRKDAT for HEX Coil Problem ..................................... 6-5  
Figure 6.1-2  Format for File CRKDAT for EXHEX Problem ........................................ 6-9  
Figure 6.1-3  Data Blocks for Input File CRKDAT ............................................................ 6-11  

xviii
Figure 6.1-4  Detail of the HPOTP Heat Exchanger Coll Small Tube Outlet Near Weld 3  . . . 6-32
Figure 6.2-1  Format for File BDLLCD  .................................................................. 6-49
Figure 6.2-2  Format for File RELATD ................................................................. 6-52
Figure 6.2-3  Data Blocks for Input File ............................................................... 6-52
Figure 7.1-1  Tree Structure for Program PROCRK ............................................. 7-4
Figure 7.2-1  Tree Structure for Program BDLDCF for the Uniform Variation in
Materials Shape Parameter $m$ ........................................................................... 7-78
Figure 7.2-2  Tree Structure for Program BDLDCF for the Truncated Normal Variation
in Materials Shape Parameter $m$ ........................................................................ 7-79
Figure 7.2-3  Tree Structure for Program BDLDCF V3.4B1.3 for the Bootstrapping
of the Materials Shape Parameter $m$ ................................................................. 7-80
Figure 7.3-1  Tree Structure for Program BLDHCF for the Uniform Variation in
Materials Shape Parameter $m$ ........................................................................... 7-256
Figure 7.3-2  Tree Structure for Program BLDHCF for the Truncated Normal Variation
in Materials Shape Parameter $m$ ........................................................................ 7-257
List of Tables

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

Figure 5.1-1 Main Flowchart for Crack Growth Analysis Program
PROCRK
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. \( AK \) 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].
Figure 5.1-2  Flowchart for Subprogram GRODAT
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 \( \frac{da}{dB} \)
(See Section 5.1.2.10)

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

RETURN

Figure 5.1-3  Flowchart for Subprogram LIFCAL
Figure 5.1-4  Flowchart for Subprogram STRAN1
ASSIGN STATIC STRESSES

\[ \sigma(t) = \sigma_{ST} \]

DO I = 1

TO LOAD BY 1

LOOP OVER ALL LOAD

SOURCES

SCALE TIME VARYING STRESSES

\[ \sigma(t) = \sigma(t) + \lambda \sum_{i} \sigma_{i}(t) \]  \hspace{1cm} (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 magni-
itudes 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 calcu-
lated 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
\[ \sigma(t) = \lambda_{ST} \dot{\lambda}_{ST} \sigma_{ST} \]

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

SCALE TIME VARYING STRESSES
\[ \sigma(t) = \sigma(t) + \lambda_{D} \dot{\lambda}_{DYN} \sqrt{D(t)} \sigma(t) \] (Eq. 2-5)

RETURN

Figure 5.1-5  Flowchart for Subprogram STRAN2
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

\[ E_k = \sigma_j \]

TRUE

K < 3

FALSE

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

FALSE

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

TRUE

\begin{align*}
I &= I + 1 \\
\sigma_{\text{eff}1,i} &= \max [E_{k-1}, E_{k-2}] \\
\sigma_{\text{eff}2,i} &= \min [E_{k-1}, E_{k-2}] \\
\end{align*}

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 (\(\text{IRET} = 1\)), 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.
PROB = STRIF1
CALCULATE SIF COEFFICIENTS FOR HEX COIL
(See Section 5.1.2.11)

KPROB = 2

STRIF2
CALCULATE SIF COEFFICIENTS FOR EXHEX
(See Section 5.1.2.11)

DO I ← 1
TO 100 BY 1
LOOP FOR EVERY STRESS LEVEL BIN

DO IDIR ← 1
TO NDIR BY 1
LOOP FOR "a" AND "c" CRACK DIRECTION

IRET = 1
FALSE

CALCULATE $K_{\text{max,eff}}$ AND $K_{\text{min,eff}}$ BASED ON PLASTIC ZONE SIZE DUE TO OVERLOAD AND CURRENT LOAD AS PER WILLENBORG MODEL

$\Delta K_{\text{eff}} = K_{\text{max,eff}} - K_{\text{min,eff}}$
$(\text{Eq } 2-16)$

$R_{\text{eff}} = \frac{K_{\text{min,eff}}}{K_{\text{max,eff}}}$

Figure 5.1-7 Flowchart for Subprogram BLKGRO
CALCULATE BLOCK GROWTH BY SUMMING CYCLE GROWTH RATE

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

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.
• PROBABILISTIC CHARACTERIZATION OF DRIVER UNCERTAINTIES
• NOMINAL STRAINS AND ENVIRONMENT
• PARAMETRIC SENSITIVITIES
• MATERIALS DATA
• REFERENCE STRAIN-TIME HISTORY
• NUMBER OF CRITICAL LOCATIONS

RANDOM SELECTION OF A PROBABILITY DISTRIBUTION FOR $h_{gas}$, $T_{gas}$, $\lambda_G$, $m$

MATERIALS MODEL INFORMATION AGGREGATION

RANDOM SELECTION OF VALUES FOR $\lambda_{MA}$, $\lambda_{TH}$, $\lambda_B$

SELECTION OF MATERIAL RANDOM VARIABLE, $\varphi$

LOOP 50 TIMES

MIN $\varphi$

TURBINE BLADE LCF FAILURE SIMULATION

PREDICTED FAILURE TIME

LOOP $n$ TIMES

ACCUMULATE SMALLEST $n \times N / 100$ FAILURE TIMES

SELECTED MEDIAN S/N CURVE

Figure 5.2-1 Structure of the Turbine Blade LCF Probabilistic Failure Model
START
READ AND ECHO INPUT DATA

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

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

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 NLIIFE BY 1 INNER LOOP

BETAGN, NORMGN, PRYRV
DRIVER SELECTION FOR
$h_{gas}$, $T_{gas}$, $m$, $\lambda_G$, $\omega(t_5)$, $\varepsilon_A$, $T_s$,
$e_D$, $e_B$, $A_p$, $\lambda_o$, $\lambda_{dam}$, $\lambda_{TMF}$
(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 BDLCLCF

5 - 23
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 \(\lambda_{\text{MA}}\) 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 \(h_{\text{gas}}, T_{\text{gas}}, m, \lambda_G, \omega(t_s), e_A, T_s, e_D, \varepsilon_B, \lambda_P, \lambda_d, \lambda_{\text{dam}},\) and \(\lambda_{\text{TF}}\) 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

\[^3\text{Hyperparameters are discussed in [1], Section 2.1.1.}\]
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 $\varepsilon_B, \lambda_p, \omega(t_5), \lambda_{\alpha}$, $h_{gas}, T_{gas}, T_s, e_A, \lambda_G, m, e_D$

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

**DRIVER TRANSFORMATION**

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

**STRAIN-TIME HISTORY**

- RAINFLOW CYCLE COUNT
- ACCUMULATE DAMAGE
- CALCULATE LIFE

**SELECTED S/N CURVE**

$K_j, m_j, \varphi$

SELECTED VALUES FOR $\lambda_{dam}, \lambda_{TMF}$

**PREDICTED FAILURE TIME**

---

Figure 5.2-3  Flowchart for Subprogram BDLIF
Record the largest $\sigma_{\text{eff}}$ and its location

$\sigma_{\text{effmax}} = \sigma_{\text{effi}}$

$\text{JMAX} = 1$

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\(_{\text{NEWTOT}} = M + 1 \)

DO \( I \leftarrow 1 \)
TO NEWTOT BY 1

SET UP THE PEAK-TROUGH ARRAY
\( K = \) INDEX\(_{i} \)
\( \sigma_i = \sigma_{P_i} \)

INITIALIZE COUNTERS
\( I = 0, \ J = 0, \ K = 0 \)

INCREMENT COUNTERS
\( J = J + 1 \)
\( K = K + 1 \)

\( J > \) NEWTOT
TRUE

FALSE

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
\[ I = I + 1 \]
\[ \sigma_{\text{eff}1_i} = \max [E_{k-1}, E_{k-2}] \]
\[ \sigma_{\text{eff}2_i} = \min [E_{k-1}, E_{k-2}] \]

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

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

N = 1

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)

DO I = 1

TO NUMBER OF CYCLES N BY 1

\[ \text{Figure 5.2-4  Flowchart for Subprogram RAINF3 (Cont'd)} \]

5 - 32
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_i = \frac{1}{LIFE_i}

DO I ← 1

TO NUMBER OF CYCLES N BY 1

SUM THE DAMAGE FRACTIONS

SUMDAM = SUMDAM + INVLIF_i

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

- Probabilistic characterization of driver uncertainties
- Nominal strains and environment
- Parametric sensitivities
- Materials data
- Reference strain-time history
- Number of critical locations
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>BDLCO</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 BDLCO. 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 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 $h_{gas}$, $T_{gas}$, $m$, 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. 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 $h_{gas}$, $T_{gas}$, $m$, $\lambda_G$, $\omega(t_s)$, $e_A$, $T_s$, $e_D$, $\lambda_p$, $\lambda_\alpha$, $\lambda_{dam}$, 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.
START
READ AND ECHO
INPUT DATA

INFAGG
PERFORM MATERIALS
INFORMATION AGGREGATION
(See Section 5.2.3.2)

INITIALIZE ARRAYS
AND SET DEFAULTS

DO J ← 1
TO NHYPER BY 1
OUTER LOOP

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

PEB
BOOTSTRAP RESIDUALS
GENERATE PSEUDO S/N DATA
REGRESS TO OBTAIN
STRUCTURAL PARAMETERS
(See Section 5.2.3.5)

OPTIONAL (See Section 5.2.3.5)
IF MATERIALS,
PROCESS
VARIATION
WAS USED

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

DO I ← 1
TO NLIFE BY 1
INNER LOOP

BETAGN, NORMGN, PRYRV
DRIVER SELECTION FOR
$h_{\text{gas}}, T_{\text{gas}}, m, \lambda_G, \omega(t_5), e_A, T_s,$
$e_D, e_B, \lambda_p, \lambda_c, \lambda_{\text{dam}}, \lambda_{TMF}$
(See [1], Sections 4.4.3, 4.4.5 & 7.6.6)

Figure 5.2-6 Main Flowchart for the ATD Blade LCF Analysis
Program BLDLCF Using the Nonparametric
Materials Model
WORSTN
SELECT \( \min \{ \varepsilon_i \} \) 
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: 
BLDLCO 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 BDLDCF 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 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 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:

<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

\[ \text{---} \]

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)

EXPECTD 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)
MUSIG

PERFORM BAYESIAN ANALYSIS TO OBTAIN $m_e$ & $\sigma_e^2$ FOR EACH REGION
(See [1], Section 4.1.3.13)

PROCESS VARIATION ?

YES

GTPVAR

CALCULATE $\sigma^2$
(See [1], Section 4.1.3.7)

NORRNG

COMPUTE POSTERIOR CREDIBILITY RANGE
(See [1], Section 4.1.3.14)

ADDRNG

ADD INFORMATION ON REGIONS WITHOUT S/N DATA
(See [1], Section 4.1.3.15)

CONCAY

IMPOSES S/N CURVE CONCAVITY CONSTRAINTS
(See [1], Section 4.1.3.10)

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 FNDRNG 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, FNDRNG, 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 \( \sigma_0^2 \) 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 $\ln S$ and $y$ is equal to $\ln N$. 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 $L = 1, \ldots, R$, 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, $j = 0, \ldots, P$, and the inner DO loop is for each data point in each data set, $k = 1, \ldots, N_j$. 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:

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

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

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

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

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 \( L \)

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

\[
N S_{xy} \geq 0
\]

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 \( \hat{K} \)

\[
\hat{m} = -d
\]

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

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 \( L \)

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

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

Figure 5.2-8  Flowchart for Subprogram SW2SU2 (Cont'd)
CALCULATE RESIDUAL VARIANCES \( S_W^2 \) and \( S_U^2 \)

CALCULATE RESIDUALS

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

RETURN

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

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

The second set of DO loops calculates the residuals \( e \) and the residual variances \( S_w^2 \) and \( S_u^2 \) for each life region given by

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

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

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

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. 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.

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

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

USE $S_0$ ?

NO

YES

KOMO

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

WRITE TO DUMP
$\beta_o, k, [K_i, m_i], S_{i,j+1}$

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 \( \beta_0 \) 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 \( \beta_0 \). 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 \( \sqrt{\frac{N_0}{N_0 - 2}} \) 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.
START

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)

RESIDUALS S VALUES

LIFE REGION BOUNDARIES SPECIFIC S/N DATA

Figure 5.2-10 Flowchart for Subprogram PEB
FINDK

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

FINDSB

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

\[ \beta_o, \{K_j, m_j\}, S_{i,j+1}^* \]

**RETURN**

*Figure 5.2-10  Flowchart for Subprogram PEB (Cont’d)*
![Flowchart for Subprogram MREGR]

Figure 5.2-11 Flowchart for Subprogram MREGR

\[
\begin{align*}
NS_x^2 &= \sum_{k=1}^{N_x} (x_k - \bar{x})^2 \\
NS_{xy} &= \sum_{k=1}^{N_y} (x_k - \bar{x})(y_k - \bar{y})
\end{align*}
\]
CALCULATE \( d \)
(Equation 2-20 of [1])
\[ d = \frac{S_{xy}}{S_x^2} \]
CALCULATE \( \hat{m} \)
\[ \hat{m} = -d \]

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

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

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

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,

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

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

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

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

\[\text{---}
\]

\[13 \quad R \text{ is currently equal to 1 for the bootstrapping option, but the region DO loop has been included in anticipation of future work on the bootstrap option.}\]
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>BLDHC0</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 BLDHC0. 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
Figure 5.3-2  Main Flowchart for the ATD Blade HCF Analysis Program BLDHCF
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 calcula-
tions 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 φ is found from the
minimum of 54 draws of a Weibull distribution. Calls to WEIBGN provide the 54
values of φ. 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 \( \sigma_{BR} \) calculation is performed, Equation 4-1. The blade undamped vibratory stress \( \sigma_{UD} \) is calculated based on the empirical characterization as a function of \( \sigma_{BR} \). 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


SELECTED VALUE FOR ROTOR SPEED $\omega$

MODEL ACCURACY

PERFORMANCE CHARACTERIZATION

$\dot{m}, \Delta h, \omega$

SELECTED VALUES FOR $R_{\text{root}}, R_{\text{avg}}, C, l_{\text{min}}$

MODEL ACCURACY

BLADE ROOT MEAN STRESS MODEL

$\sigma_{\text{BR}} = f(\dot{m}, \Delta h, \omega, R_{\text{root}}, R_{\text{avg}}, C, l_{\text{min}})$

RATIO OF DAMPED VIBRATORY STRESS TO UNDAMPED VIBRATORY STRESS

$\sigma_{D}/\sigma_{UD}$

STOCHASTIC MODEL FOR UNDAMPED BLADE ROOT VIBRATORY STRESS

$\sigma_{UD} = f(\sigma_{BR})$

$\sigma_{\text{BR}}, \sigma_{UD}$

$\sigma_{\text{MEAN}} = \sigma_{BR}, \sigma_{\text{ALT}} = \sigma_{UD}(\sigma_{D}/\sigma_{UD})$

$\sigma_{\text{MAX}} = \sigma_{\text{MEAN}} + \sigma_{\text{ALT}}$

$\sigma_{\text{MIN}} = \sigma_{\text{MEAN}} - \sigma_{\text{ALT}}$

$R = \sigma_{\text{MIN}}/\sigma_{\text{MAX}}$

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

Figure 5.3-3 Structure of the Driver Transformation for the Turbine Blade HCF Model
SELECTED VALUES FOR $\omega, R_{\text{root}}, R_{\text{avg}}, C, r_d$

PARAMETRIC SENSITIVITIES

DRIVER TRANSFORMATION

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

SELECTED VALUES FOR $\lambda_B, \lambda_D$

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

$N_f = K_f m_f \sigma_{\text{EQ}} m_f \varphi m_f$

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

5 - 66
Appendix 5.A

Program Flowchart Symbols

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

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 \( da/dN \) 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 _ 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 | Narrow-band random load scale factor |
| 2.00 | 2.00 | 0.20 | 1.00 | Sinusoidal load scale factor |
| 486. | 666. | 29. | 56.5 | Normal distribution on inner wall temperature information |
| 799. | 908. | 49.5 | 48. | Normal distribution on outer wall temperature information |
| 3808. | 4177. | 69. | 69. | Normal distribution on internal pressure information |

| 0.80 | 1.20 | Uniform distribution bounds for weld offset accuracy factor |
| 0.50 | 1.50 | Uniform distribution bounds for aerodynamic load scale factor |
| 0.80 | 1.20 | Uniform distribution bounds for aerostatic load scale factor |
| 0.90 | 1.10 | Uniform distribution bounds for aerolods 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
Number of dynamic loads

Aerostatic load: P, Mx, My, Mz, Vy, Vz

| 0.00 | 0.00 | -0.07214 | 0.00 | 0.00 | 0.00 |

Dynamic loads: file name, load type, P, Mx, My, Mz, Vy, Vz

| 'NBM3' | 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 | 0.0 |

10 points of the piecewise linear $F_k$ vs. $R/t$ curve

| 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 |

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem (Cont’d)
Number of segments in $\sigma$ vs. $\epsilon$ curve

<table>
<thead>
<tr>
<th>$\sigma_1$</th>
<th>$\epsilon_1$</th>
<th>21.95</th>
<th>0.001</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_2$</td>
<td>$\epsilon_2$</td>
<td>55.77</td>
<td>0.002</td>
</tr>
<tr>
<td>$\sigma_3$</td>
<td>$\epsilon_3$</td>
<td>144.85</td>
<td>0.005</td>
</tr>
<tr>
<td>$\sigma_4$</td>
<td>$\epsilon_4$</td>
<td>322.73</td>
<td>0.010</td>
</tr>
<tr>
<td>$\sigma_5$</td>
<td>$\epsilon_5$</td>
<td>1945.90</td>
<td>0.050</td>
</tr>
<tr>
<td>$\sigma_6$</td>
<td>$\epsilon_6$</td>
<td>50688.0</td>
<td>0.660</td>
</tr>
</tbody>
</table>

Description of material data

'M400 F 316L WELDED FROM RkD'

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

Threshold stress intensity factor range model parameters: $\Delta K_{TH0}$, $C_0$, $d$

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

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

<table>
<thead>
<tr>
<th>$da/dN$</th>
<th>$\Delta K$</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.183E-10</td>
<td>2.56</td>
</tr>
<tr>
<td>1.138E-8</td>
<td>2.69</td>
</tr>
<tr>
<td>3.362E-8</td>
<td>2.82</td>
</tr>
<tr>
<td>8.473E-8</td>
<td>3.00</td>
</tr>
<tr>
<td>4.408E-8</td>
<td>3.33</td>
</tr>
<tr>
<td>5.838E-8</td>
<td>3.53</td>
</tr>
<tr>
<td>5.679E-8</td>
<td>3.74</td>
</tr>
<tr>
<td>7.220E-8</td>
<td>3.95</td>
</tr>
<tr>
<td>8.202E-8</td>
<td>4.18</td>
</tr>
<tr>
<td>7.440E-8</td>
<td>4.42</td>
</tr>
<tr>
<td>9.028E-8</td>
<td>4.67</td>
</tr>
<tr>
<td>1.133E-7</td>
<td>4.94</td>
</tr>
<tr>
<td>1.533E-7</td>
<td>5.22</td>
</tr>
<tr>
<td>1.629E-7</td>
<td>5.51</td>
</tr>
<tr>
<td>1.727E-7</td>
<td>5.81</td>
</tr>
<tr>
<td>2.321E-7</td>
<td>5.99</td>
</tr>
</tbody>
</table>

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

\[ \begin{array}{|c|c|} \hline 
18 & 0.70 \\
\hline 
\end{array} \]

\( \frac{da}{dN} \) vs. \( \Delta K \) data for division 2

\[
\begin{array}{|c|c|}
\hline
4.661E-9 & 3.58 \\
2.469E-8 & 3.80 \\
1.387E-7 & 4.49 \\
1.162E-7 & 4.88 \\
1.631E-7 & 5.28 \\
1.539E-7 & 5.74 \\
1.562E-7 & 6.24 \\
1.839E-7 & 6.77 \\
2.089E-7 & 7.35 \\
3.497E-7 & 7.99 \\
2.949E-7 & 9.37 \\
3.848E-7 & 10.15 \\
6.968E-7 & 11.91 \\
8.980E-7 & 12.87 \\
1.111E-6 & 13.89 \\
1.380E-6 & 15.00 \\
2.790E-6 & 17.49 \\
3.901E-6 & 18.17 \\
\hline
\end{array}
\]

Figure 6.1-1 Format for File CRKDAT for HEX Coil Problem (Cont’d)
2 Problem type
2 Crack growth model type
675 Random number seed
0 Output dump controller
1 Inner loop size
10000 Outer loop size
1 Growth retardation switch (on)
2 Neuber's rule switch (off)
5 Number of B-lives

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 |

| 2.00 | 2.00 | 0.15 | 1.00 | Narrow-band random load scale factor |
| 2.00 | 2.00 | 0.20 | 1.00 | Sinusoidal load scale factor |

| 0.90 | 1.10 | Uniform distribution bounds for static stress analysis accuracy factor |
| 0.80 | 1.20 | Uniform distribution bounds for dynamic stress analysis accuracy factor |
| 0.00 | 0.00 | Uniform distribution bounds for threshold stress intensity factor uncertainty |
| 1.00 | 1.00 | Uniform distribution bounds for critical stress intensity factor uncertainty |
| 0.90 | 1.10 | Uniform distribution bounds for stress intensity factor calculation accuracy |
| -1.38629 | 0.95166 | Uniform distribution bounds for crack growth calculation accuracy factor |
| -1.50 | -2.50 | Uniform distribution bounds for Forman equation $m$ variation |

2 Number of dynamic load sources

Static stresses: $\sigma_x, \sigma_y, \sigma_z, \sigma_{xy}, \sigma_{xz}, \sigma_{yz}$

| 0.00 | 0.00 | 552.34 | 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: $\Delta K_{TH0}$, $C_\alpha$, $d$

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
PROCRK 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 PROCRK'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
\( W_{OFF} \) 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 \( W_{OFF} \). 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 $\theta$. The third line is the decimal equivalent percentage weight for the first Beta distribution, and it must be between 0.00 and 1.00.

- **WOFFA**: lower bound of the first Beta distribution on $W_{OFF}$
- **WOFFB**: upper bound of the first Beta distribution on $W_{OFF}$
- **WOFFR1**: Uniform distribution lower bound of parameter $\rho$ in the first Beta distribution of $W_{OFF}$
- **WOFFR2**: Uniform distribution upper bound of parameter $\rho$ in the first Beta distribution of $W_{OFF}$
- **WOFFT1**: Uniform distribution lower bound of parameter $\theta$ in the first Beta distribution of $W_{OFF}$
- **WOFFT2**: Uniform distribution upper bound of parameter $\theta$ in the first Beta distribution of $W_{OFF}$
- **WOFFC**: lower bound of the second Beta distribution on $W_{OFF}$
- **WOFFD**: upper bound of the second Beta distribution on $W_{OFF}$
- **WOFFR3**: Uniform distribution lower bound of parameter $\rho$ in the second Beta distribution of $W_{OFF}$
- **WOFFR4**: Uniform distribution upper bound of parameter $\rho$ in the second Beta distribution of $W_{OFF}$
- **WOFFT3**: Uniform distribution lower bound of parameter $\theta$ in the second Beta distribution of $W_{OFF}$
- **WOFFT4**: Uniform distribution upper bound of parameter $\theta$ in the second Beta distribution of $W_{OFF}$
- **WOFFE**: decimal equivalent percentage weight occurring in the first Beta distribution of the weld offset, $W_{OFF}$

<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 $D_i$
<table>
<thead>
<tr>
<th>INDIA B</th>
<th>upper bound of the Beta distribution on $D_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>INDIR1</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $D_i$</td>
</tr>
<tr>
<td>INDIR2</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $D_i$</td>
</tr>
<tr>
<td>INDIT1</td>
<td>Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $D_i$</td>
</tr>
<tr>
<td>INDIT2</td>
<td>Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $D_i$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>THICA</th>
<th>THICB</th>
<th>THICR1</th>
<th>THICR2</th>
<th>THICT1</th>
<th>THICT2</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 wall thickness information**

$t$ (in.), the duct wall thickness. It is required for the HEX coil problem ($\text{KPROB} = 1$). 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 $\theta$.

<table>
<thead>
<tr>
<th>THICA</th>
<th>lower bound of the Beta distribution on $t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>THICB</td>
<td>upper bound of the Beta distribution on $t$</td>
</tr>
<tr>
<td>THICR1</td>
<td>Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $t$</td>
</tr>
<tr>
<td>THICR2</td>
<td>Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $t$</td>
</tr>
<tr>
<td>THICT1</td>
<td>Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $t$</td>
</tr>
<tr>
<td>THICT2</td>
<td>Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $t$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AOCA</th>
<th>AOCB</th>
<th>AOCR1</th>
<th>AOCR2</th>
<th>AOCT1</th>
<th>AOCT2</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 initial crack aspect ratio information**

$a/c$, the initial aspect ratio of an elliptic crack. It is required for the HEX coil problem ($\text{KPROB} = 1$). 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 \( \theta \).

**AOCA**

lower bound of Beta distribution on \( a/c \)

**AOCB**

upper bound of Beta distribution on \( a/c \)

**AOCR1**

Uniform distribution lower bound of parameter \( \rho \) in the Beta distribution of \( a/c \)

**AOCR2**

Uniform distribution upper bound of parameter \( \rho \) in the Beta distribution of \( a/c \)

**AOCT1**

Uniform distribution lower bound of parameter \( \theta \) in the Beta distribution of \( a/c \)

**AOCT2**

Uniform distribution upper bound of parameter \( \theta \) in the Beta distribution of \( a/c \)

**WITHA**

lower bound of the Beta distribution on \( W \)

**WITHB**

upper bound of the Beta distribution on \( W \)

**WITHR1**

Uniform distribution lower bound of parameter \( \rho \) in the Beta distribution of \( W \)

**WITHR2**

Uniform distribution upper bound of parameter \( \rho \) in the Beta distribution of \( W \)

**WITHT1**

Uniform distribution lower bound of parameter \( \theta \) in the Beta distribution of \( W \)

**WITHT2**

Uniform distribution upper bound of parameter \( \theta \) in the Beta distribution of \( W \)

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 \( \theta \).

**WITHA**

lower bound of the Beta distribution on \( W \)

**WITHB**

upper bound of the Beta distribution on \( W \)

**WITHR1**

Uniform distribution lower bound of parameter \( \rho \) in the Beta distribution of \( W \)

**WITHR2**

Uniform distribution upper bound of parameter \( \rho \) in the Beta distribution of \( W \)

**WITHT1**

Uniform distribution lower bound of parameter \( \theta \) in the Beta distribution of \( W \)

**WITHT2**

Uniform distribution upper bound of parameter \( \theta \) in the Beta distribution of \( W \)
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 \( \rho \).

Similarly, the last two parameters describe a Uniform distribution on \( \theta \).

\begin{align*}
AIA & : \text{lower bound of the Beta distribution on } a_i \\
AIB & : \text{upper bound of the Beta distribution on } a_i \\
AIR1 & : \text{Uniform distribution lower bound of parameter } \rho \text{ in the Beta distribution of } a_i \\
AIR2 & : \text{Uniform distribution upper bound of parameter } \rho \text{ in the Beta distribution of } a_i \\
AIT1 & : \text{Uniform distribution lower bound of parameter } \theta \text{ in the Beta distribution of } a_i \\
AIT2 & : \text{Uniform distribution upper bound of parameter } \theta \text{ in the Beta distribution of } a_i
\end{align*}

Distribution on narrow-band random load scale factor information

This line contains the parameters to define the narrow-band random load scale factor, \( \lambda_{D\text{RANDOM}} \) 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 \( d \).

\begin{align*}
LAMNA & : \text{lower bound of Uniform distribution of } k \text{ for the narrow-band random load scale factor} \\
LAMNB & : \text{upper bound of Uniform distribution of } k \text{ for the narrow-band random load scale factor} \\
LAMNC & : \text{coefficient of variation } C \text{ for the narrow-band random load scale factor} \\
LAMND & : \text{strain gage factor } d \text{ for the narrow-band random load scale factor}
\end{align*}
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 \( d \).

- **LAMSA**: lower bound of Uniform distribution of \( k \) for the sinusoidal load scale factor
- **LAMSB**: upper bound of Uniform distribution of \( k \) for the sinusoidal load scale factor
- **LAMSC**: coefficient of variation \( C \) for the sinusoidal load scale factor
- **LAMSD**: strain gage factor \( d \) for the sinusoidal load scale factor

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.

- **TIMUA**: Uniform distribution lower bound of parameter \( \mu \) in the Normal distribution of \( T_i \)
- **TIMUB**: Uniform distribution upper bound of parameter \( \mu \) in the Normal distribution of \( T_i \)
- **TISIGA**: Uniform distribution lower bound of parameter \( \sigma \) in the Normal distribution of \( T_i \)
- **TISIGB**: Uniform distribution upper bound of parameter \( \sigma \) in the Normal distribution of \( T_i \)

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 $p_i$
- **PCSIGA**: Uniform distribution lower bound of parameter $\sigma$ in the Normal distribution of $p_i$
- **PCSIGB**: 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 $\lambda_{OFF}$
AERDA  AERDB
[RE]      [RE]

Aerodynamic load scale factor distribution information
\( \lambda_{\text{AERO}} \) in Equation 2-5. It is required for the HEX coil problem (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

AERSA  AERSB
[RE]      [RE]

Aerostatic load scale factor distribution information
\( \lambda_{\text{ST}} \) in Equation 2-5. It is required for the HEX coil problem (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

ASTRA  ASTRB
[RE]      [RE]

Aeroloads stress analysis accuracy factor Uniform distribution information
\( \lambda_{\text{AERO}_{\text{str}}} \) in Equation 2-5. It is required for the HEX coil problem (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

SSTRA  SSTRB
[RE]      [RE]

Static stress analysis accuracy factor Uniform distribution information
\( \lambda_{\text{ST}_{\text{str}}} \) 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 (KPROB = 2).
Uniform distribution lower bound of static stress analysis accuracy factor

SSTRB Uniform distribution upper bound of static stress analysis accuracy factor

DSTRA DSTRB
[RE] [RE]

Dynamic stress analysis accuracy factor Uniform distribution information

\( \lambda_{DYNstra} \) in Equation 2-5. This is the dynamic stress analysis accuracy factor, and it is characterized by a Uniform distribution.

DSTRA Uniform distribution lower bound of dynamic stress analysis accuracy factor

DSTRB Uniform distribution upper bound of dynamic stress analysis accuracy factor

NEUBA NEUBB
[RE] [RE]

Neuber’s Rule accuracy factor Uniform distribution information

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

NEUBA Uniform distribution lower bound of Neuber’s Rule accuracy factor

NEUBB Uniform distribution upper bound of Neuber’s Rule accuracy factor

LAMKHA LAMKHB
[RE] [RE]

Threshold stress intensity factor uncertainty Uniform distribution information

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

LAMKHA Uniform distribution lower bound of threshold stress intensity factor range uncertainty

LAMKHB Uniform distribution upper bound of threshold stress intensity factor range uncertainty
Critical stress intensity factor uncertainty Uniform distribution information

\( \lambda K_c \) 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, \( \lambda_{all} \).

- **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 loge 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 \( KGROW = 2 \).
**MVARA**  Uniform distribution lower bound on Forman constant \(m\)

**MVARB**  Uniform distribution upper bound on Forman constant \(m\)

**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 (\(\sigma_x, \sigma_y\), etc.) from a node in a three-dimensional finite element mesh were used.

**NLOAD**

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

**PSTAT**  \(\text{[RE]}\)
**TSTAT**  \(\text{[RE]}\)
**MSTAT(1)**  \(\text{[RE]}\)
**MSTAT(2)**  \(\text{[RE]}\)
**VSTAT(1)**  \(\text{[RE]}\)
**VSTAT(2)**  \(\text{[RE]}\)

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

\(^1\) PROCRK 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 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.-Ibs), the dynamic torsional load magnitude for load I
M(1,I) M y (in.-Ibs) in Equation 2-1, the dynamic moment load magnitude
about the y-axis for load I
M(2,I) M z (in.-Ibs) 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

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

SXST σ x (psi), due to static loads
SYST σ y (psi), due to static loads

2 PROCRK 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]

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)  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.

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

SX(I)  \sigma_x (psi), due to dynamic load source I

SY(I)  \sigma_y (psi), due to dynamic load source I

SZ(I)  \sigma_z (psi), due to dynamic load source I

SXY(I)  \sigma_{xy} (psi), due to dynamic load source I

SXZ(I)  \sigma_{xz} (psi), due to dynamic load source I

SYZ(I)  \sigma_{yz} (psi), due to dynamic load source I

PCO
[RE]

External pressure

\rho_0 (psi) in Equation 2-1. This is the outer wall pressure. It is required for the HEX coil problem (KPROB = 1).

LOCAT
[INT]

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 ($\text{KPROB} = 1$).

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. $\text{NRAN}$ 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**: $E$ (psi) in Equation 2-2, Young's modulus of elasticity
- **COEXP**: $\alpha$ ($^\circ$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.

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

\text{E(J)} \quad \text{value of the strain } \epsilon \text{ at the upper end of the Jth segment of the stress-strain versus strain curve}

\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.

\textbf{FTY} \quad \text{KC} \quad \text{NDIV} \quad 
\text{IREGOP}

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

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.

\textbf{FTY} \quad \text{yield strength (psi)}

\textbf{KC} \quad \text{critical stress intensity factor (ksi}\sqrt{\text{in.}})

\textbf{NDIV} \quad \text{number of data divisions for the material data}

\textbf{IREGOP} \quad \text{regression option to fit the generalized Forman Equation 2-7}

\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

---

\textsuperscript{3} \text{When KGROW} = 1, \text{the selected value of } m \text{ 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.

- **DKTHO** stress intensity factor range, $\Delta K_{THO}$, at $R = 0$ (ksi $\cdot$ in.)
- **CO** empirical model constant $C_o$
- **DEE** empirical model exponent $d$

**Pee**

Parameter $p$ in the generalized Forman Equation 2-7. This is required when $\text{IREGOP} = 0$.

- **EMM**
- **QUE**

Parameters $m$ and $p$ in the generalized Forman Equation 2-7. This is required when $\text{IREGOP} = 1$.

- **EMM**
- **QUE**

Parameters $q$ and $p$ in the generalized Forman Equation 2-7. This is required when $\text{IREGOP} = 2$.

- **EMM**
- **QUE**
- **Pee**

Parameters $m$, $q$, and $p$ in the generalized Forman Equation 2-7. This is required when $\text{IREGOP} = 3$. 
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 \( i \). \( \text{NP}(i) \) for each division cannot exceed two hundred.

\[
\begin{align*}
\text{NP}(i) & \quad \text{number of data points in the division} \\
\text{RDATA}(i) & \quad \text{stress ratio for the data in the division}
\end{align*}
\]

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 \( \text{NP}(i) \) lines must be specified (i.e., the value of \( J \) goes from 1 to \( \text{NP}(i) \)). This block must be provided for each data division (i.e., the value of \( I \) goes from 1 to \( \text{NDIV} \)).

\[
\begin{align*}
\text{DADN}(I,J) & \quad \text{crack growth rate } da/dN \text{ (in./cycle)} \\
\text{DELK}(I,J) & \quad \text{stress intensity factor range } \Delta K \text{ (ksi}\sqrt{\text{in.}}) \\
\end{align*}
\]

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 \( \text{LDNAME}(I) \) in file CRKDAT. Reference time histories are typically generated by program NBSIN described in Sections 4.5, 6.6, and 7.7 of [1].

\[
\begin{align*}
\text{STRHIS}(I,J) & \quad \text{The points of the } I \text{th reference history} \\
& \quad \text{The points of the reference time history specified by } \text{LDNAME}(I). \text{ The data is entered one point per line for } J = 1, \ldots, \text{NRAN.}
\end{align*}
\]

6.1.4 Options and Capabilities
PROCRK 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 \( \text{NHYPER} \) outer loops and \( \text{NLIFE} \) 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 \( \text{NHYPER} = 10,000 \) and \( \text{NLIFE} = 1 \) were used to characterize component reliability, and single-loop runs with \( \text{NHYPER} = 1000 \) and \( \text{NLIFE} = 1 \) 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 \text{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.

### 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 (\( \text{KPROB} = 1 \)). In this example run, 1000 lives were simulated (\( \text{NLIFE} = 1 \) times \( \text{NHYPER} = 1000 \)) with no variation in the Forman constant \( m \), \( \text{KGROW} = 1 \). The Willenborg retardation model and the Neuber’s rule to calculate the mean stress are switched on (\( \text{IRET} = 1 \), \( \text{INEUB} = 1 \)). The B-lives\(^4\) to be provided are B.1, B.5, and B1 (\( \text{NBLIFE} = 3 \), \( \text{BLFPER}(1) = 0.001 \), \( \text{BLFPER}(2) = 0.005 \), \( \text{BLFPER}(3) = 0.01 \)). 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.

---

\(^4\) 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\%. 

6 - 31
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>Beta</td>
</tr>
<tr>
<td>Flow Conditions ($T_i$, $T_o$, $P_i$)</td>
<td>Beta</td>
</tr>
<tr>
<td>Weld Offset Stress Concentration Accuracy</td>
<td>Beta</td>
</tr>
<tr>
<td>Aerodynamic &amp; Static Load Scale Factors</td>
<td>Beta</td>
</tr>
<tr>
<td>Aeroloads &amp; Dynamic Stress Analysis Accuracy Factors</td>
<td>Beta</td>
</tr>
<tr>
<td>Neuber's Rule Accuracy</td>
<td>Beta</td>
</tr>
<tr>
<td>Threshold Stress Intensity Uncertainty Factor</td>
<td>Beta</td>
</tr>
<tr>
<td>Critical Stress Intensity Uncertainty Factor</td>
<td>Beta</td>
</tr>
<tr>
<td>Stress Intensity Factor Calculation Accuracy</td>
<td>Beta</td>
</tr>
<tr>
<td>Growth Calculation Accuracy</td>
<td>Beta</td>
</tr>
</tbody>
</table>

\[ 6 \to 32 \]
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°.

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\(\sqrt{\text{in}} \). 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. \( AK \) 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**

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>675</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>.001 0.005 0.010</td>
</tr>
<tr>
<td>.06</td>
<td>.06 0.00 0.00 0.0 0.0</td>
</tr>
<tr>
<td>.00</td>
<td>.00 0.00 0.00 0.0 0.0</td>
</tr>
<tr>
<td>1.00</td>
<td>0.1885 0.1915 0.50 0.50 0.5 20.</td>
</tr>
<tr>
<td>0.0113</td>
<td>0.0157 0.27273 0.27273 0.5 20.</td>
</tr>
<tr>
<td>.200</td>
<td>1.000 0.50 0.50 0.0 0.0</td>
</tr>
<tr>
<td>.005</td>
<td>0.005 0.00 0.00 0.0 0.0</td>
</tr>
<tr>
<td>2.00</td>
<td>2.00 0.15 1.00</td>
</tr>
<tr>
<td>2.00</td>
<td>2.00 0.20 1.00</td>
</tr>
<tr>
<td>486.</td>
<td>666.</td>
</tr>
<tr>
<td>799.</td>
<td>908.</td>
</tr>
<tr>
<td>3088.</td>
<td>4177.</td>
</tr>
<tr>
<td>0.80</td>
<td>1.20</td>
</tr>
<tr>
<td>0.50</td>
<td>1.50</td>
</tr>
<tr>
<td>0.80</td>
<td>1.20</td>
</tr>
<tr>
<td>0.90</td>
<td>1.10</td>
</tr>
<tr>
<td>0.80</td>
<td>1.20</td>
</tr>
<tr>
<td>0.60</td>
<td>1.40</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>0.90</td>
<td>1.10</td>
</tr>
<tr>
<td>-0.6931</td>
<td>0.557</td>
</tr>
<tr>
<td>3</td>
<td>0.00 0.00 -0.07214 0.00 0.00 0.00</td>
</tr>
<tr>
<td>'NBM3'</td>
<td>1</td>
</tr>
<tr>
<td>'SIN1'</td>
<td>2</td>
</tr>
<tr>
<td>'AEROL'</td>
<td>3</td>
</tr>
<tr>
<td>3640.</td>
<td>2</td>
</tr>
<tr>
<td>85.</td>
<td>2.30</td>
</tr>
<tr>
<td>0.00025</td>
<td>50.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>------</td>
</tr>
<tr>
<td>5</td>
<td>29000000.</td>
</tr>
<tr>
<td></td>
<td>0.615</td>
</tr>
<tr>
<td></td>
<td>0.693</td>
</tr>
<tr>
<td></td>
<td>0.753</td>
</tr>
<tr>
<td></td>
<td>0.813</td>
</tr>
<tr>
<td></td>
<td>0.873</td>
</tr>
<tr>
<td></td>
<td>0.933</td>
</tr>
<tr>
<td></td>
<td>0.993</td>
</tr>
<tr>
<td></td>
<td>1.029</td>
</tr>
<tr>
<td></td>
<td>1.053</td>
</tr>
<tr>
<td></td>
<td>1.053</td>
</tr>
<tr>
<td>6</td>
<td>21.95</td>
</tr>
<tr>
<td></td>
<td>55.77</td>
</tr>
<tr>
<td></td>
<td>144.85</td>
</tr>
<tr>
<td></td>
<td>322.73</td>
</tr>
<tr>
<td></td>
<td>1945.90</td>
</tr>
<tr>
<td></td>
<td>50688.0</td>
</tr>
<tr>
<td></td>
<td>'400F 316L WELDED, FROM RKD'</td>
</tr>
<tr>
<td>7</td>
<td>27000</td>
</tr>
<tr>
<td></td>
<td>4.0317</td>
</tr>
<tr>
<td>16</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>9.183E-10</td>
</tr>
<tr>
<td></td>
<td>1.138E-8</td>
</tr>
<tr>
<td></td>
<td>3.362E-8</td>
</tr>
<tr>
<td></td>
<td>8.473E-8</td>
</tr>
<tr>
<td></td>
<td>4.408E-8</td>
</tr>
<tr>
<td></td>
<td>5.838E-8</td>
</tr>
<tr>
<td></td>
<td>5.679E-8</td>
</tr>
<tr>
<td></td>
<td>7.220E-8</td>
</tr>
<tr>
<td></td>
<td>8.202E-8</td>
</tr>
<tr>
<td></td>
<td>7.440E-8</td>
</tr>
<tr>
<td></td>
<td>9.028E-8</td>
</tr>
<tr>
<td></td>
<td>1.133E-7</td>
</tr>
<tr>
<td></td>
<td>1.533E-7</td>
</tr>
<tr>
<td></td>
<td>1.629E-7</td>
</tr>
<tr>
<td></td>
<td>1.727E-7</td>
</tr>
<tr>
<td></td>
<td>2.321E-7</td>
</tr>
<tr>
<td>18</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>4.661E-9</td>
</tr>
<tr>
<td></td>
<td>2.469E-8</td>
</tr>
<tr>
<td></td>
<td>1.387E-7</td>
</tr>
<tr>
<td></td>
<td>1.162E-7</td>
</tr>
<tr>
<td></td>
<td>1.631E-7</td>
</tr>
<tr>
<td></td>
<td>1.539E-7</td>
</tr>
<tr>
<td></td>
<td>1.562E-7</td>
</tr>
<tr>
<td></td>
<td>1.839E-7</td>
</tr>
<tr>
<td></td>
<td>2.089E-7</td>
</tr>
<tr>
<td></td>
<td>3.497E-7</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</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>
<tr>
<td>17</td>
<td>0.16</td>
</tr>
<tr>
<td>1.775E-7</td>
<td>9.10</td>
</tr>
<tr>
<td>1.969E-7</td>
<td>9.91</td>
</tr>
<tr>
<td>2.454E-7</td>
<td>10.79</td>
</tr>
<tr>
<td>2.543E-7</td>
<td>11.78</td>
</tr>
<tr>
<td>4.050E-7</td>
<td>12.83</td>
</tr>
<tr>
<td>5.355E-7</td>
<td>13.96</td>
</tr>
<tr>
<td>7.369E-7</td>
<td>15.20</td>
</tr>
<tr>
<td>1.058E-6</td>
<td>16.53</td>
</tr>
<tr>
<td>2.008E-6</td>
<td>18.00</td>
</tr>
<tr>
<td>2.650E-6</td>
<td>19.56</td>
</tr>
<tr>
<td>4.238E-6</td>
<td>21.24</td>
</tr>
<tr>
<td>5.679E-6</td>
<td>23.11</td>
</tr>
<tr>
<td>8.308E-6</td>
<td>25.07</td>
</tr>
<tr>
<td>9.687E-6</td>
<td>27.33</td>
</tr>
<tr>
<td>1.649E-5</td>
<td>32.96</td>
</tr>
<tr>
<td>2.335E-5</td>
<td>38.56</td>
</tr>
<tr>
<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
Output File - CRKRES

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

**PROC RK**

**INPUT DATA**

<table>
<thead>
<tr>
<th>DRIVERS</th>
<th>PARAMETER DISTRIBUTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RH0</td>
</tr>
<tr>
<td></td>
<td>THETA</td>
</tr>
<tr>
<td>WELD OFFSET (%)</td>
<td>Be(0.06, 0.06)</td>
</tr>
<tr>
<td></td>
<td>U(0.000000, 0.000000)</td>
</tr>
<tr>
<td></td>
<td>U(0.00, 0.00)</td>
</tr>
<tr>
<td></td>
<td>U(0.000000, 0.000000)</td>
</tr>
<tr>
<td></td>
<td>TEST = 1.00</td>
</tr>
<tr>
<td>INNER DIAMETER</td>
<td>Be(0.1885, 0.1915)</td>
</tr>
<tr>
<td></td>
<td>U(0.500000, 0.500000)</td>
</tr>
<tr>
<td>WALL THICKNESS</td>
<td>Be(0.0113, 0.0157)</td>
</tr>
<tr>
<td></td>
<td>U(0.27273, 0.27273)</td>
</tr>
<tr>
<td>CRACK SHAPE A/C</td>
<td>Be(0.2000, 1.0000)</td>
</tr>
<tr>
<td></td>
<td>U(0.500000, 0.500000)</td>
</tr>
<tr>
<td>CRACK SIZE A</td>
<td>Be(0.0050, 0.0050)</td>
</tr>
<tr>
<td></td>
<td>U(0.000000, 0.000000)</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: 1.0000000000</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: 1.000000000</td>
</tr>
<tr>
<td>INNER TEMPERATURE</td>
<td>NORMAL: MU(486.0, 666.0)</td>
</tr>
<tr>
<td></td>
<td>SIGMA(29.0, 56.5)</td>
</tr>
<tr>
<td>OUTER TEMPERATURE</td>
<td>NORMAL: MU(799.0, 908.0)</td>
</tr>
<tr>
<td></td>
<td>SIGMA(49.5, 48.0)</td>
</tr>
<tr>
<td>INNER PRESSURE</td>
<td>NORMAL: MU(3808.0, 4177.0)</td>
</tr>
<tr>
<td></td>
<td>SIGMA(69.0, 69.0)</td>
</tr>
</tbody>
</table>

6 - 37
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>P 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>
</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>
</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>
</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>
</tr>
</tbody>
</table>

MISCELLANEOUS INPUT

EXTERNAL PRESSURE 3640.
ANALYSIS LOCATION 2
ANGLE THETA (DEGREES) 85.0
WILLENBORG OVERLOAD FACTOR 0.23000E+01

6 - 38
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
# 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>(\frac{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>da/dN</td>
<td>DELK</td>
</tr>
<tr>
<td>----------</td>
<td>-------</td>
</tr>
<tr>
<td>0.72200E-07</td>
<td>0.39500E+01</td>
</tr>
<tr>
<td>0.82020E-07</td>
<td>0.41800E+01</td>
</tr>
<tr>
<td>0.74400E-07</td>
<td>0.44200E+01</td>
</tr>
<tr>
<td>0.90280E-07</td>
<td>0.46700E+01</td>
</tr>
<tr>
<td>0.11330E-06</td>
<td>0.49400E+01</td>
</tr>
<tr>
<td>0.15330E-06</td>
<td>0.52200E+01</td>
</tr>
<tr>
<td>0.16290E-06</td>
<td>0.55100E+01</td>
</tr>
<tr>
<td>0.17270E-06</td>
<td>0.58100E+01</td>
</tr>
<tr>
<td>0.23210E-06</td>
<td>0.59900E+01</td>
</tr>
</tbody>
</table>

**STRESS RATIO R = 0.70**

<table>
<thead>
<tr>
<th>da/dN</th>
<th>DELK</th>
</tr>
</thead>
<tbody>
<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>

**STRESS RATIO R = 0.16**

<table>
<thead>
<tr>
<th>da/dN</th>
<th>DELK</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.17750E-06</td>
<td>0.91000E+01</td>
</tr>
<tr>
<td>0.19690E-06</td>
<td>0.99100E+01</td>
</tr>
<tr>
<td>0.24540E-06</td>
<td>0.10790E+02</td>
</tr>
<tr>
<td>0.25430E-06</td>
<td>0.11780E+02</td>
</tr>
<tr>
<td>0.40500E-06</td>
<td>0.12830E+02</td>
</tr>
<tr>
<td>0.53550E-06</td>
<td>0.13960E+02</td>
</tr>
<tr>
<td>0.73690E-06</td>
<td>0.15200E+02</td>
</tr>
<tr>
<td>0.10580E-05</td>
<td>0.16530E+02</td>
</tr>
<tr>
<td>0.20080E-05</td>
<td>0.18000E+02</td>
</tr>
<tr>
<td>0.26500E-05</td>
<td>0.19560E+02</td>
</tr>
<tr>
<td>0.42380E-05</td>
<td>0.21240E+02</td>
</tr>
<tr>
<td>0.56790E-05</td>
<td>0.23110E+02</td>
</tr>
<tr>
<td>0.83080E-05</td>
<td>0.25070E+02</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

\[ c \quad n \quad m \quad p \quad q \]

\[ 0.56708E-11 \quad 0.25314E+01 \quad -0.19413E+01 \quad 0.71522E+00 \quad -0.81965E+00 \]

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 = 675.0000000000000
IOUT - OUTPUT CONTROL VARIABLE = 0
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 $da/dN$ 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: BDLCO, RELATO, DUMP, IOUTPR, and LOWLIF. BDLCO 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.
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>675</td>
<td>Random number seed</td>
</tr>
<tr>
<td>0</td>
<td>Value of output dump controller</td>
</tr>
<tr>
<td>1</td>
<td>Inner loop size</td>
</tr>
<tr>
<td>20000</td>
<td>Outer loop size</td>
</tr>
<tr>
<td>50</td>
<td>Symmetry number</td>
</tr>
<tr>
<td>2</td>
<td>Type of S/N variation</td>
</tr>
<tr>
<td>0</td>
<td>Request for truncated Normal median S/N curve</td>
</tr>
<tr>
<td>0</td>
<td>Controls materials process variation</td>
</tr>
<tr>
<td>1</td>
<td>Type of materials intrinsic variation</td>
</tr>
<tr>
<td>5</td>
<td>Number of B-lives</td>
</tr>
</tbody>
</table>

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 $a_{G}$ 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. |

| 0.0 | 0.020 |
| 1640.0 | 40.67 |
| 0.0 | 0.003 |
| 0.0 | 0.003 |
| 0.96 | 1.04 |
| 0.80 | 1.20 |
| 0.975 | 1.025 |
| 0.70 | 1.30 |
| 0.00 | 0.00 |
| 0.00 | 0.00 |

Normal distribution on $a_{A}$ 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 $\lambda_{a}$

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 BDLLCD
Nominal mechanical strain $\varepsilon_{Mnom}$ and corresponding rotor speed $\omega_0$

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 $f_A$

| 0.00727362 | 0.000067442 | -0.000059109 | -3.52929E-08 | 1.07611E-08 | -2.74419E-08 |

Coefficients for the shutdown transient response surface functions $f_{D1}$, $f_{D2}$, and $f_{D3}$

| -0.132623 | 0.000227427 | -0.000059290 | 0.0 | 0.0 | 4.71714E-08 |

Nominal time history

| 225.8 | 0.0 | $\omega(t_1)$, $\varepsilon_{TH}(t_1)$ |
| 3025.1 | -0.196921 | $\omega(t_2)$, $\varepsilon_{TH}(t_2)$ |
| 6138.8 | 0.146025 | $\omega(t_3)$, $\varepsilon_{TH}(t_3)$ |
| 8309.0 | -0.200128 | $\omega(t_4)$, $\varepsilon_{TH}(t_4)$ |
| 0.0 | 0.007393 | $\omega(t_5)$, $\varepsilon_{TH}(t_5)$ |

Description of specific material S/N data set

'I'RT, PWA 1480, 001 DIRECTION

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

| 1.54 | 1.57 | 1 | 9 |

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

| 8 | -1.0 | 1 |

Figure 6.2-1 Format for File BDLLCD (Cont'd)
| $\epsilon_1, N_1$ | 0.89 | 6800. |
| $\epsilon_2, N_2$ | 0.89 | 15000. |
| $\epsilon_3, N_3$ | 0.67 | 27000. |
| $\epsilon_4, N_4$ | 0.67 | 43200. |
| $\epsilon_5, N_5$ | 0.56 | 139300. |
| $\epsilon_6, N_6$ | 0.56 | 545200. |
| $\epsilon_7, N_7$ | 0.56 | 147000. |
| $\epsilon_8, N_8$ | 0.39 | 4344800. |

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 0.00 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 BLULCD (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

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

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 ≤ 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 \( h_{gas} \)
Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $h_{gas}$

Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $h_{gas}$

Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $h_{gas}$

Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $h_{gas}$

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 $T_{gas}$

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 \( \theta \).

### Parameters

- **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

\( \lambda_G \) 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 \( \theta \).

### Parameters for \( \lambda_G \)

- **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 $\text{Normal}(\mu, \sigma^2)$ 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 $\text{Normal}(\mu, \sigma^2)$ 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 $\text{Normal}(\mu, \sigma^2)$ 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 $e_D$ 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 $e_D$
- **FDERRS** standard deviation, $\sigma$, of Normally distributed $e_D$

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 loge 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.

- **LAMDA** lower bound of damage accumulation accuracy factor
- **LAMDAB** upper bound of damage accumulation accuracy factor
LAMTMA  LAMTMB
[RE]  [RE]

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**
[RE]  [RE]

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

**PERIOD**
[RE]

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**
[RE]

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

**NTIME**
[INT]

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.

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

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.

\[
f_{D1}(m, T_s) = d_{1A} + d_{1B} \ T_s + d_{1C} \ m + d_{1D} \ T_s^2 + d_{1E} \ m^2 + d_{1F} \ T_s \ m
\]

\[
t_d = f_{D2}(m, T_s) = d_{2A} + (T_s - d_{2B})/m
\]

\[
\omega(t_6) = f_{D3}(t_d) = d_{3A} + d_{3B} \ t_d
\]
RPM(I) ETHNOM(I)
[RE] [RE]

The points of the nominal time history
The data is entered as rotor speed, thermal strain pairs, one pair per line for I = 1, ..., 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></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>yield strength corresponding to the specific material data set (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ultimate strength corresponding to the specific material data set (%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NDIV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>number of data divisions for the specific material data set</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NPTS(0)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>total number of points in the specific material S/N data set</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NUM RATIO REG
[INT] [RE] [INT]

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></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>number of S/N data points in the data division</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.
Life Boundaries\(^{11}\)
The upper boundaries of the life regions are specified (cycles). The value of \(L\) goes from \(\text{ZROREG}\) to the total number of regions (equal to \(\text{NUMREG} + \text{NNODAT}\)). If a non-zero tensile point is specified, then \(\text{ZROREG} = 0\) else \(\text{ZROREG} = 1\). The program expects the upper bound of the last life region to be \(10^36\), a proxy for \(\infty\).

CZERO

Prior information on coefficient of variation of fatigue strength\(^{12}\)
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 CZERO is not in use.

MPNT(L) MZERO(1,L) MZERO(2,L)

Prior information on the materials shape parameter \(m\)\(^{13}\)
The number of 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 (\(\text{NUMREG} + \text{NNODAT}\)). If \(\text{VARY} = 3\) is specified (truncated Normal distribution on \(m\)), then a prior range of \(m\) must be specified for each region.

\begin{align*}
\text{MPNT}(L) & \quad \text{The number of points, 0, 1, or 2 (no prior on } m, \text{ a point prior on } m, \text{ or a prior over a range of } m, \text{ respectively), in MZERO( ) for each region.} \\
\text{MZERO}(1,L) & \quad \text{The lower bound on the range of } m \text{ or the value of the point prior for } m.
\end{align*}

---

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

\(^{12}\) 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.

\(^{13}\) 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\textsuperscript{14}

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 $\text{MED } K' / \text{MED } K$ where $\text{MED } K$ is the median value over all heats for the strain (%) at a life of one cycle, and $\text{MED } K'$ 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.

\textsuperscript{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, 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

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 NSETS. If there is no related data, then file RELATD will only contain the number “0”. NSETS cannot exceed five.

DESCRP(J)

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)

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. 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.
NUM  RATIO  REG
[INT]  [RE]  [INT]

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

**RAWSTR(I,J)**  **RAWNF(I,J)**
[RE]  [RE]

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 varia-
tion 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 intrin-
sic variation of the fatigue life of the material included (VARY = 1).

A printout of intermediate calculations in various parts of the program may be ob-
tained 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 op-
tion provides for calculation of an empirical median S/N curve if the truncated Normal
distribution or "bootstrapping" is employed.\textsuperscript{16} 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 vari-
ation 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 ex-
ample run, 20,000 lives were simulated (NLIFE = 1 times NHYPER = 20,000)
using Uniform shape parameter variation, VARY = 2 and NMED = 0; Weibull in-
trinsic 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\textsuperscript{17} to be provided are B.1, B.2, B.3, B.4, B.5, B.6, B.7, B.8, B.9,
and B1 (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 en-
geineering analysis and to Section 3.3 for the results of the case study for this com-
ponent.

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

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

\textsuperscript{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. 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 ln 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. 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.

Input File - BLDLCO

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>675</td>
<td>0</td>
<td>1</td>
<td>20000</td>
<td>50</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>0.001</td>
<td>0.002</td>
<td>0.003</td>
<td>0.004</td>
<td>0.005</td>
<td>0.006</td>
<td>0.007</td>
<td>0.008</td>
<td>0.009</td>
<td>0.010</td>
</tr>
<tr>
<td>676.</td>
<td>2730.</td>
<td>0.50</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>782.</td>
<td>1982.</td>
<td>0.50</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2730.</td>
<td>2730.</td>
<td>0.50</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>1.5</td>
<td>0.50</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>37592.</td>
<td>507.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1640.0</td>
<td>40.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td>0.003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.96</td>
<td>1.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.80</td>
<td>1.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.975</td>
<td>1.025</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.70</td>
<td>1.30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.295</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38482.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.00727362</td>
<td>0.000067442</td>
<td>-0.000059109</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-3.52929E-08</td>
<td>1.07611E-08</td>
<td>-2.74419E-08</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

20 The information aggregation calculations are discussed in [1], Pages 2-6 through 2-14.
-0.132623 0.000227427 -0.000059290 0.00 0.00 4.71714E-08
0.20 950.0
30523.07 -21846.15
225.8 0.0
3025.1 -0.196921
6138.8 0.146025
8309.0 -0.200128
0.0 0.007393

'RT, PWA 1480, 001 DIRECTION'
1.54 1.57 1 8
8 -1.0 1
0.89 6800.
0.89 15000.
0.67 27000.
0.67 43200.
0.56 139300.
0.56 545200.
0.56 147000.
0.39 4344800.
0.00
1 0
1.0E+36
0.00
0 0.000 0.000

Input File - RELATO

Output File - BLDDLCO

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

INPUT DATA

<table>
<thead>
<tr>
<th>DRIVERS</th>
<th>PARAMETER DISTRIBUTIONS</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>

6 - 72
Tgas UNCERT. Be( 0.50, 1.50) U(0.50000, 0.50000) U( 0.0, 0.0) N( MEAN, STD. DEV.)

ROTOR SPEED VARIATION (rpm) AT TIME T5 N( 37592.0, 507.0)
Faccel MODELING ERROR N( 0.0, 0.2000E-01)
STARTING DECEL TEMPERATURE (deg R) N( 1640.00, 40.67)
Fdecel MODELING ERROR N( 0.0, 0.3000E-02)

STRAIN DUE TO GAS BENDING (%) U( 0.00000, 0.00000)
LAMBD A BLADE PULL U( 0.96000, 1.04000)
MECHANICAL ANALYSIS FACTOR U( 0.80000, 1.20000)
COEFFICIENT OF THERMAL EXPANSION FACTOR U( 0.97500, 1.02500)
THERMAL ANALYSIS FACTOR U( 0.70000, 1.30000)

DAMAGE MODEL ACCURACY U(ln 1.00000, ln 1.00000)
TMF MODEL ACCURACY U(ln 1.00000, ln 1.00000)

OTHER STRAIN HISTORY INPUT

NOMINAL MECHANICAL STRAIN (%) 0.2950
NOMINAL ROTOR SPEED (rpm) 38482.
STRAIN-TIME HISTORY PERIOD (missions) 1.00
STRAIN-TIME HISTORY NOISE FILTER (%) 0.00000
NUMBER OF POINTS IN HISTORIES 6
WALKER EXponent 0.50
COEFFICIENTS OF ACCELERATION AND DECELERATION FUNCTIONS

THERMAL STRAIN AT STARTUP (%):
\[ F_{\text{accel}}(T_{\text{gas}}, H_{\text{gas}}) = 0.727362 \times 10^{-2} + 0.674420 \times 10^{-4} \times T_{\text{gas}} + -0.591090 \times 10^{-4} \times H_{\text{gas}} + -0.352929 \times 10^{-7} \times T_{\text{gas}}^2 + 0.107611 \times 10^{-7} \times H_{\text{gas}}^2 + -0.274419 \times 10^{-7} \times T_{\text{gas}} \times H_{\text{gas}} \]

THERMAL STRAIN AT SHUTDOWN (%):
\[ F_{\text{decel}}(m, T_{\text{start}}) = -0.132623 \times 10^0 + 0.227427 \times 10^{-3} \times T_{\text{start}} + -0.592900 \times 10^{-4} \times m + 0.000000 \times 10^0 \times T_{\text{start}}^2 + 0.000000 \times 10^0 \times m^2 + 0.471714 \times 10^{-7} \times T_{\text{start}} \times m \]

TIME AT SHUTDOWN (sec):
\[ F_{\text{decel2}}(m, T_{\text{start}}) = 0.200000 \times 10^0 + (T_{\text{start}} - 0.950000 \times 10^3) / m \]

ROTOR SPEED AT SHUTDOWN (rpm):
\[ F_{\text{decel3}}(t) = 0.305231 \times 10^5 + -0.218462 \times 10^5 \times t \]

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: 0.15400E+01
ULTIMATE STRENGTH: 0.15700E+01
NUMBER OF POINTS: 8

<table>
<thead>
<tr>
<th>Original S/N</th>
<th>Stress</th>
<th>Transformed S/N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6 - 74
<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>

**Weibull Variation**

**B Lives:** EMPIRICAL

<table>
<thead>
<tr>
<th>B Lives</th>
<th>Value</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>
Output File - RELATO

NUMBER OF DATA SETS: 0

NOTE: ALL Kt ASSUMED TO BE 1.0

TRANSFORMED DATA

Output File - DUMP

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

RESULTS OF INFORMATION AGGREGATION CALCULATIONS

95% CONFIDENCE INTERVALS ON C AND m FOR EACH REGION

REGION: 1 
Io = ( 0.054422790, 0.185977300) 
Jo = ( 5.152090000, 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  \( E(\beta_o) = 15.7104  \)  \( E(k) = 1.0909 \)

<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

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

Output File - LOWLIF

<table>
<thead>
<tr>
<th>i</th>
<th>( b_i )</th>
<th>( l_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.500000E-04</td>
<td>11.4674</td>
</tr>
<tr>
<td>2</td>
<td>0.100000E-03</td>
<td>20.5764</td>
</tr>
<tr>
<td>3</td>
<td>0.150000E-03</td>
<td>20.9020</td>
</tr>
<tr>
<td>4</td>
<td>0.200000E-03</td>
<td>23.3439</td>
</tr>
<tr>
<td>5</td>
<td>0.250000E-03</td>
<td>28.7136</td>
</tr>
<tr>
<td>6</td>
<td>0.300000E-03</td>
<td>33.3230</td>
</tr>
<tr>
<td>7</td>
<td>0.350000E-03</td>
<td>35.4286</td>
</tr>
<tr>
<td>8</td>
<td>0.400000E-03</td>
<td>37.5925</td>
</tr>
<tr>
<td>9</td>
<td>0.450000E-03</td>
<td>45.9977</td>
</tr>
<tr>
<td>10</td>
<td>0.500000E-03</td>
<td>50.0363</td>
</tr>
<tr>
<td>11</td>
<td>0.550000E-03</td>
<td>50.1602</td>
</tr>
<tr>
<td>12</td>
<td>0.600000E-03</td>
<td>50.6590</td>
</tr>
<tr>
<td>13</td>
<td>0.650000E-03</td>
<td>54.5432</td>
</tr>
<tr>
<td>14</td>
<td>0.700000E-03</td>
<td>54.9887</td>
</tr>
<tr>
<td>15</td>
<td>0.750000E-03</td>
<td>56.3990</td>
</tr>
<tr>
<td>16</td>
<td>0.800000E-03</td>
<td>57.8591</td>
</tr>
<tr>
<td>17</td>
<td>0.850000E-03</td>
<td>62.6331</td>
</tr>
<tr>
<td>18</td>
<td>0.900000E-03</td>
<td>65.5875</td>
</tr>
<tr>
<td>19</td>
<td>0.950000E-03</td>
<td>68.4943</td>
</tr>
<tr>
<td>20</td>
<td>0.100000E-02</td>
<td>69.3627</td>
</tr>
<tr>
<td>21</td>
<td>0.105000E-02</td>
<td>73.8416</td>
</tr>
<tr>
<td>22</td>
<td>0.110000E-02</td>
<td>74.9508</td>
</tr>
<tr>
<td>23</td>
<td>0.115000E-02</td>
<td>75.4585</td>
</tr>
<tr>
<td>24</td>
<td>0.120000E-02</td>
<td>78.1945</td>
</tr>
<tr>
<td>25</td>
<td>0.125000E-02</td>
<td>82.3033</td>
</tr>
<tr>
<td></td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----------------</td>
<td>---</td>
</tr>
<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>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 BLDLCD.

**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.\(^{21}\) 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 \(Mo\)

ERROR: NO INTERSECTION BETWEEN \(Jo\) AND \(Mc\)

\(^{21}\) 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 \texttt{NTIME} larger than 50.

ERROR: SXY \( \geq 0 \) 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 BDLCF. BDLLCD 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 BDLCF. It contains the echo of the information contained in BDLLCD and provides the simulated failure distribution B-life information.

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

DUMP
This file is opened in BDLCF. It contains the results of the information aggregation portion of the materials model calculations, such as $I_o$ and $J_o$; 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 BDLCF. It contains information on the particular run that is not echoed to BDLCO 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 BLDLCF. 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, (\rho), (\theta)) 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((\alpha), 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>5.1.2.6</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((\mu), (\sigma^2)) 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>PRYRV³</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>RANDOM³</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>
### 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_{\text{AERO}} ) 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_{\text{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>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<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>$\lambda_{AERO_{act}}$ 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(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</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>$c (\circ R)$ 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 $\sqrt{in.}$) 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 $\Delta K_{\text{eff}}$ 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 $\sqrt{\text{in.}}$) at $R = 0$ used in Equation 2-10.</td>
</tr>
<tr>
<td>DLTAT</td>
<td>RE</td>
<td>DELTA T. $\Delta T (\circ R)$ 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 ε 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>INDIAA</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>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<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 $K_{max}$ (ksi).</td>
</tr>
<tr>
<td>KMAXEF</td>
<td>RE</td>
<td>Effective maximum stress intensity factor $K_{max,eff}$ after retardation given in Equation 2-12.</td>
</tr>
<tr>
<td>KMIN(2)</td>
<td>RE</td>
<td>Minimum stress intensity factor $K_{min}$ (ksi).</td>
</tr>
<tr>
<td>KMINEF</td>
<td>RE</td>
<td>Effective minimum stress intensity factor $K_{min,eff}$ after retardation given in Equation 2-12.</td>
</tr>
<tr>
<td>KOFF</td>
<td>RE</td>
<td>$K_{OFF}$ 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_{RANDOM} ) 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>$\lambda_D\text{SINUSOIDAL}$ 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 $\lambda_{OFF}$ in Equation 2-3, the accuracy factor for the weld offset eccentricity stress concentration factor, $K_{OFF}$.</td>
</tr>
<tr>
<td>LAMWA</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda_{OFF}$.</td>
</tr>
<tr>
<td>LAMWB</td>
<td>RE</td>
<td>Uniform distribution upper bound of $\lambda_{OFF}$.</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,<em>) is $M_y$ (in.-lbs) in Equation 2-1, the moment load components about the y-axis; and M(2,</em>) 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,<em>) is $M_y$ (in.-lbs) in Equation 2-1, the moment load components about the y-axis; and MLAM(2,</em>) is $M_z$ (in.-lbs) in Equation 2-1, the moment load components about the z-axis.</td>
</tr>
<tr>
<td>MSLAM(2)</td>
<td>RE</td>
<td>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;</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>MSTAT(2)</td>
<td>RE</td>
<td>1-D array containing the static moment load components. MSTAT(1) is $M_y$ (in.-lbf) in Equation 2-1, the moment load component about the y-axis; and MSTAT(2) is $M_z$ (in.-lbf) 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 divisions.</td>
</tr>
<tr>
<td>NEUB</td>
<td>RE</td>
<td>Randomly selected Neuber's rule model accuracy factor $\lambda_{neu}$.</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>NLFIFET</td>
<td>INT</td>
<td>Total number of lives calculated by program PROCRK. Value of $N_{HYPER} \times N_{LIFE}$.</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>
</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>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>$\nu$ 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>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<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 data division.</td>
</tr>
<tr>
<td>REFF</td>
<td>RE</td>
<td>Effective stress ratio $K_{min.eff}/K_{max.eff}$ 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, $K_{OFF}$.</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>$\lambda_{ST_{sp}}$ 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>SSTRB</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 $\sigma_D(t)$ (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>SYST</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>
</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>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>
</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>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. TYPE(<em>) = 1, use the narrow-band random load scale factor; TYPE(</em>) = 2, use the superimposed sinusoidal load scale factor; and 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>
</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>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>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<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>FYRV</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>BETAGN</td>
<td>7-74</td>
</tr>
<tr>
<td>GAM</td>
<td>7-75</td>
</tr>
</tbody>
</table>

PROCRK Version 92.5
<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>7-51</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* THIS PROGRAM DRAWS MANY ROUTINES FROM PROGRAM HEXHCF (JPL PUB 92-15)
C* DATE: DECEMBER 1992
C* VERSION: 92.5
C* Copyright (C) 1991, California Institute of Technology.
C* U.S. Government Sponsorship under NASA Contract NAS7-518
C* is acknowledged.
C***************************************************************************

PROGRAM PROCRK
C=================================================================
C SUBPROGRAMS: SETDEF, INPUT, GRODAT, HYPDRW, PARDRW, LIFCAL, INSORT
C FILES: 1:CRKDAT-OLD; 3:CRKRES-NEW; 8:IOUTPR-NEW;
9:LOWLIF-NEW; 11-26 :user named-OLDC
C IMPLICIT NONE
INTEGER MAXBLF, MAXLD, MAXLIF, MAXM, MAXSEG
PARAMETER (MAXBLF = 10, MAXLD = 16, MAXLIF = 1000,
& MAXM = 20000, MAXSEG = 10)
INTEGER BLFPOS, I, IOUT, J, K, LOCAT, NBLIFE, NDIR,
& NCRL, NHYPER, NLIIFE, NLIIFET, NLOAD, NRAN,
& NUM, NUMSEG, TYPE(MAXLD)
INTEGER INEUB, IRET, KGROW, KPROB
DOUBLE PRECISION RAND
REAL AERD, AERDA, AERDB, AERS, AERSA, AERSB, AI, AIA,
& AIT, AIT2, ANGLE, AOCB, AOCR, AOCR1,
& AOCR2, AOC2, AOC2T, AOC2T2, ASTR, ASTR2, ASTR2B,
& BLFPER(MAXBLF), CEE, CI, CO,
& COEXP, DEE,
& DKTTHO, DLTAT, DPCMU, DPCSIG, DSTR, DSTRA, DSTRB,
& DTIMU, DTISIG, DTMU, DTMUSIG, E(MAXSEG), EM,
& EMM, ENN, FTY,
& INDIA, INDIAA, INDIA1, INDIA2, INDIA3,
& INDIT, INDIT1, INDIT2, IC, KC,
& KLAM, KLAM2, KLAM3, LAMX, LAMXH, LAMKH, LAMKHB,
& LAMKC, LAMKCA, LAMKCB, LAMGR, LAMGR2, LAMGRB,
& LAMN, LAMN2, LAMN3, LAMN4, LAMN5, LAMNSG,
& LAMSK, LAMSB, LAMSD, LAMSM, LAMSSG, LAMSW,
& LAMX, LAMX2, LAMX3, LIFE(MAXLIF)
REAL M(2, MAXLD), MSTAT(2),
& MVAR, MVARA, MVARB, NEUB, NEUBA, NEUBB, NEULF, NU,
& P(MAXLD), PC, PCMU, PCMU2, PCO, PCSIG, PCSIGA,
& PCSIGB, PEE, PERIOD, PSTAT, QUE, RSO,
& SE(MAXSEG), SSTR, SSTRB,
& SSTRB, STH(IS(MAXL, MAXM), SX(MAXL, MAXM), SXS, SY(MAXL, SY(SX(MAXL), SXS, SY(MAXL),
& SYX(MAXL), SYX2, SYX2T, SYX2T2, SYX2T3, SYX2T4, SYX2T5, SYX2T6, SYX2T7, SYX2T8,
& T(MAXL), T(a, THIC, THICB, THICR, THICR2, THICT, THICT1, THICT2, TIMU, TIMUA,
& TIMUB, TISIG, TISIGA, TISIGB, TOMU,
& TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
& TRUNC, TSTAT, V(2, MAXLD),
& VSTAT(2), WIDTH, WTHK,
& WTHB, WTHH1, WTHH2, WTHH3, WTHH4, WTHH5, WTHH6, WTHH7, WTHH8, WTHH9,
& WOFF, WOFFA, WOFFB, WOFFC, WOFFD,
& WOFF, WOFFH, WOFFL, WOFFR, WOFFR1, WOFFR2, WOFFR3,
& WOFFR4, WOFFT, WOFFT1, WOFFT2, WOFFT3, WOFFT4
CHARACTER*6 LDNAME(MAXLID)

7 - 25
COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
& M, V, PCO, SXST, SYST, SZST, SXYS, SXXST,
& SXXST, SXXST, SX, SY, SZ, SXX, SYY
COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA,
& AIT, AIT, AIT, AIT, AIT,
& AOCA, AOCB, AOCA, AOCA, AOCT, AOCT,
& ASTR, ASPR,
& DPCMU, DPCSIG, DSTRB, DSTRB, DSTRB, DSTRB, DSTRB, DSTRB,
& KLAM, KLAM, KLAM, KLAM, KLAM, KLAM, KLAM, KLAM,
& LAMNA, LAMNB, LAMNC, LAMND, LAMNU, LAMNSG,
& LAMSA, LAMB, LAMSC, LAMSD, LAMSHU, LAMSSG,
& LAMSA, LAMB, LAMSC, LAMSD, LAMSHU, LAMSSG,
& MAVAR, MVAR, MVARB, NEUBA, NEUBB,
& PCMU, PCMUA, PCMUB, PCMSA, PCMSA, PMSIGA, PCMSIGB,
& RAND,
& SSTRA, SSTRA,
& THICA, THICB, THICR1, THICR2, THICR, THICT1, THICT2, THICT,
& TIMU, TIMU, TIMU, TIMU, TIMU, TIMU, TIMU, TIMU,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WOFFR, WOFFR, WOFFR, WOFFR, WOFFR,
& WITHA, WITHB, WITHA, WITHA, WITHA, WITHA, WITHA, WITHA,
& WITHT, WITHT, WITHT, WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
& WITHT, WITHT, WITHT, WITHT, WITHT,
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 -- \( \Delta T \) (\( Ti - To \))

EQUAL TO PCMUB - PCMUA

EQUAL TO PCSIGB - PCSIGA

EQUAL TO TIMUB - TIMUA

EQUAL TO TISIGB - TISIGA

EQUAL TO TOMUB - TOMUA

EQUAL TO TOSIGB - TOSIGA

1-D ARRAY WHICH CONTAINS THE STRAIN VALUES

COEFFICIENT \( "m" \) IN THE GENERALIZED FORMAN MODEL

COEFFICIENT \( "n" \) IN THE GENERALIZED FORMAN MODEL

LOGICAL VARIABLE TO INDICATE UNSTABLE CRACK, \( K > K_{cr} \)

1-D ARRAY CONTAINING UNIT NUMBER FOR STRESS-TIME HISTORIES

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

FILE TEST -- USED TO TEST EXISTENCE OF FILE

MATERIAL YIELD STRENGTH

SELECTED INTERIOR DIAMETER

SELECTED RHO FOR INTERIOR DIAMETER

SELECTED INTERIOR DIAMETER - RHO LOWER BOUND

SELECTED INTERIOR DIAMETER - RHO UPPER BOUND

SELECTED INTERIOR DIAMETER - THETA LOWER BOUND

SELECTED INTERIOR DIAMETER - THETA UPPER BOUND

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

CONTROLS DUMP TO SCREEN/PRINTER

WILLENBOURG’S RETARDATION MODEL CONTROLLER (1=INCLUDE, 0=EXCLUDE)

FORMAN EQUATION REGRESSION OPTION

CRITICAL STRESS INTENSITY FACTOR \( K_c \)

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

STRESS INTENSITY FACTOR CALCULATION ACCURACY

SIF CALCULATION ACCURACY UNIFORM DISTRIBUTION LOWER BOUND

SIF CALCULATION ACCURACY UNIFORM DISTRIBUTION UPPER BOUND

MAXIMUM SIF

EFFECTIVE MAXIMUM SIF AFTER RETARDATION

MINIMUM SIF

EFFECTIVE MINIMUM SIF AFTER RETARDATION

STRESS CONCENTRATION DUE TO WELD OFFSET

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

GROWTH CALCULATION ACCURACY FACTOR

GROWTH CALCULATION ACCURACY UNIFORM DISTRIBUTION LOWER BOUND

GROWTH CALCULATION ACCURACY UNIFORM DISTRIBUTION UPPER BOUND

CRITICAL SIF \( K_c \) UNCERTAINTY

CRITICAL SIF \( K_c \) UNCERTAINTY UNIFORM DISTRIBUTION LOWER BOUND

CRITICAL SIF \( K_c \) UNCERTAINTY UNIFORM DISTRIBUTION UPPER BOUND

THRESHOLD SIF \( K_{th} \) UNCERTAINTY

THRESHOLD SIF \( K_{th} \) UNCERTAINTY UNIFORM DISTRIBUTION LOWER BOUND

THRESHOLD SIF \( K_{th} \) UNCERTAINTY UNIFORM DISTRIBUTION UPPER BOUND

SELECTED LAMBDA FOR ONE SIGMA NARROW-BAND RANDOM LOADS

LAMBDA FOR NARROW-BAND RANDOM LOADS -- LOWER BOUND OF \( k \)

LAMBDA FOR NARROW-BAND RANDOM LOADS -- UPPER BOUND OF \( k \)

NARROW-BAND RANDOM LOADS STRAIN GAGE ACCURACY FACTOR

DUE TO SAMPLE SIZE

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

STANDARD DEVIATION OF LAMBDA FOR NARROW-BAND RANDOM LOADS
(SIGMA, NORMAL DISTRIBUTION)

SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS

LAMBS

LAMSA LAMBDA FOR SUPERIMPOSED SINE LOADS -- LOWER BOUND OF \( k \)

LAMSB LAMBDA FOR SUPERIMPOSED SINE LOADS -- UPPER BOUND OF \( k \)

LAMSC LAMBDA FOR SUPERIMPOSED SINE LOADS COEFFICIENT OF VARIATION

LAMND SUPERIMPOSED SINE LOADS STRAIN GAGE ACCURACY FACTOR

LAMSK LAMBDA FOR SUPERIMPOSED SINE LOADS \( k \) -- INDICATES VARIATION

DUE TO SAMPLE SIZE

LAMSMU MEAN OF LAMBDA FOR SUPERIMPOSED SINE LOADS (MU, NORMAL DISTRIBUTION)

LAMSSG STANDARD DEVIATION OF LAMBDA FOR SUPERIMPOSED SINE LOADS

LAMW (SIGMA, NORMAL DISTRIBUTION)

SELECTED ACCURACY FACTOR FOR WELD ECCENTRICITY STRESS CONCENTRATION FACTOR, \( k \)

LAMNH LAMW LOWER BOUND

LAMNW LAMW UPPER BOUND

LDNAME() 1-D ARRAY CONTAINING LOAD NAMES FOR THE TIME-VARYING LOADS

LIFE() 1-ARRAY CONTAINING VALUES OF THE LIVES GENERATED -- SORTED VALUES OF THE LEFT-HAND TAIL

LOCAT LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE DUCT, AND 2 IS THE INTERIOR SURFACE OF THE DUCT

M() 2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- \( M(1,*) \)

M(2,*) ARE THE M3 LOADS

MAXBLF MAXIMUM NUMBER OF PERCENTAGE PROBABILITY LEVELS

MAXDAT MAXIMUM NUMBER OF POINTS PER DATA DIVISION ALLOWED

MAXDIV MAXIMUM NUMBER OF DATA DIVISIONS ALLOWED

MAXLD MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED

MAXLIF MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY

MAXSEG MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN CURVE)

MAXM MOMENT OF INERTIA FOR DUCT

MLAM() 2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS SCALED BY DSTR OR ASTR AND LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY TYPE()) -- MLM(1,*) ARE THE M2 LOADS; MLM(2,*) ARE THE M3 LOADS

MSLAM() 1-D ARRAY CONTAINING THE STATIC LOADS SCALED BY ASTR AND AERS OR SSTR AS APPROPRIATE -- MSLAM(1) IS THE M2 LOAD; MSLAM(2) IS THE M3 LOAD

MSTAT() 1-D ARRAY CONTAINING THE STATIC LOADS -- MSTAT(1) IS THE M2 LOAD; MSTAT(2) IS THE M3 LOAD

MVAR SELECTED FORMAN COEFFICIENT \( m \)

MVARA FORMAN COEFFICIENT \( m \) UNIFORM DISTRIBUTION LOWER BOUND

MVARB FORMAN COEFFICIENT \( m \) UNIFORM DISTRIBUTION UPPER BOUND

NBIN(100) 1-D ARRAY CONTAINING THE NUMBER OF CYCLES AFTER RF COUNTING

NBLIFE NUMBER OF BLIVES TO BE CALCULATED

NCLRL NUMBER OF CRACK LENGTHS FROM \( A1 \) TO \( A2 \) TO DO GROWTH INTEGRATION

NDIR NUMBER OF DEGREES OF FREEDOM FOR CRACK GROWTH (1 OR 2)

NDIV NUMBER OF DIVISIONS OF GROWTH RATE DATA

NEUB SELECTED NEUBER'S RULE MODEL ACCURACY FACTOR

NEUBA NEUB UNIFORM DISTRIBUTION LOWER BOUND

NEUBB NEUB UNIFORM DISTRIBUTION UPPER BOUND

NEWLIF LIFE VALUE RETURNED FROM CALL TO LIFCAL

NHYPER NUMBER OF SETS OF HYPERPARAMETER DISTRIBUTIONS TO BE SAMPLED FROM

NLIFE NUMBER OF DUCT FAILURE LIVES TO BE CALCULATED

NLIFET TOTAL NUMBER OF LIVES CALCULATED

NLOAD NUMBER OF TIME-VARYING LOADS

NORM RANDOM VARIABLE (SOMETIMES UNIFORM, SOMETIMES NORMAL) USED TO OBTAIN SELECTED TEMPERATURES AND PRESSURE

NP() 1-D ARRAY CONTAINING NUMBER OF POINTS PER DATA DIVISION FOR CRACK GROWTH RATE DATA

NRRAN NUMBER OF POINTS IN STRESS-TIME HISTORY

NU POISSON'S RATIO

NUMSEG NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE

P() 1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS

PC SELECTED INTERNAL PRESSURE, PSI

PCMU SELECTED MEAN OF NORMALLY DISTRIBUTED INTERNAL PRESSURE

PCMUA MEAN OF INTERNAL PRESSURE LOWER BOUND

PCMUB MEAN OF INTERNAL PRESSURE UPPER BOUND

PCO EXTERNAL PRESSURE, PSI

PCS1G SELECTED STANDARD DEVIATION OF NORMALLY DISTRIBUTED INTERNA PRESSURE

PCS1GA STANDARD DEVIATION OF INTERNAL PRESSURE LOWER BOUND

PCS1GB STANDARD DEVIATION OF INTERNAL PRESSURE UPPER BOUND

PFE COEFFICIENT "p" IN THE GENERALIZED FORMAN MODEL
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

INDICATED BY TYPE()

COEFFICIENT "g" IN THE GENERALIZED FORMAN MODEL

THE STRESS-STRAIN VS STRAIN CURVE

PRINCIPAL STRESS HISTORY (MAXM)

SSTR UNIFORM DISTRIBUTION LOWER BOUND

SSTRB UNIFORM DISTRIBUTION UPPER BOUND

1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES

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

2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING

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

STATIC SIGMAX STRESS COMPONENT

STATIC SIGMAXY STRESS COMPONENT

STATIC 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

STATIC SIGMAY STRESS COMPONENT

STATIC SIGMAYZ STRESS COMPONENT

STATIC SIGMAZ STRESS COMPONENT

I-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS

UNIFORM(0,1) RANDOM VARIATE USED TO DETERMINE HYPERDISTRIBUTION TO SELECT FROM

SELECTED WALL THICKNESS AT WELD, IN

WALL THICKNESS LOWER BOUND

WALL THICKNESS UPPER BOUND

SELECTED RHO FOR WALL THICKNESS

WALL THICKNESS - RHO LOWER Bound

WALL THICKNESS - RHO UPPER Bound

SELECTED THETA FOR WALL THICKNESS

WALL THICKNESS - THETA LOWER Bound

WALL THICKNESS - THETA UPPER Bound

SELECTED INNER WALL SURFACE TEMPERATURE (RANKINE)

SELECTED MEAN OF INNER WALL TEMPERATURE

NORMAL DISTRIBUTION, HYPER-

SELECTED MEAN OF INNER WALL TEMPERATURE LOWER BOUND

SELECTED STD DEVIATION OF INNER WALL TEMPERATURE

SELECTED STD DEVIATION OF INNER WALL TEMPERATURE LOWER BOUND

SELECTED OUTER WALL SURFACE TEMPERATURE (RANKINE)

SELECTED MEAN OF OUTER WALL TEMPERATURE

NORMAL DISTRIBUTION, HYPER-

SELECTED MEAN OF OUTER WALL TEMPERATURE LOWER BOUND

SELECTED STD DEVIATION OF OUTER WALL TEMPERATURE

SELECTED STD DEVIATION OF OUTER WALL TEMPERATURE LOWER BOUND

SELECTED STD DEVIATION OF OUTER WALL TEMPERATURE UPPER Bound

SELECTED SIGMA LOADS

VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY

1-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 AERODYNAMIC
V() 2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(I,*) 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
WITHA WIDTH LOWER BOUND
WITHB WIDTH UPPER BOUND
WITHR SELECTED RHO FOR WIDTH  BETA HYPER
WITHRI WIDTH - RHO LOWER BOUND DISTRIBUTION
WITHRT SELECTED THETA FOR WIDTH
WITHTI WIDTH - THETA LOWER BOUND
WITHTT WIDTH - THETA UPPER BOUND
WOFF SELECTED WELD OFFSET (%)
WOFFA WELD OFFSET LOWER BOUND - HYPERDISTRIBUTION 1
WOFFB WELD OFFSET UPPER BOUND - HYPERDISTRIBUTION 1
WOFFC WELD OFFSET LOWER BOUND - HYPERDISTRIBUTION 2
WOFFD WELD OFFSET UPPER BOUND - HYPERDISTRIBUTION 2
WOFFE PERCENTAGE OCCURRING IN HYPERDISTRIBUTION 1
WOFFH SELECTED WELD OFFSET UPPER BOUND
WOFFL SELECTED WELD OFFSET LOWER BOUND
WOFFR SELECTED RHO FOR WELD OFFSET
WOFFR1 WELD OFFSET - RHO LOWER BOUND - HYPERDISTRIBUTION 1  BETA
WOFFR2 WELD OFFSET - RHO LOWER 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 LOWER BOUND - HYPERDISTRIBUTION 2
WOFFT3 WELD OFFSET - THETA LOWER BOUND - HYPERDISTRIBUTION 2
WOFFT4 WELD OFFSET - THETA UPPER BOUND - HYPERDISTRIBUTION 2

OPEN THE INPUT AND OUTPUT FILES

C OPEN (1, FILE = 'CRRDAT', STATUS = 'OLD')
C OPEN (3, FILE = 'CRRKRES', STATUS = 'NEW')
C OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
C OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')
C
C SET DEFAULT VALUES
C CALL SETDEF (LIFE, NCRL)
C
C READ AND ECHO GENERAL DATA
C CALL INPUT (ANGLE, BLFPER, COEXP, E, EM,
& LOCAT, NBLIFE, NHYPER, NLIFE, NLIFET, NRAN,
& NU, NUMSEG, PERIOD, RSO, SE, STRHIS, TRUNC)
C
C READ MATERIAL PROPERTIES AND PERFORM REGRESSION ON CRACK GROWTH DATA
C CALL GRODAT (CEE, CO, DEE, DKTHO, EMM, ENN, FTY, KC, PEE, QUE)
C
C FOR HEX COIL GROW CRACK IN TWO DIRECTIONS BUT FOR EHEX ONLY ONE
C IF(KPROB .EQ. 1) THEN
C NDIR = 2
C ELSE
C NDIR = 1
C ENDFI
C
C >>> THIS LOOP SAMPLES HYPERPARAMETER SETS <<<<
C DO 300 K = 1, NHYPER
C CALL HYDRW (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
  CALL PARDRW (AI, CI, DLTAT, INDIA, LAMN, LAMS, PC,
                 THIC, WIDTH, WOFF)
  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, NRAN, NU, 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
    DO 400 I = 1, NUM
      WRITE(3,1100) LIFE(I)
      WRITE(3,1200) LIFE(I)/FLOAT(NLIFET), LIFE(I)
  ENDIF
400 CONTINUE
  WRITE(3,1300)
  DO 500 J = 1, NBLIFE
    BLFPOS = NINT (BLFPER(J) * FLOAT (NLIFET))
    WRITE(3,1400) BLFPER(J), LIFE(BLFPOS)
  ENDIF
500 CONTINUE
  ELSE
    WRITE(3,1500) NEWLIF
  ENDIF
STOP

------------------------------------------------------------------- FORMATS -------------------------------------------------------------------
1000 FORMAT(///,30X,'SIMULATION OUTPUT',///)
1100 FORMAT(20X,E12.5)
1200 FORMAT(///,20X,'SHORTEST 1% OF CRACK GROWTH LIVES ',///,
        20X,'LIFE ',/)
1300 FORMAT(///,2X,'B LIVES:   EMPIRICAL',/)
1400 FORMAT(2X,F7.5,5X,E12.5)
1500 FORMAT(13X,'CRACK GROWTH LIFE = ',E12.5)
END

************************************************************************************
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************************************************************************************
& EMM, ENN, FTY, INDIA, KC, KLAM, LAMGR, LAMKC,
& LAMKH, LAMN, LAMS, LAMW, LOCAT, NEUB, NEWLIF,
& NDIR, NCRL, NRAN, NU, 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, PDCDB, RATIO, SM, SPR(MAXM), TOTLIF

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,
& LAMGR, LAMKC, LAMKH, LAMN, LAMS, LAMW, M(2,MAXLD),
& MSTAT(2), NEUB, NEWLIF NU, P(MAXLD), PC, PCO, PEE,
& PERIOD, PSTAT, QUE, RSO, SE(MAXSEG), SSTR,
& STRHIS(MAXLD,MAXM),
& SX(MAXLD), SXST, SY(MAXLD), SYST, SZ(MAXLD),
& SXST, SYST, SYST, SXST, SYZST, SXY(MAXLD),
& SXYST, SXZ(MAXLD), SXZST, SYZ(MAXLD), SYZST, SXY,
& SX, SY, SZ, SYX, SXZ, SYZ

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, SXST, SYZST,
& SX, SY, SZ, SKX, SXZ, SYZ

COMMON IOUT

C PERFORM LOAD TO STRESS TRANSFORMATION

IF (KPROB .EQ. 1) THEN
  AF = THIC
  CALL STRAN1 (AERD, AERS, ASTR, AI, ANGLE, CI, CEE, CO,
& INDIA, LAMN, LAMS, LAMW, LOCAT, NRAN, NU, PC,
& SPR, STRHIS, THIC, WOFF)
ELSEIF (KPROB .EQ. 2) THEN
  AF = WIDTH/2.0
  CALL STRAN2 (DSTR, 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)
TOLIF = 0.0
FAIL = .FALSE.
JLAST = 1

IF (NDIR .EQ. 2) THEN
   A(2) = CI
   PDADB = 0.0
   PDCDB = 0.0
ENDIF

C CALCULATE CRACK-GROWTH LIFE FOR THE LOAD BLOCK
C >>>> THIS LOOP IS FOR EVERY CRACK LENGTH
C
DO 100 J = 1, NCRL
   CALL BLKGRO (A, CEE, CO, DADB, DEE, DKTHO, DALT, EMM, ENN, FAIL, PGT, INDIA, KC, KLAM, & LAMKC, LAMKH, NBMN, 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)
      TOLIF = 2.0*DELA/(DADB(1) + PDADB) + TOLIF
   ELSEIF (PDCDB .GT. 0.0) THEN
      TOLIF = 2.0*DELC/(DADB(2) + PDCDB) + TOLIF
   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 110
      ELSEIF (NDIR .EQ. 2 .AND. A(2) .LT. WIDTH/2.0) 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
   ELSE
      IF (FAIL) THEN
         FAIL = .FALSE.
         WRITE(8,*)'K GT Kcr AT A = ',A(1)
         GO TO 110
      ELSE
         IF (NDIR .EQ. 1 .OR. A(2) .GT. WIDTH/2.0 .OR. DADB(2) .EQ. 0.0) THEN
            TOLIF = 1.0E+37
            WRITE(8,*)'NO GROWTH AT', J, 'th CRACK LENGTH'
            GO TO 110
         ENDIF
         IF (NDIR .EQ. 2 .AND. A(2) .LT. WIDTH/2.0) 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
   ENDIF

100 CONTINUE

END
ENDIF
WRITE(8,*) 'NO GROWTH IN A DIRECTION AT, J,'th CRACK LENGTH'
ENDIF
PDADB = DADB(1)
PDADB = DADB(2)
CC
WRITE(8,*) A(1), A(2), DELA, DELC, TOTLIF
100 CONTINUE
C CALCULATE LIFE
110 CONTINUE
NEWLIF = LAMGR * PERIOD * TOTLIF
RETURN
END
C**************************************************************
SUBROUTINE BLKGRO CALCULATES THE CRACK GROWTH RATE PER BLOCK
C PROGRAMMER : S. SUTHARSHA
C DATE : DECEMBER 1992
C VERSION: 92.5
C Copyright (C) 1991, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.
C**************************************************************
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 IMPLICIT NONE
INTEGER I, IDIR, IOUT, NBIN(100), NDIR
INTEGER INEUB, IRET, KGROW, KPROB
REAL A(2), AB(2), AO(2), AORPO, AORPA, ARPI, CEE, 
& CO, CONST, DA, DADB(2), DEE, DK, &
& DKEFF, DKTH, DKTHO, DSALT, EMM, ENN, &
& F0(2), F2(2), FTY, INDIA, KC, KCR, KLAM, &
& KMAX(2), KMAXE, KMAXRQ, KMIN(2), KMINF, LAMKC, LAMKH, &
& PI, PLSR(2), PEE, QUE, REFF, RPI, &
& RPO(2), RSO, SALMAX, SALTF, SM, THIC, WIDTH
C
C*************************************************************
C DESCRIPTION OF LOCAL VARIABLES =_---------=====_
C AB() CRACK LENGTHS DURING GROWTH IN THE BLOCK
C AO() CRACK LENGTHS AT THE LAST OVERLOAD
C AORPO AORPO - AB
C ARPI AORPO - AB
C F0() SIF COEFF FOR TENSILE STRESS
C F2() SIF COEFF FOR BENDING STRESS
C KMAXQ REQUIRED SIF FOR WILLENBOURG MODEL
C PLSR() PLANE STRAIN/STRESS PLASTIC ZONE SIZE COEFF
C RPI CURRENT PLASTIC ZONE SIZE
C RPO() OVERLOAD PLASTIC ZONE SIZE
C*************************************************************
LOGICAL FAIL
COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
COMMON IOUT
DATA PI/3.14159265358979/, PLSR/0.053051647, 0.159154943/
AB(1) = A(1)
AB(2) = A(2)
AO(1) = 0.0
AO(2) = 0.0
RPO(1) = 0.0
7 - 34
RPO(2) = 0.0

IF(IOUT .EQ. 20) THEN
  WRITE(8,*),'INSIDE BLKGRO ROUTINE'
  WRITE(8,*),'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(1), A(2), F0, F2, INDIA, THIC)
ELSEIF(KPROB .EQ. 2) THEN
  CALL STRIF2(A(1), 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*101.0
DO 200 I = 1, 100
  IF (NBIN(I) .GT. 0) THEN
    SALTF = SALMAX - FLOAT(I)*DSALT
    DO 100 IDIR=I,NDIR
      KMAX(IDIR) = CONST*(F0(IDIR)*SALTF + F2(IDIR)*SM)
      KMIN(IDIR) = CONST*(-F0(IDIR)*SALTF + F2(IDIR)*SM)
    DO
    IF(IOUT .EQ. 20) THEN
      WRITE(8,*), DIRECTION
      WRITE(8,*) DIR, KMAX 1,2' IDIR, KMIN 1,2' IDIR, KMAX(IDIR)& KMIN ( IDIR )
    ENDIF
    C IF MAXIMUM SIF IS NEGATIVE OR ZERO NO GROWTH IN THIS DIRECTION
    IF (KMAX(IDIR).LE.0.0) THEN
      GO TO 95
     ENDIF
    C
    C RESET MINIMUM SIF TO ZERO IF NEGATIVE
    IF(KMIN(IDIR).LE.0.0) THEN
      KMIN(IDIR) = 0.0
     ENDIF
    DKEFF = KMAX(IDIR) - KMIN(IDIR)
    REFF = KMIN(IDIR)/KMAX(IDIR)
    KMAXEF = KMAX(IDIR)
    IF (IRET .EQ. 1 ) THEN
      AORPO = AO(IDIR) + RPO(IDIR)
      RPI = PLSR(IDIR) * (KMAX(IDIR)/FTY)**2
      ARPI = AB(IDIR) + RPI
      IF(ARPI .GT. AORPO) THEN
        RPO(IDIR) = RPI
      ENDIF
      AO(IDIR) = AB(IDIR)
    ELSE
      AORPA = AORPO - AB(IDIR)
      KMAXRQ = FTY*( AORPA/PLSR(IDIR))**0.5
      KMAXEF = KMAX(IDIR) - (KMAXRQ - KMAX(IDIR))/(RSO-1.0)
      KMINEF = KMIN(IDIR) - (KMAXRQ - KMAX(IDIR))/(RSO-1.0)
      IF( KMAXEF.GT.0.0 ) THEN
        IF( KMINEF.GT.0.0 ) THEN
          DKEFF = KMAXEF - KMINEF
          REFF = KMINEF/KMAXEF
        ELSE
          DKEFF = KMAXEF
          REFF = 0.0
        ENDIF
      ELSE
        DKEFF = KMAXEF
        REFF = 0.0
      ENDIF
    ENDIF

7 - 35
END
ENDIF
IF (IOUT .EQ. 20) THEN
WRITE(8,*) 'AORPA, AORPO, KMAXRQ, RPI, RPO, AO',
& AORPA, AORPO, KMAXRQ, RPI, RPO(IDIR),
& AO(IDIR)
ENDIF
ENDIF

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 *((1.0 - REFF)**EMM) * (DK)**ENN
& / ( (1.0 - REFF) * KCR - DK )**QUE)
DADB(IDIR) = DADB(IDIR) + DA
IF (IOUT .EQ. 20) THEN
WRITE(8,*) 'DIR, DA, DADB, AB ', I, IDIR, DADB, AB
ENDIF
ELSEIF (KMAXEF .GT. KCR ) THEN
FAIL = . TRUE.
RETURN
ENDIF

95 CONTINUE
100 CONTINUE
ENDIF
200 CONTINUE

RETURN
END

C===================================================================================================
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
C COMMON IOUT

7 - 36
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( )
TEMP( ) 1-D ARRAY CONTAINING VALUES OF LIFE( ) TO BE SHIFTED UPON INSERTION OF NEWLIF

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

***********************************************************************
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.
***********************************************************************
SUBROUTINE PRYRV (RAND, RHOL, 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
Y = FRAC * (THE2 - THE1) + THE1
IF (IOUT .EQ. 15) WRITE(8,*),'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
VERSION: MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
V4.3, V4.4, V4.5
C*******************************************************************************

SUBROUTINE RANDOM (FRAC, RAND)
C IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL FRAC
DOUBLE PRECISION RANA, RANC, RAND, RANDIV, RANM, RANSUB,
& RANT, RANX
C
C LIST OF VARIABLES
C
C FRAC UNIFORM (0,1) RANDOM VARIATE
C IOUT OUTPUT DUMP CONTROLLER
C RANA CONSTANT FOR LCG
C RANC CONSTANT FOR LCG
C RAND RANDOM NUMBER SEED
C RANDIV INTERNAL CALCULATION
C RANM CONSTANT FOR LCG
C RANSUB INTERNAL CALCULATION
C RANT INTERNAL CALCULATION
C RANX INTERNAL CALCULATION
C
C USING LCG RANDOM # GENERATOR
RANA = 671093.0
RANC = 7090885.0
RANM = 33554432.0
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

The random variates are generated using the "Direct Method"
Abrasowitz, M., and Stegun, I. A., editors, Handbook of
Mathematical Functions, National Bureau of Standards, Applied
Mathematics Series 55, Issued June 1964, Ninth Printing, November
1970 with corrections pg. 953.

********************************************************************************

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. 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 M4LI 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 M4LI (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 INPUTS: ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT, NLOAD, NU, P,
& PC, PCO, PSTAT, STATIC, T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT
C OUTPUTS: STATIC, STRAMP
C IMPLICIT NONE
COMMON I OUT
INTEGER I, IOUT, J, MAXLD, NLOAD
REAL PI
PARAMETER (MAXLD = 16, PI = 3.1415926536)
REAL ALPHA, ANGLE, AREA, DI, DLTAT, EM, FK, K(2, 2), KOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU,
& P(MAXLD), PC, PCO, PSTAT, RDIFF, RI, R12, RO, RO2,
& ROT, RT(10), SIG1A(MAXLD), SIG1B, SIG2A, SIG2B, SIGT1, SIGT2,
& STATIC, SIGT3, SIGT4, SIGT5, SIGT6, SIGT7, STR1A, STR1B, STR2A, STR2B, STRIC,
& STRAMP, 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  STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
LAMW  ACCURACY FACTOR OF FK - r/t CURVE
M(1)  2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
M(2)  ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
MAXLD  MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
M1  MOMENT OF INERTIA
MSTAT()  1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS 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
PCO  LIMIT PRESSURE ON INSIDE OF THE VESSEL
P1  SELF EXPLANATORY CONSTANT
PSTAT  STATIC AXIAL LOAD
RDIFF  EQUAL TO R02 - RI2
RI  INTERIOR RADIUS
RI2  INNER RADIUS SQUARED
RO  OUTER RADIUS
RO2  OUTER RADIUS SQUARED
ROF  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
       FOR THE TIME-VARYING LOADS
SIGIA()  1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO FORCE
       FOR THE TIME-VARYING LOADS
SIGIB()  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
STR1A  THE STATIC AXIAL STRESS DUE TO FORCE
STR1B  THE STATIC AXIAL STRESS DUE TO BENDING
STR1C  THE STATIC AXIAL STRESS DUE TO MOMENTUM CHANGE (FLUID)
STR2A  THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO INTERNAL PRESSURE
STR2B  THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO EXTERNAL PRESSURE
STRAMP()  2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
          -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
          THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
          STRAMP(4,*) ARE THE SHEAR STRESSES
T()  1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
THIC  WALL THICKNESS AT DUCT OUTER RADIUS
TSTAT  STATIC TORQUE LOAD
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 -- V(1) IS THE V2
        LOAD; V(2) IS THE V3 LOAD
WOFF  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
    INTERPOLATE TO FIND FACTOR FK CORRESPONDING TO VALUE OF r/t
    IF ((ROT .LE. RT(J)) .AND. (ROT .GE. RT(J-1))) THEN
        FK = FK(J) - FK(J-1) * (ROT - RT(J-1)) / (RT(J) - RT(J-1))
    ENDIF
50 CONTINUE

KOFF = LAMW * (1.0 + 3.0 * IFK * WOFF)

IF (IOUT .EQ. 25) THEN
    WRITE(8,*), 'DI = ', DI, ' RI = ', RI
    WRITE(8,*), 'THIC = ', THIC, ' WOFF = ', WOFF
ENDIF

7-41
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 RADIi 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
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
STRI B = (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,*) '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,*) 'STR2A = ', 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
ENDIF
DO 100 I = 1, NLOAD
C AXIAL STRESS CALCULATIONS
SIGIA(I) = P(I) / AREA
SIGIB(I) = (M(1,I) * COS(ANGLE) + M(2,I) * SIN(ANGLE)) * RO / MI
& STRAMP(1,I) = (SIGIA(I) + SIGIB(I)) * SKTI * KOFF
C HOOP (2) AND RADIAL (3) STRESSES ARE ZERO
BECAUSE PRESSURES ARE CONSTANT
STRAMP(2,I) = 0.0
STRAMP(3,I) = 0.0
C SHEAR STRESS
STRAMP(4,I) = T(I) * RO / (2.0 * MI) - (2.0 / AREA)
& * (V(1,I) * COS(ANGLE) + V(2,I) * SIN(ANGLE))
IF (IOUT.EQ.25) THEN
WRITE(8,*) 'STRESS VALUES FOR I = ',I
WRITE(8,*) 'AXIAL STRESSES'
WRITE(8,*) 'SIGIA = ', SIGIA(I), ' SIGIB = ', SIGIB(I)
WRITE(8,*) 'STRAMP(I,I) = ', STRAMP(I,I)
WRITE(8,*) 'HOOP STRESSES', ' STRAMP(2,I) = ',STRAMP(2,I)
WRITE(8,*) 'RADIAL STRESS', ' STRAMP(3,I) = ',STRAMP(3,I)
WRITE(8,*) 'SHEAR STRESS', ' STRAMP(4,I) = ',STRAMP(4,I)
ENDIF
100 CONTINUE

IF (IOUT.EQ.25) THEN
WRITE(8,*) 'I AXIAL HOOP RADIAL SHEAR'
WRITE(8,*) STATIC(I), STATIC(2), STATIC(3), STATIC(4)
DO 300 I = 1, NLOAD
WRITE(8,*) I, STRAMP(I,I), STRAMP(2,I), STRAMP(3,I), STRAMP(4,I)
300 CONTINUE
ENDIF
RETURN
END

***************************************************************************
SUBROUTINE M4L2 (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)
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
SUBROUTINE M4L2 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
FOR LOCATION 2 (PLAIN WELD, INTERIOR SURFACE OF THE DUCT), UNDER
THERMAL LOADING
PROGRAMMER: L. NEWLIN
DATE: JUL92
VERSION: 92.4
***************************************************************************
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) IS 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
PCO     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 - RI2
RI      INTERIOR RADIUS
RI2     INNER RADIUS SQUARED
RO2     OUTER RADIUS SQUARED
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
STR1A   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(1,*') 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
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 -- 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
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 .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 RADII APPROPRIATE TO LOCATION
THIS IS THE INTERIOR SURFACE
SKTI = 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
RDIF2 = RO2 - RI2
GEOM = 1.00 - 2.00 * LOG (RO / RI) * RO2 / RDIF2
TEMPERATURE STRESS
STHMA = ((EM * ALPHA * DLTAT) / (2.00 * (1.00 - NU)))
\[ \text{STATIC}(1) = (\text{STRIA} + \text{STRIB} + \text{STRIC}) \times \text{SKT1} \times \text{KOFF} + \text{STHMA} \]

**HOOP (2) AND RADIAL (3) STRESS CALCULATIONS**

\[ \text{STR2A} = \text{PC} \times (\text{RI}^2 + \text{R0}^2) / \text{RDIFF} \]
\[ \text{STR2B} = -2.0 \times \text{PCO} \times \text{R0}^2 / \text{RDIFF} \]

\[ \text{STATIC}(2) = (\text{STR2A} + \text{STR2B}) \times \text{SKT2} + \text{STHMA} \]

\[ \text{STATIC}(3) = -\text{PC} \]

**SHEAR STRESS**

\[ \text{STATIC}(4) = \text{TSTAT} \times \text{RI} / (2.0 \times \text{MI}) - (2.0 / \text{AREA} \times \text{VSTAT}(1) \times \text{COS (ANGLE)} + \text{VSTAT}(2) \times \text{SIN (ANGLE)}) \]

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,*), 'SIGIA = ', SIGIA(I), 'SIGIB = ', SIGIB(I)
WRITE(8,*), 'STRAMP(I,I) = ', STRAMP(I,I)
WRITE(8,*), 'STHMA = ', STHMA(I)
WRITE(8,*), 'RADIAL STRESS', STRAMP(3,I) = 0.0
WRITE(8,*), 'SHEAR STRESS', STRAMP(4,I) = T(I) / (2.0 * MI)
ENDIF

DO 100 I = 1, NLOAD

**AXIAL STRESS CALCULATIONS**

\[ \text{SIGIA}(I) = \text{P}(I) \times \text{AREA} \]
\[ \text{SIGIB}(I) = (\text{M}(I,1) \times \text{COS (ANGLE)} + \text{M}(2,I) \times \text{SIN (ANGLE)}) \times \text{RI} / \text{MI} \]

\[ \text{STRAMP}(1,I) = (\text{SIGIA}(I) + \text{SIGIB}(I)) \times \text{SKT1} \times \text{KOFF} \]

**HOOP (2) AND RADIAL (3) STRESSES ARE ZERO**

\[ \text{STRAMP}(2,I) = 0.0 \]
\[ \text{STRAMP}(3,I) = 0.0 \]

**SHEAR STRESS**

\[ \text{STRAMP}(4,I) = \text{T}(I) \times \text{RI} / (2.0 \times \text{MI}) - (2.0 / \text{AREA} \times \text{V}(1,I) \times \text{COS (ANGLE)} + \text{V}(2,I) \times \text{SIN (ANGLE)}) \]

IF (IOUT.EQ.25) THEN

WRITE(8,*), 'STRESS VALUES FOR I = ', I
WRITE(8,*), 'AXIAL STRESSES'
WRITE(8,*), 'SIGIA = ', SIGIA(I), 'SIGIB = ', SIGIB(I)
WRITE(8,*), 'STRAMP(1,I) = ', STRAMP(1,I)
WRITE(8,*), 'HOOP STRESSES', STRAMP(2,I) = 0.0
WRITE(8,*), 'RADIAL STRESS', STRAMP(3,I) = 0.0
WRITE(8,*), 'SHEAR STRESS', STRAMP(4,I) = 0.0
WRITE(8,*)

7 - 46
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 Calculates the number of cycles by rainflow counting, creates a stress vs. cycles table, and determines the equivalent mean stress
PROGRAMMER: S. Sutharshana, L. Newlin
DATE: December 1992
VERSION: 92.5
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

LIST OF VARIABLES

input variables:

E() 1-D array containing the strain values
EM() Young's modulus before yield
IOUT output dump controller
M total number of stress data points per period
MAXM maximum number of points in stress time history
MAXSEG maximum number of segments allowed
NUMREG number of regions of interest
NUMSEG number of segments of interest
SE() 1-D array containing the stress-strain products
C SPR(M)   PRINCIPAL STRESS HISTORY
C TRUNC   VALUE USED TO FILTER OUT NOISE
C intermediate variables:
C EE()    HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
C I,J,K   COUNTERS FOR EFFECTIVE STRESSES
C JMAX    INDEX (LOCATION) OF SEFMAX IN SPR()
C N       NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
C NEUB    NEUBER'S RULE MODEL ACCURACY FACTOR
C NEWTOT  FUNCTION TO CALCULATE EQUIVALENT MEAN STRESS
C OVER    OVER FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
C S(NEXTOT) FILTERED EFFECTIVE STRESSES
C SALTF   ALTERNATING STRESS FOR LARGEST STRESS RANGE CYCLE
C SEFMAX  LARGEST EFFECTIVE STRESS
C SM      SM = EQUIVALENT MEAN STRESS
C SMEANF  MEAN STRESS FOR LARGEST STRESS RANGE CYCLE
C SP(M+1) RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
C TEST1() 1-D ARRAY USED IN FILTERING THE STRESSES
C TEST2() 1-D ARRAY USED IN FILTERING THE STRESSES
C dump input data
   if (IOUT.EQ.30) then
     WRITE(*,*)'cycoun inputs'
     WRITE(*,*)'m : ',M
     WRITE(*,150) 'EM : ',EM,' TRUNC : ',TRUNC,' NEUB : ',NEUB
   endif
C INITIALIZE ARRAYS
   DO 50 I = 1, MAXM
     SM = 0.0
     S(I) = 0.0
     EE(I) = 0.0
     INDEX(I) = 0
     TEST1(I) = 0.0
     TEST2(I) = 0.0
   CONTINUE
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
     ENDF
   CONTINUE
C assign all points from JMAX out, to the beginning of SP()
   DO 210 I = 1, M-JMAX+1
     J = JMAX-I + I
     SP(I) = SPR(J)
   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)
   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=l,m+l)
endif

C****************************** END RESQUENCE ******************************
C****************************** BEGIN FILTER ******************************
C Filter the resequenced effective stresses, leaving only peaks and valleys (excursions larger than TRUNC are deleted during rainflow counting) in (NEWTOT), where NEWTOT is the new number of points

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
K = INDEX(I)
S(I) = SP(K)
320 CONTINUE
if (iout.eq.30) 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 stress data, 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, 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
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=lon
      write(8,*) spr(i)
 12 continue
  endif
  IF (N .EQ. 0) THEN
    C TRUNCATION FILTER TOO LARGE -- NO CYCLES LEFT
    GOTO 710
  ENDIF

C************************ END RAINFLOW ***************************

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
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 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 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
NUMSEG TOTAL EQUIVALENT MEAN STRESS
OVER NUMBER OR SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
PRODCT STRESS STRAIN PRODUCT (WHERE PLASTIC=ELASTIC DEFORMATION)
SALT TOTAL ALTERNATING STRESS
SE() 1-DIMENSIONAL ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
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 .GT. SE(I)) THEN
IF (PRODCT .LT. SE(I+1)) THEN
EPSLON = E(I) + ((E(I+1) - E(I)) / (SE(I+1) - SE(I))) * (PRODCT - SE(I))
TEMP = PRODCT / EPSLON - SALT
ELSE
EPSLON = E(I) + ((E(I+1) - E(I)) / (SE(I+1) - SE(I))) * (PRODCT - SE(I))
TEMP = PRODCT / EPSLON - SALT
ENDIF
ENDIF
800 CONTINUE
ENDIF
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

SUBROUTINE HYPDRW (AERD, AERS, ASTR, DSTR, KLAM, LAMGR, LAMKC, LAMKH, LAMW, NEUB, SSTR, MVAR)

SUBPROGRAMS: PRYRV, RANDOM

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,
& AOCa, AOCb, AOc1, AOc2, AOcT1, AOcT2, AOcR, AOcT,
& ASTR, ASTRa, ASTRb,
& DPCM, DPCSiG, DSTR, DSTRA, DSTRb, DTMU, DTMiG,
& DTMU, DTMiG, DUM, DINDA, DINDb,
& KLAM, KLAMa, KLAMb, LAMKH, LAMKHb, LAMKHb,
& LAMKC, LAMKCA, LAMKCB, LAMGR, LAMGRA, LAMGRb,
& LAMNA, LAMNB, LAMNC, LAMND,
& LAMNK, LAMNMU, LAMNSG, LAMSA, LAMSB, LAMSC, LAMSD,
& LAMSK, LAMSMu, LAMSSG,
& LAMW, LAMWA, LAMWB, MVAR, MVARa, MVARB

REAL NEUB, NEUba, NEUbb, NORM, PCM, PCMUA, PCMUB, PCSiG,
& PCSiGa, PCSiGb, SSTR, SSTRA,
& SSTRb, TEST, THICA, THICb,
& THICK, THIC1, THIC2, THICt, THICT1, THICT2, TIMU,
& TIMU, TIMUb, TISiG, TISiGa, TISiGb, TOMU, TOMUA, TOMUB,
& TOSiG, TOSiGa, TOSiGb, WITHa.
& WIThB, WITHb1, WITHb2, WIThT1, WIThT2, WIThR, WIThT,
& WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO, WOFFR,
& WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFT, WOFFT1, WOFFT2,
& WOFFT3, WOFFT4

COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AIA,
& AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT,
& AOCR, AOCR2, AOCR1, AOCR2, AOCR1, AOCR, AOCR,
& ASTR, ASTRB,
& DPCMU, DPCSIG, DSTRA, DSTRB, DTMU, DTSIG, DTMU, DTSIG,
& INDIA, INDIA1, INDIA2, INDIA1, INDIA2, INDIA, INDIA1,
& KLANA, KLAB, LAMRA, LAMRB, LAMKH, LAMKB, LAMKCA, LAMKCB,
& LAMNA, LAMB, LAMNC, LAMND, LAMMU, LAMNSG,
& LAMSA, LAMSB, LAMSG, LAMSD, LAMSMU, LAMSSG,
& LAMWA, LAMWB, LAMWR, LAMWRB, LAMWRA, LAMWRC,
& LCDIA, LCDIA1, LCDIA2, LCDIA1, LCDIA2, LCDIA,
& LAMNCW, LAMNCW2, LAMNCW3, LAMNCW2, LAMNCW3, LAMNCW,
& LAMNDU, LAMNDW, LAMND, LAMND, LAMNDU, LAMNDW,
& LAMMUU, LAMMCS, LAMMSG, LAMMSG2, LAMMSG,
& LAMSDU, LAMSSG, LAMSSG2, LAMSSG,
& LAMWA, LAMWB, LAMWR, LAMWRB, LAMWRA, LAMWRC,
& PCMU, PCMB, PCSIG, PCSIGA, PCSIGB,
& RAND, SSTR, SSTRB,
& THICA, THICB, THIC1, THIC2, THIC1, THIC2, THIC,
& TIMU, TIMU1, TIMUB, TISIG, TISIGA, TISIGB,
& TOMU, TOMU1, TOMUB, TOSIG, TOSIGA, TOSIGB,
& WITHA, WITHB, WITH1, WITH2, WITH1, WITH2, WITHA, WITHB,
& WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
& WOFFR, WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR, WOFFT1, WOFFT2,
& WOFFT3, WOFFT4, WOFFT

COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
COMMON IOUT

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
ENDIF

IF (IOUT .EQ. 15) THEN
  WRITE(8,*), 'TEST =', TEST, ' WOFFE =', WOFFE
  WRITE(8,*), 'WOFFLO =', WOFFLO, ' WOFFHI =', WOFFHI
ENDIF

CALL PRYRV (RAND, INDIR1, IND12, INDIR1, INDIT2, INDIR, INDIT)
CALL PRYRV (RAND, THIC1, THIC2, THIC1, THIC2, THIC)
CALL PRYRV (RAND, AOCA, AOCR2, AOCR1, AOCR2, AOCA, AOCR)
CALL PRYRV (RAND, AERDA, AERDB, AERSA, AERSB, AERD, AERS)
CALL PRYRV (RAND, ASTR, ASTRB, DUM, DUM, ASTR, DUM)
ELSEIF (KPROB .EQ. 2) THEN
  CALL PRYRV (RAND, WITH1, WITH2, WITH1, WITH2, WITH1, WITH2,
& WITH1, WITH2)
ENDIF

CALL PRYRV (RAND, LAMN, LAMNB, LAMSA, LAMSB, LAMNK, LAMSK)
LAMMU = LAMND / (1.0 + LAMNK * LAMNC)
LAMNSG = LAMNC / (1.0 + LAMNK * LAMNC)
LAMM = LAMSD / (1.0 + LAMSK * LAMSC)
LAMSSG = LAMSC / (1.0 + LAMSK * LAMSC)

IF (IOUT .EQ. 15) THEN
  WRITE(8,*), 'LAMNK = ', LAMNK, ' LAMMMU = ', LAMMMU,
& LAMNSG = ', LAMNSG
  WRITE(8,*), 'LAMSK = ', LAMSK, ' LAMMSM = ', LAMMSM,
& LAMSSG = ', LAMSSG
ENDIF

IF (KPROB .EQ. 1) THEN
  CALL RANDOM (NORM, RAND)
  TIMU = TIMUA + NORM * DTMU
ENDIF
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, LAMKH, LAMKCA, LAMKCB, LAMKH, LAMKC)
CALL PRYRV (RAND, LAMWA, LAMWB, NEUBA, NEUBB, LAMN, NEUB)
CALL PRYRV (RAND, DSTRA, DSTRB, KLAM, KLAMB, DSTR, KLAM)
CALL PRYRV (RAND, SASTRA, SASTRB, DUM, DUM, SASTR, 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, LAMS, 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, AOC1, AOCA, AOCA1, AOCA2, &
AOCT1, AOCT2, AOCTR, AOCT, ASTR, ASTRB, &
CL, DLTAT, DPCMU, DPCSIG, DSTR, DSTRB, DTIMU, DTISIG, &
DTOMU, DTOSIG, INDIA, INDIAA, &
INDIA1, INDIR, INDIR1, INDIR2, INDIR1, INDIR2, &
KLAM, KLAMB, LAMKHA, LAMKH, &
LAMKCA, LAMKCB, LAMGR, LAMGRB, &
LAMN, LAMNB, LAMNC, LAMND, &
LAMMU, LAMMSG, LAMS, LAMS, LAMS, LAMS, LAMSC, LAMS, LAMS, &
LAMSH, LAMSHG, LAMSH, LAMSH, MVAR, MVARB
COMMON/DRIVRS/ AERDA, AERDB, AERSA, AERSB, AI, AIA, AIB, AIR1, AIR2, &
AIT1, AIT2, AIR, AIT, AOC, AOC1, AOCA, AOCA1, AOCA2, &
AOCT1, AOCT2, AOCTR, AOCT, ASTR, ASTRB, &
CL, DLTAT, DPCMU, DPCSIG, DSTR, DSTRB, DTIMU, DTISIG, &
DTOMU, DTOSIG, INDIA, INDIAA, &
INDIA1, INDIR, INDIR1, INDIR2, INDIR1, INDIR2, &
KLAM, KLAMB, LAMKHA, LAMKH, &
LAMKCA, LAMKCB, LAMGR, LAMGRB, &
LAMN, LAMNB, LAMNC, LAMND, &
LAMMU, LAMMSG, LAMS, LAMS, LAMS, LAMS, LAMSC, LAMS, &
LAMSH, LAMSHG, LAMSH, LAMSH, MVAR, MVARB

RETURN
END
COMMON/CNTRL/INEUB, IRET, KGROW, KPROB
COMMON IOUT

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, WITHR, 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)
  DLTAT = TIN - TOUT
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
  WRITE (8, *) 'WOFF =', WOFF
  WRITE (8, *) 'WIDTH =', WIDTH
ENDIF
RETURN
END

SUBROUTINE STRIFI (A, C, F0, F2, INDIA, THIC)

A - CRACK DEPTH, A
C - CRACK LENGTH, C

INTERNAL VARIABLES
AOC75 - EVENTUALLY, (A/T)**.75
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)  

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, HI, 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 = SQRT(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  
POWR = 1.0  
ENDIF

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*SQRT(SQCOA)  
H1 = 1.0 + AOT*(-0.4 - 0.41*COA + AOT*(0.55 +
& H2 = 1.0 + AOT*(-2.11 + 0.77*COA + AOT* 0.55 + AOC75*(-0.72 + AOC75*0.14) )
& Q = 1.0 + 1.464*COA**1.65
FPA = SQRT(COA)
PFC = 1.0
ENDIF

C 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', 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 GRODAT READS MATERIAL PROPERTIES AND PERFORMS REGRESSION ON
CRACK GROWTH DATA
PROGRAMMER : S. SUTHARSHANA
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), &
  NPR, NPRMN

REAL AA, BB, CC, CEE, DADN(MAXDIV, MAXDAT), &
  CO, DEE, DELK(MAXDIV, MAXDAT), DENOM, DETER4, &
  DIFFX1, DIFFX2, DIFFX3, DIFFX4, DIFFY, DKTH, DKTHO, &
  EMM, ENN, FTY, KC, LNCCE, MEANK1, MEANX2, MEANX3, MEANX4, &
  MEANY, PEE, QUE, RDATA(MAXDIV), RK
  SX1Y, SX2Y, SX3Y, SX1X1, SX1X2, SX1X3, SX2X2, SX2X3, &
  SX3X3, SX4Y, SXIX4, SX2X4, SX3X4, SX4X4, Y

C**********************************************************
C DESCRIPTION OF LOCAL VARIABLES
C**********************************************************

C DADN() CRACK GROWTH RATE
C DELK() SIF RANGE
C DENOM DENOMINATOR
C RDATA() STRESS RATIOS
C RK (1-R)KC

C 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
ENDIF

DO 180 J = 1, NP(I)
  READ(1,*) DADN(I,J), DELK(I,J)
  WRITE(3,6200) DADN(I,J), DELK(I,J)
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(DELX(J,K))
MEANX2 = MEANX2 + LOG10(1.-RDATA(J))
MEANX3 = MEANX3 - LOG10(RKC - DELK(J,K))
CONTINUE
250
NPR = NPR + NP(J)
275
CONTINUE

MEANY = MEANY/FLOAT(NPR)
MEANX1 = MEANX1/FLOAT(NPR)
MEANX2 = MEANX2/FLOAT(NPR)
MEANX3 = MEANX3/FLOAT(NPR)

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(DELX(J,K)) - MEANX1
DIFFX2 = LOG10(1.-RDATA(J)) - MEANX2
DIFFX3 = - LOG10(RKC - DELK(J,K)) - MEANX3
SX1Y = SX1Y + DIFFX1 * DIFFY
SX2Y = SX2Y + DIFFX2 * DIFFY
SX3Y = SX3Y + DIFFX3 * DIFFY
SX1X1 = SX1X1 + DIFFX1 * DIFFX1
SX1X2 = SX1X2 + DIFFX1 * DIFFX2
SX1X3 = SX1X3 + DIFFX1 * DIFFX3
SX2X2 = SX2X2 + DIFFX2 * DIFFX2
SX2X3 = SX2X3 + DIFFX2 * DIFFX3
SX3X3 = SX3X3 + DIFFX3 * DIFFX3
CONTINUE
300
CONTINUE
350
NPRMN1 = NPR - 1
SX1Y = SX1Y/FLOAT(NPRMN1)
SX2Y = SX2Y/FLOAT(NPRMN1)
SX3Y = SX3Y/FLOAT(NPRMN1)
SX1X1 = SX1X1/FLOAT(NPRMN1)
SX1X2 = SX1X2/FLOAT(NPRMN1)
SX1X3 = SX1X3/FLOAT(NPRMN1)
SX2X2 = SX2X2/FLOAT(NPRMN1)
SX2X3 = SX2X3/FLOAT(NPRMN1)
SX3X3 = SX3X3/FLOAT(NPRMN1)
C CALCULATE THE COEFFICIENTS

\[
AA = SX1X2 \times SX3X3 - SX2X3 \times SX1X3 \\
BB = SXlX2 \times SX3X3 - SX2X3 \times SXlX3 \\
CC = SXlX2 \times SX2X3 - SX2X2 \times SXlX3 \\
DENOM = SXlXI \times AA - SXIX2 \times BB + SXIX3 \times CC
\]

\[
ENN = \left( SX1Y \times AA - SX1X2 \times (SX2Y \times SX3X3 - SX2X3 \times SX3Y) + SX1X3 \times (SX2Y \times SX2X3 - SX2X2 \times SX3Y) \right) / DENOM \\
EMM = \left( SX1X1 \times (SX2Y \times SX3X3 - SX2X3 \times SX3Y) - SX1Y \times BB + SX1X3 \times (SXIX2 \times SX3Y - SX2X3 \times SX1X3) \right) / DENOM \\
QUE = \left( SX1X1 \times (SX2X2 \times SX3Y - SX2Y \times SX2X3) + SX1Y \times CC - SX1X2 \times (SXIX2 \times SX3Y - SX2Y \times SX1X3) \right) / DENOM
\]

\[
LNCEE = MEANY - EENN*MEANXI - EMM*MEANX2 - QUE*MEANX3 \\
CEE = 10.0**LNCEE
\]

ELSEIF (IREGOP .EQ. i) THEN

NPR = 0
MEANY = 0.0
MEANXI = 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))*KC
DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
DO 1250 K = 1, NP(J)
Y = LOG10(DADN(J,K))
\&
- EMM * LOG10(1.-RDATA(J))
\&
- PEE * LOG10(DELK(J,K) - DKTH)
MEANY = MEANY + Y
MEANXI = MEANXI + LOG10(DELK(J,K))
MEANX3 = MEANX3 - LOG10(RKC - DELK(J,K))
1250 CONTINUE
NPR = NPR + NP(J)
1275 CONTINUE
MEANY = MEANY/FLOAT(NPR)
MEANXI = MEANXI/FLOAT(NPR)
MEANX3 = MEANX3/FLOAT(NPR)

C NOW CALCULATE SY2, SX2, AND SXY

DO 1350 J = 1, NDIV
RKC = (1.-RDATA(J))*KC
DKTH = DKTHO*(1.-CO*RDATA(J))**DEE
DO 1300 K = 1, NP(J)
Y = LOG10(DADN(J,K))
\&
- EMM * LOG10(1.-RDATA(J))
\&
- PEE * LOG10(DELK(J,K) - DKTH)
DIFFY = Y - MEANY
DIFFX1 = LOG10(DELK(J,K)) - MEANXI
DIFFX3 = LOG10(RKC - DELK(J,K)) - MEANX3
sx1Y = sx1Y + DIFFX1 * DIFFY
sx3Y = sx3Y + DIFFX3 * DIFFY
sx1X1 = sx1X1 + DIFFX1 * DIFFX1
sx1X3 = sx1X1 + DIFFX3 * DIFFX3
sx3X3 = sx3X3 + DIFFX3 * DIFFX3
1300 CONTINUE
1350 CONTINUE

NPRMN1 = NPR - 1
sx1Y = sx1Y/FLOAT(NPRMN1)
sx3Y = sx3Y/FLOAT(NPRMN1)
sx1X1 = sx1X1/FLOAT(NPRMN1)
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

C LCEE = MEANY - ENN*MEANX1 - QUE*MEANX3
CEE = 10.0**((LCEE)

1600 CONTINUE

ELSEIF ( IREGOP .EQ. 2 ) THEN

NPR = 0
MEANY = 0.0
MEANX1 = 0.0
SXY = 0.0
SXY2 = 0.0
DO 2275 J = 1, NDIV
RXC = (1.-RDATA(J))*KC
DKTH = DKTH0*(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
MEANX1 = MEANX1 + LOG10(DELK(J,K))
MEANX2 = MEANX2 + LOG10(1.-RDATA(J))
2250 CONTINUE
NPR = NPR + NP(J)
2275 CONTINUE

MEANY = MEANY/FLOAT(NPR)
MEANX1 = MEANX1/FLOAT(NPR)
MEANX2 = MEANX2/FLOAT(NPR)

C NOW CALCULATE SY2, SX2, AND SXY

DO 2350 J= 1, NDIV
RXC = (1.-RDATA(J))*KC
DKTH = DKTH0*(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)) - MEANX1
DIFFX2 = LOG10(1.-RDATA(J)) - MEANX2
SXY = SX1Y + DIFFX1 * DIFFY
SXY2 = SX2Y + DIFFX2 * DIFFY
SXY1 = SX1X1 + DIFFX1 * DIFFX1
SXY2 = SX2X2 + DIFFX2 * DIFFX2
2300 CONTINUE
2350 CONTINUE

NPRMNI = NPR - 1

SXY = SXY/FLOAT(NPRMNI)
SXY2 = SXY2/FLOAT(NPRMNI)
SXY1 = SXY1/FLOAT(NPRMNI)
SXY2 = SXY2/FLOAT(NPRMNI)

C CALCULATE THE COEFFICIENTS

7 - 61
DENOM = SXIX1 * SX2X2 - SXIX2 ** 2
ENN = ( SX1Y * SX2X2 - SXIX2 * SX2Y) / DENOM
EMM = ( SXIX1 * SX2Y - SX1Y * SXIX2 ) / DENOM
LNCEE = MEANY - ENN*MEANX1 - EMM*MEANX2
CEE = 10.0**(LNCEE)

ELSEIF (IREGOP .EQ. 3) THEN

NPR = 0
MEANY = 0.0
MEANX1 = 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))
      & - ENN * LOG10(1.-RDATA(J))
      & - PEE * LOG10(DELK(J,K) - DKTH)
      & + QUE * LOG10(RKC - DELK(J,K))
      MEANY = MEANY + Y
      MEANX1 = MEANX1 + LOG10(DELK(J,K))
   CONTINUE
NPR = NPR + NP(J)
3250 CONTINUE
3275 CONTINUE
MEANY = MEANY/FLOAT(NPR)
MEANX1 = MEANX1/FLOAT(NPR)

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))
      & - ENN * LOG10(1.-RDATA(J))
      & - PEE * LOG10(DELK(J,K) - DKTH)
      & + QUE * LOG10(RKC - DELK(J,K))
      DIFFY = Y - MEANY
      DIFFX1 = LOG10(DELK(J,K)) - MEANX1
      SX1Y = SX1Y + DIFFX1 * DIFFY
      SX1X1 = SX1X1 + DIFFX1 * DIFFY
   CONTINUE
3300 CONTINUE
3350 CONTINUE
NPRMNI = NPR - 1
SX1Y = SX1Y/FLOAT(NPRMNI)
SX1X1 = SX1X1/FLOAT(NPRMNI)

C CALCULATE THE COEFFICIENTS
ENN = SX1Y / SX1X1
LNCEE = MEANY - ENN*MEANX1
CEE = 10.0**(LNCEE)

ELSEIF(IREGOP .EQ. 4) THEN

NPR = 0
MEANY = 0.0
MEANX1 = 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

7 - 62
SXIX2 = 0.0
SXIX3 = 0.0
SXIX4 = 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 = (I. - RDATA(J)) * KC
  DKTH = DKTHO * (I. - CO * RDATA(J)) ** DEE
  DO 4250 K = 1, NP(J)
    MEANY = MEANY + LOG10(DADN(J, K))
    MEANXI = MEANXI + LOG10(DELK(J, K))
    MEANX2 = MEANX2 + LOG10(I. - RDATA(J))
    MEANX3 = MEANX3 + LOG10(DELK(J, K) - DKTH)
    MEANX4 = MEANX4 - LOG10(RKC - DELK(J, K))
  4250 CONTINUE
  NPR = NPR + NP(J)
  4275 CONTINUE
  MEANY = MEANY / FLOAT(NPR)
  MEANXI = MEANXI / FLOAT(NPR)
  MEANX2 = MEANX2 / FLOAT(NPR)
  MEANX3 = MEANX3 / FLOAT(NPR)
  MEANX4 = MEANX4 / FLOAT(NPR)

C NOW CALCULATE SY2, SX2, AND SXY

DO 4350 J = 1, NDIV
  RKC = (I. - RDATA(J)) * KC
  DKTH = DKTHO * (I. - CO * RDATA(J)) ** DEE
  DO 4300 K = 1, NP(J)
    DIFFY = LOG10(DADN(J, K)) - MEANY
    DIFFX1 = LOG10(DELK(J, K)) - MEANX1
    DIFFX2 = LOG10(I. - RDATA(J)) - MEANX2
    DIFFX3 = LOG10(DELK(J, K) - DKTH) - MEANX3
    DIFFX4 = - LOG10(RKC - DELK(J, K)) - MEANX4
    SXIX1 = SXIX1 + DIFFX1 * DIFFY
    SXIX2 = SXIX2 + DIFFX2 * DIFFY
    SXIX3 = SXIX3 + DIFFX3 * DIFFY
    SXIX4 = SXIX4 + DIFFX4 * DIFFY
    SX2X2 = SX2X2 + DIFFX2 * DIFFX2
    SX2X3 = SX2X3 + DIFFX2 * DIFFX3
    SX2X4 = SX2X4 + DIFFX2 * DIFFX4
    SX3X3 = SX3X3 + DIFFX3 * DIFFX3
    SX3X4 = SX3X4 + DIFFX3 * DIFFX4
    SX4X4 = SX4X4 + DIFFX4 * DIFFX4
  4300 CONTINUE
  4350 CONTINUE

NPRMNI = NPR - 1
SX1Y = SX1Y / FLOAT(NPRMNI)
SX2Y = SX2Y / FLOAT(NPRMNI)
SX3Y = SX3Y / FLOAT(NPRMNI)
SX4Y = SX4Y / FLOAT(NPRMNI)
SXIX1 = SXIX1 / FLOAT(NPRMNI)
SXIX2 = SXIX2 / FLOAT(NPRMNI)
SXIX3 = SXIX3 / FLOAT(NPRMNI)
SXIX4 = SXIX4 / FLOAT(NPRMNI)
SX2X2 = SX2X2 / FLOAT(NPRMNI)
SX2X3 = SX2X3 / FLOAT(NPRMNI)
SX2X4 = SX2X4 / FLOAT(NPRMNI)
SX3X3 = SX3X3 / FLOAT(NPRMNI)
SX3X4 = SX3X4 / FLOAT(NPRMNI)
SX4X4 = SX4X4 / FLOAT(NPRMNI)

C CALCULATE THE COEFFICIENTS
DENOM = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1X2, SX2X2, SX1X3, SX2X3, SX1X4, SX2X4, SX1X4, SX2X4, SX3X4, SX4X4)

ENN = DETER4(SX1Y, SX2Y, SX3Y, SX4Y, SX1X2, SX2X2, SX1X3, SX2X3, SX1X4, SX2X4, SX3X4, SX4X4) / DENOM

EMM = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1Y, SX2Y, SX3Y, SX4Y, SX1X4, SX2X4, SX3X4, SX4X4) / DENOM

PEE = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1Y, SX2Y, SX3Y, SX4Y, SX1X4, SX2X4, SX3X4, SX4X4) / DENOM

QUE = DETER4(SX1X1, SX1X2, SX1X3, SX1X4, SX1Y, SX2Y, SX3Y, SX4Y, SX1X4, SX2X4, SX3X4, SX4X4) / DENOM

LNCEE = MEANY - ENN*MEANXI - EMM*MEANX2 - PEE*MEANX3 - QUE*MEANX4

CEE = 10.0**(LNCEE)

C WRITE OUT THE REGRESSED VALUES
WRITE(3,6300) CEE, ENN, EMM, PEE, QUE
RETURN

C-------------------------------------- FORMATS --------------------------------------
6000 FORMAT(///,'MATERIAL INPUT',///,'DESCRIPTION: ',2X,A40,///, & 2X,'YIELD STRENGTH',18X,F7.0,///, & 2X,'CRITICAL S I F', & 18X,F7.0,///,'NUMBER OF DIVISIONS',14X,I1, & /,'REREGRESSION OPTION',16X,I1,///)
6050 FORMAT(///,'THRESHOLD MODEL DESCRIPTION', & ///,'DKTHO = ',E12.5, & /,'CO = ',E12.5, & /,'d = ',E12.5)
6150 FORMAT(///,'STRESS RATIO R = ',F7.2,///,'da/dN',8X,'DELK')
6200 FORMAT(2X,E12.5,2X,E12.5)
6300 FORMAT(///,'REGRESSION OUTCOME',///,'C',12X,'n',12X,'m',12X,'p',12X,'q',/)  
6400 FORMAT(E12.5,4(4X,E12.5))
6600 FORMAT(IOX,II,'-',II,5X,E12.5,5X,E12.5)

END

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
C Implicit None
C***************************************************************
C SUBROUTINE INPUT READS IN THE DATA AND ECHOES IT
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 INPUT (ANGLE, BLFPER, COEXP, E, EM, LOCAT, NBLIFE, NHYPER, N
LIFE, NLIFET, NRAN, NU, NUMSEG, PERIOD, RSO, SE, STRHIS, TRUNC)
C 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, NUMSEG, TYPE(MAXLD)
DOUBLE PRECISION RAND
REAL AREDA, AERDS, AERSA, AERSB, AIA, AIB,
AIR1, AIR2, AT1, AT2, AIR,
LAT, ANGLE, BLFPER(MAXBLF), COEXP, AOCRA, AOCR2, AOCT1, AOCT2,
AOCR, AOCT, ASTRA, ASTRB, DPCMU, DPCSIG,
DSTRA, DSTRB, DTIMU, DTISIG,
DTOMU, DTOSIG, E(MAXSEG), EM, FK(10),
INDIAA, INDIAB, INDIR, INDIR2,
INDIR3, INDIR3T, INDIR2T,
KAMA, KLAMB, LAMKA, LAMKB, LAMKC, LAMKB,
LAMGR, LAMGRB,
LAMNA, LAMB, LAMNC, LAMD, LAMNU, LAMNSG,
LAMSA, LAMS, LAMSC, LAMSD, LAMSMU, LAMSSG,
LAMWA, LAMB
REAL M(2, MAXLD), MSTAT(2), MVARA,
MVARB, NEUA, NEUBB, NU,
P(MAXLD), PCMU, PCO, PCSIG, PCSIGN,
PCSIGB, PERIOD,
PSTAT, RSO, RT(10), SE(MAXSEG), SXTRA,
SXTRA, STRHIS(MAXLIFE, MAXM), SX(MAXLD), SXST,
SY(MAXLD), SYST, SXZ(MAXLD), SXZST, SYZ(MAXLD), SZST,
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, WITHR1, WITHR2, WITHR, WITHT,
& WOFFA, WOFFB, WOFFC, WOFFD,
& WOFFHI, WOFFLO, WOFFR, WOFFR1, WOFFR2, WOFFR3,
& WOFFR4, WOFFT, WOFFT1, WOFFT2, WOFFT3, WOFFT4

CHARACTER*6 LDNAME(MAXLD)

COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
& P, T, M, V, PCO, SXST, SYST, SZST, SXYST,
& SXST, SYST, SX, SY, SZ, SXY, SXZ, SYZ

COMMON/DRIVRS/ AENDA, AERDB, AERSA, AERSB, AIA,
& AIB, AIR1, AIR2, AIT1, AIT2, AIR, AIT,
& AOCA, AOCR1, AOCR2, AOCCT1, AOCCT2, AOCR, AOCCT,
& ASTR, ASTRB,
& DPCMU, DPCSIG, DSTR, DSTRB, DTIMU, DTISIG, DTOMU, DTOSIG,
& INDIA, INDIAB, INDIR, INDIR1, INDIR2, INDIR1, INDIR2, INDIR1,
& KLAMA, KLAMB, LAMRA, LAMRB, LAMRHA, LAMRKB, LAMKCA, LAMKCB,
& LAMNA, LAMNB, LAMNC, LAMND, LAMNNU, LAMNSG,
& LAMSA, LAMSB, LAMSC, LAMSD, LAMSS, LAMSG,
& LAMWA, LAMWB, MVRA, MVARB, NEUBA, NEUBB,
& PCMU, PCMUA, PCMUB, PCSIG, PCSIGA, PCSIGB,
& RAND,
& SSTR, SSTRB,
& THIC, THICB, THICR1, THICR2, THICR, THICT1, THICT2, THICT,
& TIMU, TIMUA, TIMUB, TISIG, TISIGA, TISIGB,
& TOMU, TOMUA, TOMUB, TOSIG, TOSIGA, TOSIGB,
& WITHA, WITHB, WITHR1, WITHR2, WITHR1, WITHR2, WITHR, WITHT,
& WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
& WOFFR, 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,*)' IOUT - 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'
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(I,*) (BLFPER(J), J = 1, NBLIFE)
ENDIF

C READ DRIVER INFORMATION

IF (KPROB .EQ. 1) THEN
  READ(I,*) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,
  WOFFC, WOFFD, WOFFR3, WOFFR4, WOFFT3, WOFFT4,
  WOFF
  & INDIAA, INDIB, INDIR1, INDIR2, INDIT1, INDIT2,
  & THICA, THICB, THICR1, THICR2, THICT1, THICT2,
  & AOCB, AOCR1, AOCR2, AOCR3, AOCR4, AOCR5, AOCR6, AOCR7
ELSEIF (KPROB .EQ. 2) THEN
  READ(I,*) WITHA, WITHB, WITHR1, WITHR2, WITHT1, WITHT2
ENDIF

READ(I,*)
AIA, AIB, AIR1, AIR2, AIT1, AIT2,
& LAMNA, LAMNB, LAMNC, LAMND,
& LAMS, LAMCB, LAMCB, LAMSD

IF (KPROB .EQ. 1) THEN
  READ(I,*) TIMUA, TIMUB, TISIGA, TISIGB,
  & TOMUA, TOMUB, TOSIGA, TOSIGB,
  & PCMU, PCMB, PCSIGA, PCSIGB
ENDIF

C CALCULATE SOME DRIVER VARIABLES

DTIMU = TIMUB - TIMUA
DTISIG = TISIGB - TISIGA
DTOMU = TOMUB - TOMUA
DTOSIG = TOSIGB - TOSIGA
DPCMU = PCMB - PCMU
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,*) LAMKA, LAMKB, AERDA, AERDB, AERSA, AERSB,
& ASTRA, ASTRB, DSTRA, DSTRB
IF(INEUB .EQ. 1) THEN
  READ(I,*) NEUBA, NEUBB
ENDIF
ELSE
  READ(I,*) SSTRA, SSTRB, DSTRA, DSTRB
ENDIF

READ(1,*) LAMKHA, LAMKHB, LAMKCA, LAMKCB,
& KLAMA, KLAMB, LAMGRA, LAMGRB
IF(KGROW .EQ. 2) THEN
  READ(I,*) MVARA, MVARB
ENDIF

C READ THE LOADS OR STRESSES

IF (KPROB .EQ. 1) THEN
  READ(I,*) NLOAD, PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1),
  & VSTAT(2)
DO 15 I = 1, NLOAD
  READ(I,*) LDNAME(I), TYPE(I), P(I), T(I), M(I,1), M(2,I),
  & M(2,1), M(2,2), M(2,3), M(2,4), M(2,5), M(2,6), M(2,7), M(2,8)
& V(1,I), V(2,I)
IF ((TYPE(I) .LT. 1) .OR. (TYPE(I) .GT. 3)) THEN
  WRITE(*,*) 'ERROR: LOAD INCORRECTLY TYPED'
  CALL TRMNAT
ENDIF

15 CONTINUE
ELSEIF(KPROB .EQ. 2) THEN
  READ(1,*), NLOAD, SXST, SYST, SZST, SXYST, SYYST
  DO 16 I = 1, NLOAD
    READ(1,*) LDNAME(I), TYPE(I), SX(I), SY(I), SZ(I), SXY(I),
    & SXZ(I), SYZ(I)
    IF ((TYPE(I) .LT. I) .OR. (TYPE(I) .GT. 2)) THEN
      WRITE(*,*) 'ERROR: LOAD INCORRECTLY TYPED'
      CALL TRMNAT
    ENDIF
  16 CONTINUE
ENDIF

C READ MISCELLANEOUS INFO
IF(KPROB .EQ. 1) THEN
  READ(1,*) PCO, LOCAT, ANGLE
ENDIF

READ(1,*) RSO, PERIOD, TRUNC, NRAN
C ECHO DATA TO CRKRES
WRITE(3,900)
IF(KPROB .EQ. 1) THEN
  WRITE(3,901) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,
  & WOFFC, WOFFD, WOFFR3, WOFFR4, WOFFT3, WOFFT4,
  & WOFFE
  WRITE(3,905) THICA, THICB, THICR1, THICR2, THICT1, THICT2
  WRITE(3,911) AOCA, AOCB, AOCR1, AOCR2, AOCIT1, AOCIT2
ELSEIF(KPROB .EQ. 2) THEN
  WRITE(3,902) WITHA, WITHB, WITHRI, WITHR2, WITHT1, WITHT2
  ENDIF
WRITE(3,910) AIA, AIB, AIR1, AIR2, AIT1, AIT2
WRITE(3,906) LAMNA, LAMNB, LAMNC, LAMND
WRITE(3,907) LAMSA, LAMSB, LAMSC, LAMSD
IF(KPROB .EQ. 1) THEN
  WRITE(3,908) TIMUA, TIMUB, TISISGA, TISISGB,
  & PCMUA, PCMUB, PCSIGA, PCSIGB
  WRITE(3,9081) LAMWA, LAMWB, AERDA, AERDB, AERSA, AERSB,
  & ASTRA, ASTRB, DASTRA, DASTRB
ELSEIF(KPROB .EQ. 1) THEN
  WRITE(3,9082) SSTRA, SSTRB, DSTRA, DSTRB
ENDIF
WRITE(3,909) LAMKHA, LAMKHB, LAMKCA, LAMKCB,
  & KLAMA, KLAMB, LAMGRA, LAMGRB
IF(KGROW .EQ. 2) THEN
  WRITE(3,9091) MVARA, MVARB
ENDIF
IF(KPROB .EQ. 1) THEN
  WRITE(3,920) PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1), VSTAT(2)
  DO 20 I = 1, NLOAD
    WRITE(3,921) LDNAME(I), P(I), T(I), M(I,1), M(2,I), V(I,1),
    & V(2,I)
  20 CONTINUE
ELSEIF(KPROB .EQ. 2) THEN
  WRITE(3,922) SXST, SYST, SZST, SXYST, SXZST, SYZST
  DO 21 I = 1, NLOAD
    WRITE(3,921) LDNAME(I), SX(I), SY(I), SZ(I), SXY(I),
    & SXZ(I), SYZ(I)
21 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
C READ THE FK VS. RT CURVE FOR WELD STRESS CONCENTRATION FOR HEX COIL PROBLEM
WRITE(3,928)
DO 30 I = 1, 10
    READ(I,*) 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
C====== FORMAT STATEMENTS TO ECHO INPUT DATA TO CRKRES ======
900 FORMAT(2X,'Copyright (C) 1991, California Institute of '
       & 'Technology. U.S. Government is acknowledged.',/,'NASA Contract NAS7-918 is acknowledged.',/,'30X,P R O C R K ',
       & 'INPUT DATA',/,'4X,PARAMETER DISTRIBUTIONS',/,'48X,RHO',
       & '6X,THETA')
901 FORMAT(/,'6X,WELD OFFSET (%)',/,'3X,Be({F4.2,','F5.2,}),',
       & '1X,','F5.2,','F7.5,','U(',F4.1,','F5.1,')',
       & '2X,','F7.5,','F9.5,','U(',F4.1,','F5.1,')',
       & '20X,','F7.5,','F9.5,','U(',F4.1,','F5.1,')',
       & '4X,','U(',F4.1,','F5.1,')',/,'26X,'TEST ','F4.2')
902 FORMAT(2X,'CHANNEL WIDTH=',4X,'4X','BE(',F6.4,':',F7.4,')',2X,
&'U(',F7.5,':',F6.5,')',4X,'U(',F4.1,':',F5.1,')}
904 FORMAT(2X,'INNER DIAMETER=',4X,'BE(',F6.4,':',F7.4,')',2X,
&'U(',F7.5,':',F6.5,')',4X,'U(',F4.1,':',F5.1,')}
905 FORMAT(2X,'WALL THICKNESS=',4X,'BE(',F6.4,':',F7.4,')',2X,
&'U(',F7.5,':',F6.5,')',4X,'U(',F4.1,':',F5.1,')}
906 FORMAT(2X,'LAMBDA RANDOM',5X,'K: U(',F7.5,':',F8.5,')',
&20X,'STRAIN GAGE FACTOR: ',F9.7,)
907 FORMAT(2X,'LAMBDA SINE',7X,'K: U(',F7.5,':',F8.5,')',
&20X,'STRAIN GAGE FACTOR: ',F9.7,)
908 FORMAT(2X,'INNER TEMPERATURE',4X,'NORMAL: MU(',
&F5.1,':',F6.1,') SIGMA(',F5.1,':',F6.1,')',
&2X,'OUTER TEMPERATURE',4X,'NORMAL: MU(',
&F5.1,':',F6.1,') SIGMA(',F5.1,':',F6.1,')',
&2X,'INNER PRESSURE',4X,'NORMAL: MU(',
&F5.1,':',F6.1,') SIGMA(',F5.1,':',F6.1,')',
909 FORMAT(2X,'DYN AERO LOAD FAC ',3X,'U(',FS.5,':',F9.5,')',
&2X,'STAT AERO LOAD FAC ',3X,'U(',F4.5,':',F5.5,')',
&2X,'AERO STR ANAL FAC ',3X,'U(',F8.5,':',F9.5,')',
&2X,'DYN STR ANAL FAC ',3X,'U(',F8.5,':',F9.5,')',
910 FORMAT(2X,'CRACK SIZE A ',5X,'BE(',F6.4,'!',F7.4,')',4X,'U(',F7.5,':',F8.5,')',4X,'U(',F4.1,':',F5.1,')',
911 FORMAT(2X,'CRACK SHAPE A/C',5X,'BE(',F6.4,'!',F7.4,')',2X,
&'U(',F4.1,':',F5.1,')',
912 FORMAT(2X,'STRESS INPUT',
&///,5X,'SX ',5X,'SY ',5X,'SZ ',4X,'SXY ',4X,'SXZ ',4X,'SYZ ',
&///,2X,'STATIC',
&///,2X,F9.6,5(3X,E9.3))
913 FORMAT(2X,'ELASTIC MODULUS',32X,E9.3,
7 - 70
SUBROUTINE SETDEF(LIFE, NCRL)

C 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

C**********************************************************************
SUBROUTINE STRAN1 PERFORMS THE STRESS TRANSFORMATION FOR THE PARTICULAR
MODE AND LOCATION AND CALCULATES EQUIVALENT STRESS HISTORY
PROGRAMER: S. SUTHARSHANA
DATE: DECEMBER 1992
VERSION: 92.5
C**********************************************************************

SUBROUTINE STRAN1 (AERD, AERS, ASTR, ANGLE, COEXP, DLTAT, &
   DSTR, EM, INDIA, LAMN, LAMS, LAMW, LOCAT, &
   NRAN, NU, FC, SPR, STRHIS, THIC, WOFF)

C SUBPROGRAMS: M4L1, M4L2
C 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)

RETURN
END
REAL AERD, AERS, ASTR, ANGLE, COEXP, DLTAT, DSTR, EM, FK(10),
INDIA, KT(2,2), LAMN, LAMM, LAMW,
M(2, MAXLD), MSLAM(2, MAXLD), MSTAT(2),
NU, P(MAXLD), PC, PCO, PLAM(MAXLD), PSLAM, PSTAT,
P(10), SCLFAC, SFR(MAXM), STATIC(4),
STRAMP(4, MAXLD), STRHIS(MAXLD, MAXM), SX(MAXLD), SXST,
SYX(MAXLD), SXYS, SYZ(MAXLD), SXYZST, SYZST, T(MAXLD),
THIC, TLAM(MAXLD), TSLAM, TSTAT,
V(2, MAXLD), VSLAM(2), VSTAT(2), WOFF

COMMON/LOADS/NLOAD, PSTAT, TSTAT, MSTAT, VSTAT, TYPE,
P, T, M, V, PCO,
SXST, SYST, SXYZST, SYZST, SYZST,
SX, SY, SZ, SXYZ, SYZ

COMMON/FKVSRT/FK, RT

DATA KT/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)
   TLM(II) = 0.0
   MSLAM(1,II) = SCLFAC * M(1,II)
   MSLAM(2,II) = SCLFAC * M(2,II)
   VSLAM(1,II) = 0.0
   VSLAM(2,II) = 0.0
ELSE IF (TYPE(II) .EQ. 2) THEN
   SCLFAC = LAMS * DSTR
   PLAM(II) = SCLFAC * P(II)
   TLM(II) = 0.0
   MSLAM(1,II) = SCLFAC * M(1,II)
   MSLAM(2,II) = SCLFAC * M(2,II)
   VSLAM(1,II) = 0.0
   VSLAM(2,II) = 0.0
ELSE
   SCLFAC = AERD * ASTR
   PLAM(II) = SCLFAC * P(II)
   TLM(II) = 0.0
   MSLAM(1,II) = SCLFAC * M(1,II)
   MSLAM(2,II) = SCLFAC * M(2,II)
   VSLAM(1,II) = 0.0
   VSLAM(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 = ', TLM(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 SUBROUTINE STRAN2 PERFORMS THE STRESS CALCULATION FOR THE EXHEX
C PROGRAMMER: S. SUTHARSHANA
C DATE: 19 NOV 1989
C VERSION: 92.1, 92.2, 92.3, 92.4, 92.5
C***************************************************************************
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), PSTAT, PSTAT, SPR(MAXM), SSTR,
& STRAMP(MAXM), STRHIS(MAXM, 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, FC0,
$\& SXST, SYST, SZST, SXYST, SXZST, SYZST,$
$\& SX, SY, SZ, SXY, SXZ, SYZ$

C SET UP THE STRESS AMPLITUDES

DO 50 II = I, NLOAD
  IF (TYPE(II).EQ.I) 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 I00 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. GRONDALESI, L. NEWLIN
C DATE: 9MAB87
C SUBPROGRAM: GAM
C
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

7 - 74

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(*,*) 'A =', A, ' ALPHA =', ALPHA

CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)

V1 = - ALOG(U1)
V2 = - ALOG(U2)

ARG = A * (V1 - ALOG(V1) - 1.)

IF (V2 .LT. ARG) GOTO 10

GAM = ALPHA * V1

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 BDLDCF 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 7.2-1 List of Subprograms For Program BDLDCF
(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>ADDREG¹</td>
<td>4.1.3.9</td>
<td>Adds the <em>m</em> ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.</td>
</tr>
<tr>
<td>ADDRGN¹</td>
<td>4.1.3.15</td>
<td>Adds the <em>m</em> ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.</td>
</tr>
<tr>
<td>BETAGN²</td>
<td>4.4.5</td>
<td>Generates Beta(<em>a</em>, <em>b</em>, <em>p</em>, <em>θ</em>) random variates.</td>
</tr>
<tr>
<td>BDLCF</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>CONCAV³</td>
<td>4.1.3.10</td>
<td>Adjusts the upper bound of the posterior ranges on <em>m</em> to be consistent with concavity constraints.</td>
</tr>
<tr>
<td>CONVRT⁴</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>EXPCTD⁵</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 <em>K</em> (where <em>A</em> = <em>K</em>/<em>m</em>) 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 <em>m</em> for each life region by adjusting the range (to ensure concavity) and then sampling from the Uniform distribution over the appropriate <em>m</em> range.</td>
</tr>
<tr>
<td>FINDMC</td>
<td>4.1.3.5</td>
<td>Calculates the <em>m</em> range implied by the constraint on the coefficient of variation of fatigue strength, <em>C</em>, for each life region, by using Equations 2-28 through 2-32 of [1].</td>
</tr>
<tr>
<td>FINDMN⁶</td>
<td>4.1.5.2</td>
<td>Obtains the value of <em>m</em> for each life region by sampling from the appropriate truncated Normal distribution on <em>m</em>.</td>
</tr>
<tr>
<td>FINDSB</td>
<td>4.1.5.7</td>
<td>Calculates the life region &quot;tie-points&quot; or strain values which correspond to the &quot;life boundaries,&quot; conditional on the randomly selected <em>m</em> for each region. Also calculates <em>K</em>, characterizing the specific material S/N data set, which is a function of β₀ and <em>k</em>.</td>
</tr>
<tr>
<td>FNDRNG⁷</td>
<td>4.1.3.8</td>
<td>Combines the 95% confidence interval, <em>J₀</em>, with the implicit and explicit constraints on <em>m</em>, to obtain posterior credibility ranges on <em>m</em> for each life region.</td>
</tr>
<tr>
<td>NAME</td>
<td>SECTION</td>
<td>PURPOSE</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
<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 $\sigma^2$, 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>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>5.8*</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 $I_o$ 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 $\beta_o$ 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 nonparametric materials characterization model.</td>
</tr>
<tr>
<td>MUSIG</td>
<td>4.1.3.13*</td>
<td>Calculates the posterior Normal distribution parameters, mean $m_<em>$ and standard deviation $\sigma_</em>$, for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NORMGN</td>
<td>4.4.3*</td>
<td>Generates 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>PEB</td>
<td>5.2.3.5</td>
<td>Controls the logical flow of the bootstrapping portion of the nonparametric 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>
</tbody>
</table>
Table 7.2-1  List of Subprograms For Program BLDLCF (Cont’d)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<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>RAINF$^{3}$</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 BLDDLCD and RELATD; calls CONVRT to transform the strain data to a strain ratio of -1.0; and echoes the data to BLDDLCO 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 “worst of $N$” 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, BLDLDF, 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.</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 $\beta_0$, 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>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<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 $\omega(t_b)$.</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 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>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<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>λα 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 λα.</td>
</tr>
<tr>
<td>LAMAB</td>
<td>RE</td>
<td>Uniform distribution upper bound of λα.</td>
</tr>
<tr>
<td>LAMDA</td>
<td>RE</td>
<td>λdam 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>LAMDAA</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>λG 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 λG.</td>
</tr>
<tr>
<td>LAMGB</td>
<td>RE</td>
<td>Upper bound of the Beta distribution on λG.</td>
</tr>
<tr>
<td>LAMGR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter p for λG.</td>
</tr>
<tr>
<td>LAMGR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of parameter p in the Beta distribution of λG.</td>
</tr>
<tr>
<td>LAMGR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of parameter p in the Beta distribution of λG.</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>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 BLDLCF. 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 BLDLCF 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 BLDLCF 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 $\text{LIFEL( )}$ 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 $\text{LIFEW( )}$ 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 $\ln(A) = \ln(BIGK) \times MM$ 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) 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 $\ln(\text{PHI}) \times \text{MM}$ 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>MANALB</td>
<td>RE</td>
<td>Uniform distribution lower bound of $\lambda^{MA}$.</td>
</tr>
<tr>
<td>MAXBLF</td>
<td>INT</td>
<td>Uniform distribution upper bound of $\lambda^{MA}$.</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>
</tbody>
</table>
Table 7.2-2  List of Variables For Program BDLDCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<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(^{5}) 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.(^{2})</td>
</tr>
<tr>
<td>NBND(0:MAXREG)</td>
<td>RE</td>
<td>( N*_{j, 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 BLDLCF. 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>$\omega_o$ (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($\beta_o, \eta_0(\beta_o)$) 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.6</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_i) ) (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 ( \sigma ) 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_{s}) ) (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 2-D array containing strain points with strain ratio $= -1.0$, for the specific material S/N data set partitioned into life regions.</td>
<td></td>
</tr>
<tr>
<td>SZERO</td>
<td>RE Strain tensile test point, $S^o_0$ (%).^8</td>
<td></td>
</tr>
<tr>
<td>TANAL</td>
<td>RE The randomly selected thermal strain analysis accuracy factor, $\lambda_{TA}$ in Equation 3-4.</td>
<td></td>
</tr>
<tr>
<td>TANALA</td>
<td>RE Uniform distribution lower bound of $\lambda_{TA}$.</td>
<td></td>
</tr>
<tr>
<td>TANALB</td>
<td>RE Uniform distribution upper bound of $\lambda_{TA}$.</td>
<td></td>
</tr>
<tr>
<td>TGAS</td>
<td>RE $T_{gas}$ °R in Equation 3-2, the randomly selected gas temperature at $t_1$.</td>
<td></td>
</tr>
<tr>
<td>TGASA</td>
<td>RE Lower bound of the Beta distribution on $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TGASB</td>
<td>RE Upper bound of the Beta distribution on $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TGASR</td>
<td>RE Randomly selected Beta distribution location parameter $\rho$ for $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TGASR1</td>
<td>RE Uniform distribution lower bound of parameter $\rho$ in the Beta distribution of $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TGASR2</td>
<td>RE Uniform distribution upper bound of parameter $\rho$ in the Beta distribution of $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TGAST</td>
<td>RE Randomly selected Beta distribution shape parameter $\theta$ for $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TGAST1</td>
<td>RE Uniform distribution lower bound of parameter $\theta$ in the Beta distribution of $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TGAST2</td>
<td>RE Uniform distribution upper bound of parameter $\theta$ in the Beta distribution of $T_{gas}$.</td>
<td></td>
</tr>
<tr>
<td>TRBIGK(0:MAXREG)</td>
<td>RE 1-D array containing values of the materials model location parameter $K$ consistent with the tensile point $S^o_0$.^8</td>
<td></td>
</tr>
<tr>
<td>TRSBND(0:MAXREG)</td>
<td>RE 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 $S^o_0$.^8</td>
<td></td>
</tr>
<tr>
<td>TRUNC</td>
<td>RE 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>
<td></td>
</tr>
</tbody>
</table>
### Table 7.2-2
List of Variables For Program BLDCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTART</td>
<td>RE</td>
<td>$T_s$ ($^\circ$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 $S_0$. 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.8</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>GTPVAR</td>
<td>7-139</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-140</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-144</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-145</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-146</td>
</tr>
<tr>
<td>EXPECTD</td>
<td>7-147</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-149</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-151</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>7-139</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-140</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-144</td>
</tr>
<tr>
<td>FNSRNG</td>
<td>7-145</td>
</tr>
<tr>
<td>ADDRREG</td>
<td>7-146</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-147</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-162</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-163</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-164</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-165</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-166</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-168</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-158</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-115</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-158</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-157</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-157</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-158</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-158</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-162</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-163</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-164</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-165</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-161</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-158</td>
</tr>
<tr>
<td>GAM</td>
<td>7-115</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>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-169</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-114</td>
</tr>
</tbody>
</table>
### Program BDLDCF Listing Temporal Order, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BDLDCF</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>BDLLIF</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>TRNSFMD</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>
PROGRAM BLDLCF CONTROLS THE FLOW OF LOGIC OF THE LOW CYCLE FATIGUE ANALYSIS OF THE TURBINE BLADE FOIL PROBLEM

PROGRAMMER: L. NEWLIN

DATE: CODE: 7JAN92 COMMENTS: 3APR92

VERSION: 3.4 (MATCHR V8.5, RAINF3 V1.1, INSORT V2.1)

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

C** PROGRAM BLDLCF

C** SUBPROGRAMS:

C** FILES:

C** IMPLICIT NONE

C** PARAMETER

C** COMMON IOUT

C** COMMON

DOUBLE PRECISION RAND

REAL 
ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1, BDLIF, 
BLFPER(MAXBLF), BZERO, DUM, EBEND, EBENDA, EBENDB, 
EMNOM, ETHNOM(MAXM), FAAA, FAAA, FAC, FACTR, FAD, EFA, 
FAF, FAFRM, FAFR, FD1A, FD1B, FD1C, FD1D, FD1F, 
FD2A, FD2B, FD3A, FD3B, FDERR, FDERRA, FIFTY, FTY, 
HGAS, HGASA, HGASR, HGASRI, HGASR2, HGAST, 
HGAST1, HGAST2, KRAINT, LAMA, LAMA, LAMAB, LAMAD, 
LAMADA, LAMDB, LAMAM, LAMGB, LAMGR, LAMGR1, 
LAMGTR, LAMOT, LAMOT1, LAMOT2, LAMB, LAMPS, 
LAMTP, LANTMA, LANTMB, LIFE(MAXLIF), LINA(0:MAXREG), 
LNPHI, LNZ, LPHINC(MAXREG), MANAL, MANALA, MANALB, 
MEDM(MAXREG), MINPHI, MM(0:MAXREG), MODER1, MODER2, 
MU(MAXREG), NBDND(0:MAXREG), NEWLNF, NF(MAXDAT, MAXREG), 
NOMSF, PERIOD, PHI, PHISP, PSIG, PSIG, 
RANGEM(2, MAXREG), RPM(MAXMM), SBND(0:MAXREG), 
SIG(MAXREG), SLOPE, SLOPE, SLOPE, SLOPE, SLOPE1, 
SLOPE2, SLOPE2, SLOPE2, SLOPE2, SLOPE2, SLOPE2, 
SLOP2, SLOP2, SLOP3, SLOP3, SLOP3, SLOP3, SLOP3, 
SLOP3, SLOP3, SLOP3, SLOP3, SLOP3, SLOP3, SLOP3, 
TANAL, TANALA, TANALA, TANALB, TGAS, 
TGAS, TGAS, TGAS, TGAS, TGAS, TGAS, TGAS, TGAS, TGAS, TGAS, TGAS, 
TGAST, TGAST2, TRBIGK(0:MAXREG), TRSBD(0:MAXREG), TROUN, 
TSTART, TSTMU, TSTSIG, TEXP, T

C** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

OPEN (1, FILE = 'BLDLCD', STATUS = 'OLD')
OPEN (3, FILE = 'BLDLCO', STATUS = 'NEW')
OPEN (7, FILE = 'DUMP-NEW', STATUS = 'NEW')
OPEN (8, FILE = 'IOUTPR-NEW', STATUS = 'NEW')
OPEN (9, FILE = 'LOWLIF-NEW', STATUS = 'NEW')

READ(1,*) RAND
WRITE(8,*) ' RANDOM NUMBER SEED = ', RAND
READ(1,*) IOUT
WRITE(8,*) ' IOUT (MATCHR = 10, BDLDCF = 15, RAINF3 = 20) = ', IOUT
READ(1,*) NLIIF
WRITE(8,*) ' INNER LOOP SIZE = ', NLIFE
READ(1,*) NHYPER
WRITE(8,*) ' OUTER LOOP SIZE = ', NHYPER

C

7 - 102
READ(1,*), NSYM
WRITE(8,*), SYM
WRITE(8,*), (O-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
READ(1,*), VARPHI
WRITE(8,*), TYPE OF INTRINSIC VARIATION DESIRED '
WRITE(8,*), (1 - Weibull; 2 - Lognormal) = VARPHI

IF ((VARY .LT. 0) OR. (VARY .GT. 3)) THEN
  CALL TRMNAT
ENDIF

IF ((NMED .NE. 0) AND. (NMED .NE. 1)) THEN
  CALL TRMNAT
ENDIF

IF ((MPROC .LT. 0) OR. (MPROC .GT. 1)) THEN
  CALL TRMNAT
ENDIF

IF ((VARPHI .LT. 1) OR. (VARPHI .GT. 2)) THEN
  CALL TRMNAT
ENDIF

READ(1,*), NBLIFE
IF (NBLIFE .GT. 0) READ(1,*), (BLFPER(J), J = 1, NBLIFE)

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,
& MANALA, MANALB, LAMPA, LAMPB,
& LAMTA, LAMTBL, LAMAA, LAMAB,
& LAMTB, LAMTMB
READ(1,*), EMNOM, NOMSPD, 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

WRITE(3,900)
WRITE(3,901) HGASA, HGASB, HGASR1, HGASR2, HGAST1, HGAST2,
& TGASA, TGASB, TGASR1, TGASR2, TGAST1, TGAST2,
& SLOPEA, SLOPEB, SLOPR1, SLOPR2, SLOPT1, SLOPT2,
& LAMGA, LAMGB, LAMGR1, LAMGR2, LAMGT1, LAMGT2
WRITE(3,903) TSUBI, SPEEDM, SPEEDS, FAERRM, FAERRS,
& TSTMU, TSIG, FDERRM, FDERRS
WRITE(3,904) EBENDA, EBENDB, LAMPA, LAMPB, MANALA, MANALB,
& LAMAA, LAMAB, TANALA, TANALB
WRITE(3,905) EXP(LAMDAAA), EXP(LAMDAB), EXP(LAMTHA), EXP(LAMTHB)
WRITE(3,906) ENOM, NOMSPD, 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 PRACT TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS
CALL PRACT (RAND, HGASRI, HGASR2, HGASI, HGAST2, HGASR, HGAST)
CALL PRACT (RAND, TGASRI, TGASR2, TGASI, TGAST2, TGASR, TGAST)
CALL PRACT (RAND, SLOPRI, SLOPR2, SLOPT1, SLOPT2, SLOPR, SLOPT)
CALL PRACT (RAND, LAMGRI, LAMGR2, LAMGTI, LAMGT2, LAMGR, LAMGT)
CALL PRACT (RAND, MANALA, MANALB, TANALA, TANALB, MANAL, TANAL)
C ** CALL PAREST TO PERFORM THE PARAMETER ESTIMATION ASPECT OF THE
C 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
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, LAMA, LAMB, LAM, 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
  WEIBULL INTRINSIC MATERIALS VARIATION
  DO 230 M = 1, NSYM
    CALL WEIBGN (BZERO, RAND, PHI)
    MINPHI = MIN (PHI, MINPHI)
  230 CONTINUE
  PHI = MINPHI
ELSE
  LOGNORMAL INTRINSIC MATERIALS VARIATION
  DO 231 M = 1, 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, ' TGAS = ', TGAS
  WRITE(8,*) 'SLOPE = ', SLOPE, ' LAM = ', LAMG
  WRITE(8,*) 'LAM = ', LAM, ' LAM = ', LAM
  WRITE(8,*) 'LAM = ', LAM, ' LAM = ', LAM
  WRITE(8,*) 'LAMT = ', LAMAT, ' PHI = ', PHI
  WRITE(8,*) 'MANAL = ', MANAL, ' TANAL = ', TANAL
  WRITE(8,*) 'TSTART = ', TSTART, ' MODER = ', MODER1
  WRITE(8,*) 'MODER = ', MODER1
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)
WRITE(8,*) 'KRATIO = ', KRATIO, ' Z = ', Z
WRITE(8,*) 'TRSBND =', TRSBND(L), ' FACTR = ', FACTR

250 CONTINUE

C ** CALL BLDLIF TO OBTAIN BLADE LCF LIFE

NEWLIF = LAMDA * LAMTM * BLDLIF (TGAS, HGAS, FAA, FAB, FAC, 
& FAD, FAE, FAF, MODER1, RPM, TSUBI, SPEED, SLOPE, 
& TSTART, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F, 
& MODER2, FD2A, FD2B, FD23, FD28, ETHNOM, MANAL, 
& LAMP, NOMSPD, EMNOM, TANAL, LAMA, LAMG, ETBEND, 
& NTIME, TRUNC, PERIOD, WEXP, MM, LMA, LPHIM, 
& KRATIO, 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
IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES TO FILE LOWLIF

DO 100 J = 1, (NLIFET / 100)
   WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
100 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 BLDLCO

7 - 106
INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,

INPUT DATA

DRIVERS

PARAMETER DISTRIBUTIONS

RHO

THETA,
STOP
END

C**************************************************************
C SAMPLE 'BLDLCD' INPUT FILE
C**************************************************************
C 675 .................................... RANDOM NUMBER SEED
C 100 .................................... OUTPUT DUMP CONTROLLER
C 200 .................................... INNER LOOP SIZE
C 50 .................................... OUTER LOOP SIZE
C 2 .................................... SYMMETRY NUMBER
C 0 .................................... NORMAL MEDIAN NOT REQUIRED
C 1 .................................... WEIBULL INTRINSIC VARIATION
C 0 .................................... MAT. PROC. VAR. NOT REQUIRED
C 0 .................................... NUMBER OF BLIVES REQUESTED
C 0.0001 ................................ B.01 LIFE
C 0.001 .................................. B.1 LIFE
C 0.01 ................................... B1 LIFE
C 676.2730.0.50.50.00.0 ................................ Hgas (A,B) (R1,R2) (T1,T2)
C 800.2000.0.50.50.00.0 ................................ Tgas (A,B) (R1,R2) (T1,T2)
C 2730.2730.0.50.50.00.0 ................................ Tgas UNCERTAINTY FACTOR
C 10 ....................................... DECEL SLOPE (A,B) (R1,R2) (T1,T2)
C 0.80 1.20 .................................. Tgas UNCERTAINTY FACTOR
C 0.000 .................................. Rotor Speed Variation Parameters:
C 0.0 ....................................... Faccel Modeling Error Mean & Std. Dev.
C 160.0 40.67 .................................. Decel Tstart Mean & Standard Deviation
C 975.3 28.6 .................................. Standard Response Probe Mean & Std Dev
C 0.0 0.003 .................................. Fdecel Modeling Error Mean & Std Dev
C 0.0 0.0 ................................... Strain Due to Gas Bending (%)
C 0.563283 0.00 .................................. LAMBDA BLADE PULL
C 0.00 0.00 .................................. Mechanical Analysis Accuracy Factor
C -0.295 38482 .................................. Coefficient of Thermal Expansion
C 1.0 ....................................... Thermal Analysis Accuracy Factor
C 1.0 ....................................... Damage Accumulation Model Accuracy
C 0.00 0.00 .................................. TFM Model Accuracy
C 0.20 950.0 .................................. Nominal Mech. Strain & Rotor Speed (% , RPM)
C 6 ....................................... Strain-Time History Period (Missions)
C 0.0 ....................................... Strain-Time History Noise Filter (%)
C 0.5 ....................................... Number of Points in Strain-Time History
C 0.00 0.00 .................................. 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
A B C D E F
0.00727362 0.000067442 -0.000059109 -3.52929E-08 1.07611E-08 -2.74419E-08

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
A B C D E F
-0.132623 0.000227427 -0.000059290 0.00 0.00 4.71714E-08

COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR TIME:
Fdecel2(m,Tstart) = A + (Tstart - B) / m
A B
0.20 950.0

COEFFICIENTS FOR SHUTDOWN RESPONSE SURFACE FOR RPM:
Fdecel3(t) = A + B * t
A B
30523.07 -21846.15

RPM(TIME) THERMAL STRAIN (%) ...... STRAIN HISTORY INFORMATION
225.8 0.0
3025.1 -0.196921
6138.8 0.146025
8309.0 -0.200128
0.0 0.007393
LIST OF VARIABLES

ALLM() 2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
BIGK1 EQUAL TO BIGK(1) – DUMMY PARAMETER FOR CALLS TO SUBROUTINE
BLDDLIF REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND LCF LIFE CALCULATION
BLFPER() 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, BETAO, CHARACTERIZING S/N DATA SET
DUM DUMMY VARIABLE
EBEND SELECTED VALUE FOR BENDING STRAIN (%)
EBENDA EBEND LOWER BOUND
EBENDB EBEND UPPER BOUND
EMNOM NOMINAL MECHANICAL STRAIN (%)
ETHNOM() 1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIME HISTORY
FAA, FAB, FAC, FAD, FAE, FAF COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
FACTR SCALE FACTOR EQUAL TO PHI * KRATIO * Z
FAERRM DECELERATION THERMAL STRAIN RESPONSE SURFACE MEAN
FAERRS DECELERATION THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
FD2A, FD2B COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
FD3A, FD3B COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
FDERRM DECELERATION THERMAL STRAIN RESPONSE SURFACE MEAN
FDERRS DECELERATION THERMAL STRAIN RESPONSE SURFACE STANDARD DEV.
FIFTY EQUAL TO .5 – USED TO ACCESS 50% POINT IN LIFE()
FTU MATERIAL ULTIMATE STRENGTH (%)
FTY MATERIAL YIELD STRENGTH (%)
HGAS SELECTED HOT GAS FILM COEFFICIENT, Hgas
HGASA HGAS LOWER BOUND
HGASB HGAS UPPER BOUND
HGASR SELECTED RHO FOR HGAS
HGASR1 HGAS - RHO LOWER BOUND
HGASR2 HGAS - RHO UPPER BOUND
HGASR SELECTED THETA FOR HGAS
HGAST SELECTED THETA LOWER BOUND
HGAST1 HGAS - THETA LOWER BOUND
HGAST2 HGAS - THETA UPPER BOUND
I CONTROLS INNER DO LOOP
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
LAMA SELECTED COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR, Lambda
LAMAA LAMA LOWER BOUND
LAMAB LAMA UPPER BOUND
LAMDA SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, Lambda
Damage Accumulation

LAMDA
LAMDA LOWER BOUND
LAMDB
LAMDA UPPER BOUND
LAMG
SELECTED UNCERTAINTY IN Tgas
LAMGA
LAMG LOWER BOUND
LAMGB
LAMG UPPER BOUND
LAMGR
SELECTED RHO FOR LAMG
LAMGR1
LAMG - RHO LOWER BOUND
LAMGR2
LAMG - RHO UPPER BOUND
LAMTS
SELECTED THETA FOR LAMG
LAMTS1
LAMG - THETA LOWER BOUND
LAMTS2
LAMG - THETA UPPER BOUND
LAMP
SELECTED DEVIATION IN BLADE PULL DUE TO BLADE MASS, LAMbda Pull
LAMPA
LAMP LOWER BOUND
LAMPB
LAMP UPPER BOUND
LAMTM
SELECTED TMF MODEL ACCURACY FACTOR, LAMbda TMf
LAMTM2
LAMTM LOWER BOUND
LAMTM2
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
LNP
NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
LN2
NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
LPHIM
1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
M
CONTROLS SYMMETRY DO LOOP
MANAL
SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
MANALB
SELECTED 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 USED TO CALCULATE THE TRUNCATED NORMAL
MEDM()
1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
MIN
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
MODER1
MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
MODER2
MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
MPROC
MATERIALS PROCESS VARIATION - CONTROLS MATERIALS PROCESS
LANG
VARIATION - 0 - NO VARIATION; 1 - VARIATION
MU()
1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
NBLIFE
NUMBER OF BLIVES TO BE PROVIDED
NBND()
1-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
NHYP
SIZE OF OUTER LOOP
NLIF
SIZE OF INNER LOOP
NLTET
TOTAL NUMBER OF LIVES CALCULATED BY PFM
NMED
CONTROLS MEDIAN CALCULATION FOR THE TRUNCATED NORMAL DISTRIBUTION CASE - 0 - NO MEDIAN CALCULATION; 1 - MEDIAN CALCULATION DESIRED
NOMSPD
NOMINAL ROTOR SPEED, RPM
NPSMT
1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
SYSTEM
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(BETAo, ETAo) GENERATED RANDOM VARIATE
PHISIG
EQUAL TO PT * (5 ** .5) / BZERO - VALUE OF LOGNORMAL PARAMETER, SIGMA, CHARACTERIZING S/N DATA SET
PSIG
EQUAL TO SQRT(PVAR) - MATERIALS PROCESS STANDARD DEVIATION
PVAR
MATERIALS PROCESS VARIATION
RAND
RANDOM NUMBER SEED
RANGEM()
2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON M FOR
FUNCTION BLDLIF PERFORMS THE DRIVER TRANSFORMATION AND CALLS RAINF3
TO CALCULATE THE FATIGUE LIFE
PROGRAMMER: L. NEWLIN
DATE: CODE: 7JAN92 COMMENTS: 3APR92
VERSION: BLDLCP 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, MODER2, FD2A, 
& FD2B, FD3A, FD3B, ETHNOM, MANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, 
& ENNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, 
& WALKER Exponent
& LOGNORMAL(0, PVAR) GENERATED RANDOM VARIATE
& ZERO Region -- Values chosen to facilitate region DO loop
& BEGINNING VALUE = 0 - NO ZERO Region EXISTS, 1 - NO ZERO Region

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 (% = -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, So
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
TGBB 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
TRSSBD() 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
TSTRSIG THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS 
VARPHI CONTROLS TYPE OF INTRINSIC MATERIALS VARIATION DESIRED -- 
1 - WEIBULL VARIATION; 2 - LOGNORMAL VARIATION 
VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION; 
1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUN-
CATED NORMAL VARIATION
WEXP WALKER EXPONENT
Z LOGNORMAL(0, PVAR) GENERATED RANDOM VARIATE
ZPZERO Zero Region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
& BEGINNING VALUE = 0 - NO 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: BLDLCP 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, MODER2, FD2A, 
& FD2B, FD3A, FD3B, ETHNOM, MANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, 
& ENNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, 
& WALKER Exponent
& LOGNORMAL(0, PVAR) GENERATED RANDOM VARIATE
& ZERO Region -- Values chosen to facilitate region DO loop
& BEGINNING VALUE = 0 - NO ZERO Region EXISTS, 1 - NO ZERO Region

7 - 111
IMPLICIT NONE

INTEGER MAXM, MAXREG
PARAMETER (MAXM = 50, MAXREG = 3)

COMMON IOUT
  INTEGER I, IOUT, NTIME, NUMREG, TSUBI, ZROREG
  REAL  BLDLIF, EBEND, EM(MAXM), EMNOM, ETH(MAXM), ETHNOM(MAXM),
       ETOT(MAXM), FAA, FAB, FAC, FAD, FAE, FAF, FD1, FD1A, FD1B, FD1C, FD1D, FD1E, FD1F,
       FD2, FD2A, FD2B, FD3, FD3A, FD3B, HGAS, KRATI0, LAMA, LAMG, LAMP, MM(0:MAXREG), LNZ, LPHIM(0:MAXREG), MANAL,
       MM(0:MAXREG), MODER1, MODER2, NOSPD, PERIOD, RAINF3,
       TRUNC, PERIOD, ZROREG, NUMREG, NZERO, TRSBND, TSTART, WEXP

C LIST OF VARIABLES
C
C EBEND  SELECTED VALUE FOR BENDING STRAIN (%)
C EM()   1-D ARRAY CONTAINING THE SIMULATED MECHANICAL STRAIN-TIMe
C EMNOM  NOMINAL MECHANICAL STRAIN (%)
C ETH()  1-D ARRAY CONTAINING THE SIMULATED THERMAL STRAIN-TIMe HISTORY (%)
C ETHNOM() 1-D ARRAY CONTAINING THE NOMINAL THERMAL STRAIN-TIMe HISTORY
C ETOT()  1-D ARRAY CONTAINING THE TOTAL STRAIN-TIMe HISTOY
C FA     VALUE OF ACCELERATION FUNCTION FOR THERMAL STRAIN — SECOND ORDER POLYNOMIAL AS A FUNCTION OF TGAS AND HGAS
C FAA, FAB, FAC, FAD, FAE, FAF  COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C FD1   VALUE OF DECELERATION FUNCTION FOR THERMAL STRAIN — SECOND ORDER POLYNOMIAL AS A FUNCTION OF FA, THE ACCELERATION FUNCTION
C FD1A, FD1B, FD1C, FD1D, FD1E, FD1F  COEFFICIENTS FOR FD1, ONE OF THE DECELERATION FUNCTIONS
C FD2   VALUE OF DECELERATION FUNCTION FOR TIME — SECOND ORDER POLYNOMIAL AS A FUNCTION OF FA, THE ACCELERATION FUNCTION
C FD2A, FD2B  COEFFICIENTS FOR FD2, ONE OF THE DECELERATION FUNCTIONS
C FD3   VALUE OF DECELERATION FUNCTION FOR ROTOR SPEED — FIRST ORDER POLYNOMIAL (LINEAR) FUNCTION OF TIME
C FD3A, FD3B  COEFFICIENTS FOR FD3, ONE OF THE DECELERATION FUNCTIONS
C HGAS   SELECTED HOT GAS FILM COEFFICIENT, Hgas
C I     CONTROLS DO LOOP FOR EACH POINT IN TIME HISTORY
C IOUT  CONTROLS DUMP TO FILE IOUTPR
C KRATI0 RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C LAMA   SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR, Lambda Alpha
C LAMG   THE UNCERTAINTY IN Tgas
C LAMP  SELECTED VALUE FOR DEVIATION IN BLADE PULL DUE TO BLADE MASS, LAMbda Pull
C LNA()  1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNZ   NORMAL(0,PFAR) GENERATED RANDOM VARIABLE
C LPHIM() 1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C MANAL  SELECTED VALUE FOR MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
C MAXM   MAXIMUM NUMBER OF POINTS ALLOWED IN TIME HISTORY
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MM()   1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MODER1 MODEL ERROR FOR STARTUP THERMAL STRAIN RESPONSE SURFACE
C MODER2
MODEL ERROR FOR DECELERATION THERMAL STRAIN RESPONSE SURFACE
C NOMSPD
NOMINAL ROTOR SPEED, RPM
C NTIME
NUMBER OF POINTS IN STRAIN-TIME HISTORY
C NUMREG
NUMBER OF REGIONS OF INTEREST
C PERIOD
LENGTH OF TIME IN MISSIONS OF TIME HISTORY
C RAINF3
REAL FUNCTION PERFORMING RAINFLOW COUNTING, DAMAGE ACCUMULATION AND FATIGUE LIFE PREDICTION (USING THE MATERIALS CHARACTERIZATION MODEL)
C RPM()
1-D ARRAY CONTAINING ROTOR SPEED HISTORY
C SLOPE
SELECTED VALUE FOR DECELERATION SLOPE, deg R / sec
C SPEED
SELECTED VALUE FOR STEADY STATE ROTOR SPEED, rpm
C SZERO
STRAIN TENSILE TEST POINT, So
C TANAL
SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C TGAS
SELECTED VALUE FOR HOT GAS TEMPERATURE Tgas (deg R)
C TRSBND()
1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR EACH REGION CALCULATED FOR EACH TRIAL
C TSUBI
VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)
C TSTART
THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS
C ZROREG
ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C WEXP
WALKER EXPONENT
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. TSUBI)) & 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, ' RPM = ', RPM(NTIME)
WRITE(8,*) 'ETH(I) = ', ETH(I), ' ETOT = ', ETOT(I)
DO 125 I = 1, NTIME
WRITE(8,*) 'I = ', I, ' EM = ', EM(I)
125 CONTINUE
ENDIF

C ** CALL RAINF3 TO CALCULATE DAMAGE AND RESULTING FATIGUE LIFE

BLDLIF = 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

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(TH1,TH2)
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.
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,*J 'FRAC =', FRAC
X = FRAC * (RHO2 - RHO1) + RHO1

CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE(8,*J 'FRAC =', FRAC
Y = FRAC * (THE2 - THE1) + THE1

IF (IOUT .EQ. 15) WRITE(8,*J 'RHO1 =', RHO1, ' RHO2 =', RHO2, &
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

IF (IOUT .EQ. 15) WRITE(8,*J 'RAND =', RAND, ' RHO =', RHO, &
' THETA =', THETA, ' A =', A, ' B =', B, ' X =', X,
Y1 = GAM((RHO * THETA + 1.), RAND)
Y2 = GAM(((1. - RHO) * THETA + 1.), RAND)

7 - 115
\[
W = \frac{Y_1}{(Y_1 + Y_2)}
\]

TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

\[
X = W \cdot (B - A) + A
\]

REAL FUNCTION GAM (ALPHA, RAND)

\[
A = \text{ALPHA} - 1.
\]

CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALLOG(U1)
V2 = - ALLOG(U2)
GAM = ALPHA * V1
RETURN
END

SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
FOR THE STRESS FORMULATION
PROGRAMMER: L. NEWLIN
DATE: 13JUL89
VERSION: MATCHR V8.4, V8.5 MATGRM V4.4, V4.5
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)

INPUTS: READS DATA FROM SPECIFIED AND RELATED; VARY, MPROC
OUTPUTS: RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG,
NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
SUBPROGRAMS: INIT, RCE, SW2SU2, FINDMC, INTRVL, FNRNG, ADDRGN, ADDREG,
CONCAV, MEDIAN, EXPCTD, MUSIG, NORM, ADDRGN, GTPVAR
FILE: 5:RELATD-OLD; 6:RELATO-NEW

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, LNNF(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), RATSTR(MAXDAT, 0:MAXSET),
& RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
& SIG(MAXDAT), SIGMA2(MAXREG), STR(MAXDAT, MAXREG),
& SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(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 SX/(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
LAMN 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
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION BOUNDARY 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() 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
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)
NBND() IS THE UPPER BOUND
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

NPTR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP(i)-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

RAWSTR() 2-D ARRAY CONTAINING RAW CYCLES TO FAILURE 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 %)

SUHAT2() 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)

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)

STRESS TENSILE TEST POINT, SO

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

ZZERO REGION - VALUES CHOSEN TO FACILITATE REGION DO LOOP 
BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION EXIST

OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
OPEN(6, FILE = 'RELATO', STATUS = 'NEW')

RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION

RELA TO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET INFORMATION

PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION

INITIALIZE PRIMARY ARRAYS

CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP, & NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

READ, CONVERT, ECHO INFORMATION

CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, & LNNF, REFNP, STR, NF, SZERO, ZZERO, NNUMRES, NNODAT, & NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, & SIGMA2, KRAZIO, LAMN)

CALCULATE RESIDUAL VARIANCES

CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD, & SWHAT2, SUHAT2, NPTR)
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, SWHAT2, SUHAT2, NPPR, IZERO, JZERO, MCHAT)
    ENDIF

    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 ADDEG (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)
    CONTINUE
    WRITE(7,910)
    DO 50 L = 1, NUMREG
      WRITE(7,915) L, MCHAT(2,L), MCHAT(I,L)
    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
      CONTINUE
    ENDIF
    WRITE(7,920)
    WRITE(7,930)
    DO 100 L = 1, NUMREG
      WRITE(7,940) L, RANGEM(I,L), RANGEM(2,L)
    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

7 - 119
CALL EXPCETD (1, MEDM, REFPN, STR, NF, SZERO, NUMREG, ZROREG, 
& 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
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. i) 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 = i, NUMREG
    WRITE(7,980) L, MCHAT(I,L)
  CONTINUE
IF (CZERO .GT. 0.0) THEN
  WRITE(7,960)
  DO 360 L = I, 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 = i, NUMREG
    WRITE(7,940) L, RANGEM(I,L), RANGEM(2,L)
  CONTINUE
WRITE(7,950)
WRITE(7,985)
  DO 380 L = I, NUMREG
    WRITE(7,990) L, MU(L), SIG(L)
  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.',//,
  & 2X,'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)',12X,'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)

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',
  & //,2X,'ESTIMATE 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

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
SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR,
& REFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)
C INPUTS:
C OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
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 OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
K CONTROLS DO LOOP FOR EACH DATA SET
L CONTROLS DO LOOP FOR EACH REGION
NPTS() 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
NFS() 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
2-D ARRAY CONTAINING RAWFN() 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
      RATSTR(I,J) = 0.0
350 CONTINUE
300 CONTINUE

DO 400 L = 1, MAXREG
   DO 425 K = 1, MAXDAT
      DO 450 J = 0, MAXSET
         LNSTR(K,J,L) = 0.0
         LNNF(K,J,L) = 0.0
450 CONTINUE
425 CONTINUE
400 CONTINUE

DO 500 L = 1, MAXREG
   DO 550 K = 1, MAXDAT
      NF(K,L) = 0.0
      STR(K,L) = 0.0
550 CONTINUE
500 CONTINUE

DO 600 L = 1, MAXREG
   REFNP(L) = 0
   MPNT(L) = 0
   MZERO(1,L) = 0.0
   MZERO(2,L) = 0.0
   DELTA(L) = 0.0
   MO(L) = 0.0
   SIGMA2(L) = 0.0
600 CONTINUE

RETURN
END

******************************************************************************************************

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 (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, &
                 LINSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
NUMBER OF REGIONS OF INTEREST
STRESS RATIO (R = -1.0 IS DESIRED)
2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS
2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
REGION OF INTEREST IN A PARTICULAR DIVISION
2-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
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
ZERO REGION – VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE – 0 – ZERO REGION EXISTS, 1 – NO ZERO REGION

COUNT = 0
DO 10 L = 0, MAXREG
NBND(L) = 0.0
10 CONTINUE

READ(1,*) 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. 10) WRITE(8,905)

STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ
FTUZ = FTU
FTYZ = FTY

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
CALL TRMNAT
ENDIF
IF (REG .GT. MAXREG) THEN
WRITE(8,*) 'ERROR: OVER REGION LIMIT IN SPECIFIC DATA SET'
CALL TRMNAT
ENDIF

7 - 125
DO 110 I = (COUNT + 1), (COUNT + NUM)
READ(1,*) RAWSTR(I,0), RAWNF(I,0)
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)
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)
130 CONTINUE

C BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2, C
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
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(1,*) NBND(L)
DO 160 L = 1, (NUMREG + NNODAT)
READ(1,*) MPNT(L), MZERO(1,L), MZERO(2,L)
CONTINUE
WRITE(3,913)
IF (ZROREG .EQ. 0) WRITE(3,914) SZERO
IF (IOUT .EQ. 10) THEN
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)
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
ENDIF
CONTINUE
180 CONTINUE
C 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)
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
CONTINUE
190 CONTINUE
ENDIF
C IF (MPROC .EQ. 1) THEN
READ(1,*) KRATIO, LAMN
WRITE(3,955) KRATIO, LAMN
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 = (COUNT + 1), (COUNT + NUM)
    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
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)
& 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 ',
& IF (IOUT .EQ. 10) 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 ',
& '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)
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',& /*,'CYCLES':*/',')
920 FORMAT(10X,E9.3)
925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,& 'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,& /,2X,'EXPICIT CONSTRAINT ON m FOR EACH REGION:',& /,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',& 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 COMMENTS: 13JUL89

SUBROUTINE CONVRT (J, NUMI, NUM2, STR, RSTR, R, FTU, FTY)

C INPUTS: J, NUMI, NUM2, STR, R, FTU, FTY
C OUTPUTS: RSTR
C IMPLICIT NONE

INTEGER MAXDAT, MAXSET
PARAMETER (MAXDAT = 50, MAXSET = 5)
COMMON IOUT
INTEGER I, IOUT, J, NUMI, NUM2
REAL FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
& STR(MAXDAT, 0:MAXSET), TEST

C LIST OF VARIABLES
C FTU ULTIMATE STRENGTH OF MATERIAL (PSI)
C FTY YIELD STRENGTH OF MATERIAL (PSI)
C I CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C IOUT OUTPUT DUMP CONTROLLER
C J DATA SET OF INTEREST
C MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C NUMI FIRST INDEX TO BE TRANSFORMED
C NUM2 LAST INDEX TO BE TRANSFORMED
C R STRESS RATIO (R = -1.0 IS DESIRED)
C RSTR() STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
C 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

RETURN
END
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,*),I: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*************************************/

SUBROUTINE SW2SU2 CALCULATES, WHAT2, THE RESIDUAL VARIANCES OF Y ON X
AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS

PROGRAMMER: L. NEWLIN
DATE: CODE: 6OCT87 COMMENTS: 13JUL89
VERSION: MATCHR V6, V6.1, V6.2, V6, V7, V7.1, V8, V8.1, V8.2, V8.3,
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 IMPPLICIT NONE
INTEGER MAXDAT, MAXREG, 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), SWHAT2(MAXREG),
& SX2(MAXREG), SYX(MAXREG), SY2(MAXREG)

LIST OF VARIABLES

BB() 1-D ARRAY CONTAINING SXY(L)/SY2(L) FOR EACH REGION
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
DIFFX() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L)
C DIFFY() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L) AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C IOUT OUTPUT DUMP CONTROLLER
C J CONTROLS DO LOOP FOR EACH DATA SET
C K CONTROLS DO LOOP FOR EACH POINT IN A REGION
C L CONTROLS DO LOOP FOR EACH REGION
C LNNF() 3-D ARRAY CONTAINING LN(Rawnf()) , ALSO INDEXED FOR REGION
C LNSTR() 3-D ARRAY CONTAINING LN(Ratstr()) , ALSO INDEXED FOR REGION
C MAXDAT MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C MEANX() 1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION L AND DATA SET J (X = Ln S)
C MEANY() 1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION L AND DATA SET J (Y = Ln N)
C NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN 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 NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
C NUMREG NUMBER OF REGIONS OF INTEREST
C SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
C SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR EACH REGION (X = Ln S, Y = Ln N)
C SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION (Y = Ln N)
C INITIALIZE ARRAYS

C 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
BB(L) = 0.0
DD(L) = 0.0
NPPR(L) = 0
50 CONTINUE
C DO 60 J = 0, MAXSET
DO 70 K = 1, MAXDAT
DIFFY(K,J) = 0.0
DIFFX(K,J) = 0.0
70 CONTINUE

C NOW PERFORM CALCULATION OF SX2, SY2, SXY, SWHAT2, SUHAT2 FOR EACH REGION

C DO 200 J = 0, NSETS
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. 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) = LN DF(K,J,L) - MEANY(J)
  DIFFX(K,J) = LNST R(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
300 CONTINUE

NPPR(L) = NPPR(L) + NP(J,L) - 1
IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', 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 = i, 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) * DIFFY(K,J)) ** 2
    IF (IOUT .EQ. 10) WRITE(8,*) 'K =', K, ' SWHAT2(L) =',
    & ' SUHAT2(L) =', SUHAT2(L)
500 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),

& ' WHAT2(L) =', WHAT2(L), ' SUHAT2(L) =', SUHAT2(L)
100 CONTINUE

RETURN
END

C SUBROUTINE INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, IO, ON
C 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, V7, V8, V8.1, V8.2, V8.3,
& V8.4, V8.5 ' '

SUBROUTINE INTRVL

C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
& JZERO, MCHAT
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
& 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 TO 75) /
& 0.000982699, 0.506356, 0.215795, 0.484449, 0.831211,
& 1.237347, 1.68977, 2.17973, 2.70039, 3.24697,
& 3.811575, 4.40379, 5.00874, 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,
& 21.39, 22.14, 22.89, 23.65, 24.4311,
& 25.21, 25.99, 26.78, 27.57, 28.36,
& 29.15, 29.95, 30.75, 31.55, 32.3574,
& 33.16, 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 (CHI025(I), I = 76 TO 150) /
& 53.78, 54.62, 55.46, 56.30, 57.152,
& 57.80, 58.64, 59.49, 60.34, 61.30,
& 62.24, 63.09, 63.94, 64.79, 65.6466,
& 66.50, 67.35, 68.21, 69.07, 69.92,
& 70.76, 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,
& 88.08, 88.95, 89.82, 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, 103.85, 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 /

7 - 134
### DATA (CHI975(I), I = l, 75) /

<table>
<thead>
<tr>
<th>DATA</th>
<th>CHI975(I)6210</th>
<th>76</th>
<th>1</th>
<th>0.2389</th>
<th>7.3776</th>
<th>9.34840</th>
<th>11.1433</th>
<th>12.8325</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>21.9200</td>
<td>23.3367</td>
<td>24.7356</td>
<td>26.1190</td>
<td>27.4884</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>21.1943</td>
<td>30.1910</td>
<td>31.5264</td>
<td>32.8523</td>
<td>34.1606</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>35.4789</td>
<td>36.7807</td>
<td>38.0757</td>
<td>39.3641</td>
<td>40.6465</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>41.9232</td>
<td>43.1944</td>
<td>44.4607</td>
<td>45.7222</td>
<td>46.9792</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>48.23</td>
<td>49.48</td>
<td>50.72</td>
<td>51.96</td>
<td>53.20</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>54.44</td>
<td>55.67</td>
<td>56.89</td>
<td>58.12</td>
<td>59.34</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>60.56</td>
<td>61.77</td>
<td>62.99</td>
<td>64.20</td>
<td>65.41</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>66.62</td>
<td>67.82</td>
<td>69.02</td>
<td>70.22</td>
<td>71.42</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>72.61</td>
<td>73.81</td>
<td>75.00</td>
<td>76.19</td>
<td>77.38</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>78.57</td>
<td>79.75</td>
<td>80.93</td>
<td>82.12</td>
<td>83.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>84.48</td>
<td>85.65</td>
<td>86.83</td>
<td>88.00</td>
<td>89.18</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>90.35</td>
<td>91.52</td>
<td>92.69</td>
<td>93.86</td>
<td>95.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>96.19</td>
<td>97.35</td>
<td>98.52</td>
<td>99.68</td>
<td>100.84</td>
<td></td>
</tr>
</tbody>
</table>

### DATA (CHI975(I), I = 76, 150) /

<table>
<thead>
<tr>
<th>DATA</th>
<th>CHI975(I)6210</th>
<th>76</th>
<th>1</th>
<th>0.2389</th>
<th>7.3776</th>
<th>9.34840</th>
<th>11.1433</th>
<th>12.8325</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>102.00</td>
<td>03.16</td>
<td>104.31</td>
<td>105.47</td>
<td>106.62</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>107.78</td>
<td>108.94</td>
<td>110.09</td>
<td>111.24</td>
<td>112.39</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>113.54</td>
<td>114.69</td>
<td>115.84</td>
<td>116.99</td>
<td>118.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>119.28</td>
<td>120.43</td>
<td>121.57</td>
<td>122.72</td>
<td>123.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>125.00</td>
<td>126.14</td>
<td>127.28</td>
<td>128.42</td>
<td>129.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>130.70</td>
<td>131.84</td>
<td>132.98</td>
<td>134.11</td>
<td>135.25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>136.38</td>
<td>137.52</td>
<td>138.65</td>
<td>139.79</td>
<td>140.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>142.05</td>
<td>143.18</td>
<td>144.31</td>
<td>145.44</td>
<td>146.58</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>147.70</td>
<td>148.83</td>
<td>149.96</td>
<td>151.09</td>
<td>152.21</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>153.34</td>
<td>154.47</td>
<td>155.59</td>
<td>156.72</td>
<td>157.84</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>158.97</td>
<td>160.09</td>
<td>161.21</td>
<td>162.33</td>
<td>163.46</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>164.58</td>
<td>165.70</td>
<td>166.82</td>
<td>167.94</td>
<td>169.06</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>170.18</td>
<td>171.30</td>
<td>172.41</td>
<td>173.53</td>
<td>174.65</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>175.77</td>
<td>176.88</td>
<td>178.00</td>
<td>179.12</td>
<td>180.23</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>181.35</td>
<td>182.46</td>
<td>183.58</td>
<td>184.69</td>
<td>185.80</td>
<td></td>
</tr>
</tbody>
</table>

### 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

### LIST OF VARIABLES

<table>
<thead>
<tr>
<th>ARG</th>
<th>INTERMEDIATE CALCULATION VARIABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHI025()</td>
<td>TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION</td>
</tr>
<tr>
<td>CHI975()</td>
<td>TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION</td>
</tr>
<tr>
<td>CHITAB</td>
<td>MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975</td>
</tr>
<tr>
<td>DD()</td>
<td>1-D ARRAY CONTAINING SYX(L)/SX2(L) FOR EACH REGION</td>
</tr>
<tr>
<td>I</td>
<td>CONTROLS LOOP FOR CHI025() AND CHI975()</td>
</tr>
<tr>
<td>IOUT</td>
<td>OUTPUT DUMP CONTROLLER</td>
</tr>
<tr>
<td>IZERO()</td>
<td>2-D ARRAY CONTAINING IZ, THE 95% CONFIDENCE INTERVALS ON C FOR EACH REGION</td>
</tr>
<tr>
<td>JZERO()</td>
<td>2-D ARRAY CONTAINING J0, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION</td>
</tr>
<tr>
<td>L</td>
<td>CONTROLS DO LOOP FOR EACH REGION</td>
</tr>
<tr>
<td>MAXREG</td>
<td>MAXIMUM NUMBER OF REGIONS ALLOWED</td>
</tr>
<tr>
<td>MCHAT()</td>
<td>2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION BASED ON MATERIALS DATA ONLY – MCHAT(L) = -DD, THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C</td>
</tr>
<tr>
<td>NPR(1)</td>
<td>1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-I))-I) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)</td>
</tr>
<tr>
<td>NUM</td>
<td>EQUAL TO 2*NPPR() FOR A SET OF CALCULATIONS</td>
</tr>
<tr>
<td>NUMREG</td>
<td>NUMBER OF REGIONS OF INTEREST</td>
</tr>
<tr>
<td>SUHAT</td>
<td>EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS</td>
</tr>
</tbody>
</table>

7 - 135
C SUHAT2()  1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
REGRESSION FOR EACH REGION (X = Ln S,  Y = Ln N)
SWHAT  EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
SWHAT2()  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)
T025() VALUE OF T025() USED IN CALCULATIONS
TTAB  TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C INITIALIZE IZERO, JZERO AND MCHAT

DO 50 L = i, MAXREG
IZERO(I,L) = 0.0
IZERO(2,L) = 0.0
JZERO(I,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 = I, 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
C CONFIDENCE INTERVALS FOR EACH REGION

DO 100 L = I, 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), ' SWHAT2 =', 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) =',
7 - 136
C** SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C THE CO GIVEN BY THE USER
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 8OCT87 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
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
    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
      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
  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)
  CONTINUE
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, V8.4, 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, NP(0:MAXSET, MAXREG) NSETS, NUM(MAXREG),
 & NUMREG, TOTAL
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), 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

RETURN
END
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)
    WRITE (8, *) 'MCHAT = ', MCHAT(2, L), ' PSIG2 = ', PSIG2(L)
 300 CONTINUE
ENDIF
PVAR = SUM / FLOAT(TOTAL)
RETURN
END

C*****************************************************************************

C SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C M AND CO WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 2FEB88  FORMAT/COMMENTS: 12AUG91
C VERSION: MATCHR V6.1, V6.2, V7, V8, V8.1, V8.2, V8.3,
C V8.4, V8.5

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
C MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN M() FOR EACH REGION
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN M() 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
   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(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
      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
   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
      WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
         &
         WRITE(8,*), 'JZERO(2,L) = ', JZERO(2,L)
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*), 'MC(1,L) = ', MC(1,L),
         &
         WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
         &
         WRITE(8,*), 'JZERO(2,L) = ', JZERO(2,L),
         &
         WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
         &
         WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
         &
         WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,L)
      ENDIF
   ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN
   ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN
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 J0 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(1,L),
  WRITE (8,*), 'JZERO(2,L) = ', JZERO(2,L),
  WRITE (8,*), '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),
  WRITE (8,*), '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),
  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. 0)) THEN
  THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT USE INTERSECTION BETWEEN J0 AND M0
  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 J0 AND M0'
    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,
  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 INTERSECT J0 AND M0, 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 J0, M0, MC'
    CALL TRMNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF
ENDIF
IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'JZERO(I,L) = ', JZERO(I,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),
  & ' 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. 2) .AND. (MCFNT(L) .EQ. 2) THEN
  THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO C0 CONSTRAINT
  INTERSECT THESE TWO RANGES WITH J0
  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 J0, M0, '
    & '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
ENDIF

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(8,*) 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
    & ' ON m IN REGION ', L
300 CONTINUE
RETURN
END

C************************************************************
SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS WITHOUT DATA
PROGRAMMER: L. NEWLIN
DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91
VERSION: MATCHR V6.1, 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
C 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)

C LIST OF VARIABLES
C IOUT
C L
C LL
C MAXREG
C MCHAT ( )
C MPNT( )
C MZERO ( )
C NNODAT
C NUMREG
C RANGEM ( )

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,*)'RANGEM(1,LL) =', RANGEM(1,LL),
             & ' MZERO(1,LL) =', MZERO(1,LL),
             & 'MZERO(2,LL) =', MZERO(2,LL)
         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
                 SPECIFY E(M) OF POSTERIOR FOR SAKE OF
                 CALCULATIONS IN SUBROUTINE EXPCTD
             ENDIF
         ENDIF
100 CONTINUE
MCHAT(I,LL) = (RANGEM(I,LL) + RANGEM(2,LL)) / 2.0
ENDIF
IF (IOUT .EQ. io) 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**************************************************************
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
& V8.4, V8.5.
SUBROUTINE CONCAV (NUMREG, RANGEM)
C
INPUTS: NUMREG, RANGEM
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT
C
IMPLICIT NONE
INTEGER MAXREG
PARAMETER (AXRGL = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL RANGEM(2, MAXREG), TESTM

C
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(1,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 CONVEXITY ADJUSTMENT
ADJUST RANGE TO INSURE CONVEXITY
DO 100 L = NUMREG, 2, -1
C
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. I0)
WRITE(8,*)
'RANGEM(ILL-I ) =,, RANGEM(IzL_I),
'RANGEM(2,L-I)=', RANGEM(2,L-I)
WRITE(8,*)
'RANGEM(IZL) =,, RANGEM(lzL),
'RANGEM(2,L) =', RANGEM(2,L)
WRITE(8,*)
'TESTM =', TESTM, ' L = ', L
ENDIF
I00 CONTINUE
RETURN
END

C**************************************************************
C SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER JO HAS
C BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR CO
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 5OCT87 COMMENTS: IDEC87
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)
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
C FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C 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
      LOWERM = AMAX1(RANGEM(1,L), MEDM(L-1))
      MEDM(L) = (LOWERM + 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

C 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, BIGKI, BZAT)
INPUTS: NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
OUTPUTS: BIGKI, BZAT
SUBPROGRAMS: TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

IMPLICIT NONE
INTEGER IOUT, MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG
REAL BIGK(0:MAXREG), BIGKI, BZAT, 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
- 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, $S_0$
- TRBIGK(): 1-D array containing values of K in this routine
- ZROREG: zero region value chosen to facilitate region DO loop
- 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 FINDSBS (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 (ZZERO, BIGK, MM, NBND, SBND, TRBIGK,
             & FACTR, NUMREG)
   ENDIF

C WRITE RESULTS TO FILE

IF (NCOMPS .EQ. 1) THEN
   WRITE(7,900) NUMREG, BZHAT, KHA T
   IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZHAT, KHA T
   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), KHA T
ENDIF

FORMAT STATEMENTS

900 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',/,,2X,
         & 'NUMBER OF REGIONS',/,'I4,5X','E(BETA)=','F8.4,5X','E(K)=',
         & 'F8.4,2X','REGION','7X','m','15X','K','9X','LIFE BOUND','7X',
         & 'STRESS BOUND',//)

910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X, E9.3,9X,E11.5)

920 FORMAT(///)

930 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
         & 'I11X','m','14X','K','13X','E(K)',//,
         & '7X,F8.5,5X,E12.5,6X,F7.4,//)

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
C
SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA,
       & MCHAT, MU, SIG)
C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2
C OUTPUTS: MCHAT, MU, SIG
C IMPLICIT NONE

7 - 149
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
VARIANCE DISTRIBUTION STANDARD DEVIATION 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 = NPPR(L) * SX2(L)
SWHAT2() EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(X = Ln S)

C INITALIZE 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
C THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C USE THE ESTIMATE OF M
MU(L) = MCHAT(1,L)
ELSE
C 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
C THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
C
C 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_) 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),
& WRITE(8,*), 'SUMX2 = ', SUMX2(L), ' DELTA = ', DELTA(L), ' ARG = ', ARG
WRITE(8,*), 'MO = ', M(L), ' MU = ', M(L),
& 'SIG = ', SIG(L), ' SIGMA2 = ', SIGMA2(L),
& 'SUHAT2 = ', SUHAT2(L), ' MCHAT2 = ', MCHAT(2,L)
& WRITE(8,*), 'NPR = ', NPPR(L), ' SX2 = ', SX2(L),
& WRITE(8,*), 'SUMX2 = ', SUMX2(L), ' DELTA = ', DELTA(L), ' ARG = ', ARG
WRITE(8,*), 'MO = ', M(L), ' MU = ', M(L),
& 'SIG = ', SIG(L), ' SIGMA2 = ', SIGMA2(L),
& 'SUHAT2 = ', SUHAT2(L), ' MCHAT2 = ', MCHAT(2,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

SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C OUTPUTS: RANGEM
C 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

C IOUT OUTPUT DUMP CONTROLLER
C L CONTROLS DO LOOP FOR EACH REGION
C LOWER LOWER BOUND OF INTERSECTION
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION
C CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
C MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
C MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
C FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN C
C MC() FOR EACH REGION
C MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN C
C 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 ((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,*), 'L = ', L, ' 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 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,*), 'L = ', L, ' MZERO(1,L) = ', MZERO(1,L),
WRITE(8,*), 'MC(1,L) = ', MC(1,L),
WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
      ' RANGEM(2,L) = ', RANGEM(2,L)
ENDIF
C
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 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
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
300 CONTINUE
RETURN
END
C***********************************************************************

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 FORMAT/COMMENTS: 12AUG91
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(MAX_G), 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
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))
ENDIF

SPECIFY E(M) OF POSTERIOR FOR SAKE OF
CALCULATIONS IN SUBROUTINE EXPCTD
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
AND IF (IOUT .EQ. IO) WRITE(8,*,'(MCHAT =', MCHAT(I,LL),
& 'MU = ', MU(LL), ', SIG = ', SIG(LL)
& 'ERROR: OVERALL PRIOR RANGE INCORRECTLY SPECIFIED IN REGION WITHOUT DATA')
& CALL TRMNAT
ENDIF
100 CONTINUE
RETURN
END

C******************************************************************************

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

MAXREG V4.3, V4.4, V4.5

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)

INPUTS: VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
& NBND, STR

OUTPUTS: BIGK, BZERO, MM, SBND

SUBPROGRAMS: FINDM, FINDMN, TRNSFM, 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), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
& RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(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, BETA0, 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

7 - 155
SAMPLE MEAN OF TRANSFORMED DATA, \( Z = F(\text{STR, NF, NBND, MM}) \)

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 %)

CONTROLS TYPE OF CURVE VARIATION DESIRED — 0 - NO VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION

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, \( Z = F(\text{STR, NF, NBND, MM}) \)

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
WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)

WRITE(7,920)

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',I5Xo,K,.gX.,LIF
    _n_n_
    _v
    _. &
    'STRESS BOUND'//)

910 FORMAT(5X,I1,5X,F9.5,SX,E12.5,5X,E9.3,6X,E11.5)

920 FORMAT(///)

RETURN
END

C SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION
C BY SAMPLING OFF THE APPROPRIATE M RANGE
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88 COMMENTS: 13JUL89
C VERSION: MATCHR V8, VS.1, V8.2, V8.4, V8.5
C
SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C INPUTS: RAND, NUMREG, RANGEM
C OUTPUTS: MM
C 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
C
C LIST OF VARIABLES
C
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

7 - 157
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)
         WRITE(8,*) '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
      ENDIF
      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(I, L) =', RANGEM(I, L),
         & PICK(1) --', PICK(1),
         & PICK(2) =', PICK(2)
         WRITE(8,*) 'RANGEM(1, L) =', RANGEM(1, L),
         & RANGEM(2, L) =', RANGEM(2, L),
         & X =', X,
         & MM(L) =', MM(L)
      ENDIF
   ENDIF

100 CONTINUE

RETURN

END

*************************************************************************

*************************************************************************

SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
UNIFORMLY DISTRIBUTED RANDOM NUMBERS

Miles, R. F., The RANDOM Computer Program: A Linear Congruential
Random Number Generator, JPL Publication 85-98, JPL Document

PROGRAMMER:  L. GRONDAHLKI, L. NEWLIN
DATE:  IDEC87
VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5,
         MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
         V4.3, V4.4, V4.5

*************************************************************************

SUBROUTINE RANDOM (FRAC, RAND)
IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL FRAC
DOUBLE PRECISION RANA, RANC, RAND, RANDIV, RANM, RANSUB,
& RANT, RANX

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 = 709085.0
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(*,*) 'RAN = ', RANX, ' RANDIV = ', RANDIV,
& ' RANT = ', RANT, ' RANSUB = ', RANSUB, ' RAND = ', RAND,
& ' FRAC = ', FRAC
RETURN
END

C NOTES: IOUT=2 DUMPS TO SCREEN

**********************************************************************

SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
PROGRAMMER: L. NEWLIN
DATE: CODE: 7JUN88 COMMENTS: 13FEB89
VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)

INPUTS: RAND, NUMREG, MU, SIG, RANGEM
OUTPUTS: MM
SUBPROGRAMS: NORMGN, TRMNAT

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
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 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
        MM(L) = RANGEM(1,L)
        & ELSEIF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
        ' RANGEM(2,L) =', RANGEM(2,L),
        ' MM(L) =', MM(L)
    ELSE
        SAMPLE ON EXISTING RANGE
        10 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),
            ' X =', X, ' MM(L) =', MM(L)
        ENDIF
        ELSE
            ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
            AND THEN SAMPLE
            20 CALL NORMGN (RAND, MU(L), SIG(L), X)
            IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
            MM(L) = X
            IF (IOUT .EQ. 10) THEN
                WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2),
                ' MM(L) =', MM(L)
            ENDIF
            ELSE
                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
                IF (IOUT .EQ. 10) THEN
                    WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2),
                    ' MM(L) =', MM(L)
                ENDIF
            ENDIF
SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

IMPLICIT NONE

COMMON IOUT

DOUBLE PRECISION RAND, FRAC, 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
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: 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
C 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)

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)
100 CONTINUE
IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =', MML, ' NPTS =', NPTS(L)

DO 200 K = 1, 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, ' 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-I) =', NBND(LL-I), ' MM(LL-I) =', MM(LL-I), ' ZZ =', ZZ(NP)

CONTINUE

CONTINUE

I00 CONTINUE
RETURN
END

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
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,

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(*,*)'NP =', NP, 'I =', I,
    & ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
100 CONTINUE
MEANZ = MEANZ / FLOAT(NP)
```

IF (IOUT .EQ. I0) WRITE(*,*)' MEANZ =', MEANZ

**C**

CALCULATE THE VARIANCE OF ZZ(), SZ2

```
DO 200 I = 1, NP
    SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
    IF (IOUT .EQ. I0) WRITE(*,*)'I =', I, ' SZ2 =', SZ2
200 CONTINUE
SZ2 = SZ2 / FLOAT(NP - 1)
```

IF (IOUT .EQ. I0) WRITE(*,*)' SZ2 =', SZ2

RETURN

END

**C**

SUBROUTINE KBETA CALCULATES k AND BETAO FROM THE SAMPLE MEAN AND VARIANCE OF Z = F(STR, NF, NBND, MM)

PROGRAMMER: L. NEWLIN

DATE: CODE: 6OCT87 COMMENTS: 13JUL89


SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)

**C**

**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

**C**

LIST OF VARIABLES

**C** BZERO

VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE SPECIFIC MATERIAL S/N DATA SET

**C** IOUT

OUTPUT DUMP CONTROLLER

**C** K

VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATABASE

**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**

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******************************************************************************

C SUBROUTINE FINDK CALCULATES THE VALUE OF K, WHERE A = K ** M FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: 7JUN88
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C
C SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)
C
C INPUTS:  BZERO, K, MM, NBND, NUMREG
C OUTPUTS: BIGK
C
C IMPLICIT NONE
C INTEGER MAXREG
C REAL GAMMA
C PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)
C COMMON IOUT
C INTEGER IOU, L, NUMREG
C REAL BIGK(0:MAXREG), BZERO, K, MM(0:MAXREG), NBND(0:MAXREG)

C LIST OF VARIABLES

C BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
C BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C GAMMA EULER'S CONSTANT
C IOU OUTPUT DUMP CONTROLLER
C K VALUE OF K -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL DATA BASE
C L CONTROLS DO LOOP FOR EACH REGION
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
C 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)
C WRITE(7,*) 'REGION: i, 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)))
C WRITE(7,*) 'REGION : L, ' 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
C VALUES WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE
C RANDOMLY SELECTED M$, 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, V8.3, V8.4, V8.5
C***************************************************************
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
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
CONTINUE

CALCULATE SBND(0) IF ZROREG = 0

IF (ZROREG .EQ. 0) THEN
SBND(0) = BIGK(1) * NBND(0) ** (-1.0 / MM(1))
ENDIF

CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES

DO I00 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
I00 CONTINUE
RETURN
END

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
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
COMMON IOUT
INTEGER IOUT
REAL ARG, BETA, ETA, FRAC, WEIB
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 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.
C PROGRAMMER : L. NEWLIN
C DATE: 1AUG91
C VERSION: MATCHR V8.5  MATRM 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, L, NUMREG
REAL BIGK(MAXREG), FACTR, MM(MAXREG), NBND(MAXREG), 1
SCLK, SZERO, TRBIGK(MAXREG), TRSBND(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 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()

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

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

IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'SZERO = ', SZERO,
    WRITE(8,*) 'FACTOR = ', FACTR,
    WRITE(8,*) 'BIGK0 = ', TRBIGK(0)
    WRITE(8,*) 'MM0 = ', MM(0)
ENDIF

RETURN
END

C**FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
BASD 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), LPHIM(0:MAXREG),
     MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

LIST OF VARIABLES

GTLIFE 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
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
C ZROREG  Zero Region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C BEGINNING VALUE -- 0 - ZEROS 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
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 Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

SUBROUTINE SORTM (ALLM, NUMREG, NUM)
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
C NUM
C NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C NUMREG
C NUMBER OF REGIONS OF INTEREST
C TEMP
C TEMPORARY SORTING VARIABLE

DO 400 L = 1, NUMREG
5 INC = NUM
10 IF (INC .GT. 1) THEN
20 INORDR = .TRUE.

DO 300 I = 1, (NUM - INC)
300 CONTINUE

IF (.NOT. INORDR) GOTO 20

GOTO 10
ENDIF

400 CONTINUE

RETURN
END

C***********************************************************************
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 (BDDLCF 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)
C INPUTS: SEFF, M, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO,
C OUTPUTS: RAINF3
C 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), GTXLIFE, INVLIF(MAXM), KRATIO,
& LIFE(MAXM), LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
& MM(0:MAXREG), PERIOD, RAINF3, S(MAXM), SBND(0:MAXREG),
& SEFF(MAXM), SEFFM(2, MAXM), SEFMAX, SP(MAXM),
& SRANGE(MAXM), SUNDA, 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(1,1) = maxmin,eff,1
SEFFM(2,1) = sigma min,eff,1
SRANGE(I) = EQUIVALENT STRAIN RANGE FOR CYCLE I
GTFLIFE REAL FUNCTION THAT CALCULATES FATIGUE LIFE FOR A GIVEN STRAIN LEVEL
LIFE(N) 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 dump controller

IOUT OUTPUT DUMP CONTROLLER
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
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
MNREG NUMBER OF REGIONS OF INTEREST
SBND() 1-D ARRAY CONTAINING THE STRAIN VALUES (%; R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN SBND() CORRECTED BY PHI, KRATIO, AND LNZ
SZERO STRAIN TENSILE POINT, So (%)
ZROREG Zero region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP

beginning value = 0 - zero region exists, 1 - no zero region

C dump input data
if (lout.eq.20) then
  write(8,*)' rainf3 inputs'
  write(8,*)' period: ',period
  write(8,*)' m : ',m, ' period: ',period
  write(8,*)' wexp : ', wexp
  write(8,*)' numreg : ', numreg
  write(8,*)' zroreg : ', zroreg
  write(8,*)' szero : ', szero
  write(8,*)' kratio : ', kratio
  write(8,*)' lnz : ', lnz
  write(8,*)' lna(i), mm(i), lphim(i), sbnd(i)
  write(8,*) (lna(i), mm(i), lphim(i), sbnd(i), i=zroreg,numreg)
endif

C INITIALIZE ARRAYS

DO 50 I = 1, MAXM
  SP(I) = 0.0
  S(I) = 0.0

50 continue

7 - 172
**RESERQURCE mortgage strains (needed for run flow analysis)**

find SEFMAX, the largest $\sigma_{eff}$, and JMAX, its location within SEFF(M)

```
SEFMAX = -1.0E+20
```

```
DO 200 I=M,1
  IF (SEFF(I) .GT. SEFMAX) THEN
    SEFMAX = SEFF(I)
    JMAX = I
  ENDIF
```

assign all points from JMAX out, to the beginning of SP()

```
DO 210 I = M-JMAX+1, M-JMAX+I, I
  J = JMAX-I + I
  SP(I) = SEFF(J)
```

assign points before JMAX to the end of SP()

```
J = 0
DO 220 I = 1, M-JMAX, 1
  J = J + 1
  SP(M+I) = SEFF(JMAX)
```

**FILTER the resequenced effective strains, leaving only peaks and valleys**

(excursions larger than TRUNC are deleted during run flow counting) in S(NEXTOT), where NEXTOT is the new number of points

```
DO 300 I = 2, M
  TEST1(I) = SP(I-1) - SP(I)
  TEST2(I) = TEST1(I) * (SP(I) - SP(I+1))
```

```
DO 305 I = 2, M
  IF ((TEST1(I) .LT. 0) .AND. (TEST2(I) .LT. 0)) THEN
    K = K + 1
  ENDIF
```

```
NEXTOT = K + 1
INDEX(NEXTOT) = 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**************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
C check J to avoid reading beyond end of filtered strain data
  IF ( J .GT. NEWTOT ) GOTO 499
C read strain point into a holding array to be checked for cycles
  E(K) = S(J)
410 IF ( K .LT. 3 ) GOTO 400
C if not, then a cycle has been found, but we need to check for truncation
  IF (ABS (E(K) - E(K-1)) .LT. ABS(E(K-1) - E(K-2)) ) GOTO 400
C cycle is large enough to save
  I = I+1
  SEFFM(1,I) = AMAX1( E(K-1), E(K-2) )
  SEFFM(2,I) = AMINI( E(K-1), E(K-2) )
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) : '(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(1,I) - SEFFM(2,I)) & *
    * ((SEFFM(1,I) - SEFFM(2,I)) / (2.0 * SEFFM(1,I)) & ** (WEXP - 1.0)
500 CONTINUE

if (iout.eq.20) write(8,*)'srnage(n) :',(srnage(I),i=1,n)
if (iout.eq.25) write(8,*) (srnage(i),i=1,n)
& exp(lphim(1)/mm(1))

C calculate lives and damage fractions: LIFE(N) and INVLIF(N)
DO 600 I=1,N
   LIFE(I) = GTLIFE (SRANGE(I), MM, LNA, LPHIM, KRATIO, LNZ,
   & SBND, ZROREG, NUMREG, ZZERO)
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),,14 continue
   endif
C Miner's Rule -- sum the damage fractions
SUMDAM = 0.0
DO 700 I=1,N
   SUMDAM = SUMDAM + INVLIF(I)
700 CONTINUE
if (iout.eq.20) write(8,*)'sumdam:,,sumdam
C calculate fatigue life (time to failure)
RAINF3 = PERIOD / SUMDAM
if (iout.eq.15) then
   chkft=period/sumdam
   write(8,*), rainf3life',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>Program BLDLCF V3.4B1.3 Listing Temporal Order</td>
<td>7-177</td>
</tr>
<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**********************************************************************
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
C VERSION: 3.4BI.3 (MATCHR VBI.3, RAINF3 VI.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**********************************************************************

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 IOUT
INTEGER BLFPOS(MAXBLF), I, IOUT, J, K, L, MCOUNT, MID, MPROC,
& NLIFE, NHYPER, NLIFE, NLIFET, NMED,
& NPTS(MAXREG), NSYM, NTIME, NUMREG, TSUBI, VARY,
& ZROREG
DOUBLE PRECISION RAND
REAL ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGKI, BLDLIF,
& BLFPER(MAXBLF), BZERO, DUM, EBEND, EBENDA, EBENDB,
& EMNOM, EPSL, EPSW, ETHNOM(MAXM), FAA, FAB, FAC, FACTR,
& FAD, FAE, FAF, FAERR, FAERRS, FD1A, FD1B, FD1D,
& FD2A, FD2B, FD3A, FD3B, FDERRM, FDERRS,
& FITY, FTV, HGS, HGSASA, HGSBA, HGSBR, HGSRI,
& HGASR2, HGAST, HGAST1, HGAST2, KRAT1, LAMA, LAMAA,
& LANAB, LAMDA, LAMDAa, LAMDAB, LAMG, LAMGA, LAMGB, LAMGR,
& LAMR1, LAMR2, LAMT1, LAMT2, LAMP, LAMPA,
& LAMPB, LAMTM, LAMTMA, LAMTMB, LIFEL(MAXLIF),
& LIFEW(MAXLIF), LIFL, LIFW, LNA(0:MAXREG), LINZ,
REALLPHIM(O:MAXREG), MANAL, MANALA, MANALB, MEDKB(O:MAXREG),
& MEDM(MAXREG), MEDMB(0:MAXREG), MM(0:MAXREG), MODER1,
& MODER2, MU(MAXREG), NBDN(0:MAXREG), NEWLIF,
& NF(MAXDAT, MAXREG), NOMSPD, PERIOD, PHI, PSIG, PVAR,
& RANGEM(2, MAXREG), RESID(MAXDAT), RPM(MAXM),
& SMB(0:MAXREG), SIG(MAXREG), SLOPE, SLOPEA, SLOPEB,
& SLOPR, SLOPR1, SLOPR2, SLOPT, SLOPT1, SLOPT2, SPEED,
& SPEEDM, SPEEDS, STR(MAXDAT, MAXREG), SZERO, TANAL,
& TANALA, TANALB, TGAS, TGA3, TGA3B, TGA3R, TGA3R1,
& TGAS2, TGA2, TGA3T, TGA3T, TBIGK(0:MAXREG),
& TRSBDN(0:MAXREG), TRUNC, TSTART, TSTMU, TSTSIG, WEXP, Z
C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

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,* ) ' RANDOM NUMBER SEED = ', RAND
READ(1,* ) IOUT
WRITE(8,* ) IOUT (MATCHR = 10, BLDLCF = 15, RAINF3 = 20) = ', IOUT
READ(1,* ) NLIFE
WRITE(8,* ) ' INNER LOOP SIZE = ', NLIFE
READ(1,* )
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 ' 
WRITE(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 ' 
WRITE(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(1,*) NBLIFE
IF (NBLIFE .GT. 0) READ(1,*) (BLFPER(J), J = 1, NBLIFE)

C ** READ DATA FROM BLDLCD
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, LAMDAA, LAMDAB
& LAMTMA, LAMTMB
READ(1,*) ENNOM, NOMSPD, PERIOD, TRUNC, NTIME, WEXP
READ(1,*) FAA, FAB, FAC, FAD, FAE, FAF, 
& FDIA, FDIB, FDIC, FDID, FDIE, FDIF, 
& 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 BLDLCD
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, 
& EXP(LAMDAA), EXP(LAMDAB), EXP(LAMTMA), EXP(LAMTMB)
WRITE(3,905) FAA, FAB, FAC, FAD, FAE, FAF, 
& FD1A, FD1B, FD1C, FD1D, FD1E, FD1F, 
& FD2A, FD2B,
& FD3A, FD3B

WRITE(3,906) ENNOM, NOMSPD, PERIOD, TRUNC, NTIME, WEXP
WRITE(3,907) FAA, FAB, FAC, FAD, FAE, FAF, 
& FD1A, FD1B, FD1C, FD1D, FD1E, FD1F, 
& FD2A, FD2B,
& FD3A, FD3B

DO 25 I = 1, (NTIME - 1)
    WRITE(3,908) RPM(I), ETHOM(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, SZERO, ZROREG, NUMREG, 
& NBND, STR, FTU, PTY, VARY, MPROC, KRATIO, PVAR, 
& MEDMB, MEDKB, RESID)

IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)

C ** INITIALIZE VARIABLES

MCOUNT = 0

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, HGASR1, HGASR2, HGAST1, HGAST2, HGASR, HGAST)
CALL PRYRV (RAND, TOASR1, TOASR2, TOAST1, TOAST2, TOASR, TOAST)
CALL PRYRV (RAND, SLOPR1, SLOPR2, SLOPT1, SLOPT2, SLOPR, SLOPT)
CALL PRYRV (RAND, LAMGR1, LAMGR2, LAMGT1, LAMGT2, LAMGR, LAMGT)
CALL PRYRV (RAND, MANAL1A, MANALB, TANALA, TANALB, MANAL, TANAL)

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 ** 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 ** 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, SLOPCA, SLOPEB, SLOPE)
CALL BETAGN (RAND, LAMGR, LAMGT, LAMGA, LAMGB, LAMG)
CALL NORMGN (RAND, SPEEDM, SPEEDS, SPEED)
CALL NORMGN (RAND, FAERRM, FAERRS, MODEM)
CALL NORMGN (RAND, TSTMU, TSTSIG, TSTART)
CALL NORMGN (RAND, FDERRM, FDERRS, MODE2)
CALL PRYRV (RAND, EBENDA, EBENDB, LAMPA, LAMPB, EBEND, LAMP)
CALL PRYRV (RAND, LAMDA, LAMDB, LAMTA, LAMTB, LAMDA, LAMTM)
LAMDA = EXP (LAMDA)
LAMTM = EXP (LAMTM)
CALL WORSTN (RAND, NSYM, BZERO, MM, EPSW, EPSL)

IF ((VARY .EQ. 0) .OR. (VARY .EQ. 4)) PHI = 1.0

IF (IOUT .EQ. 15) THEN
    WRITE(8,*) 'HGAS =', HGAS, ' TGAS =', TGAS
    WRITE(8,*) 'SLOPE =', SLOPE, ' LAMG = ', LAMG
    WRITE(8,*) 'LAMP =', LAMP, ' EBEND =', EBEND, ' LAMA =', LAMA
    WRITE(8,*) 'SPEED = ', SPEED,' LAMDA =',LAMDA
    WRITE(8,*) 'LAMTM =',LAMTM,' PHI =',PHI
    WRITE(8,*) 'MODER1 =',MODER1,
    WRITE(8,*) '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

ENDIF

C ** CALL BLDLIF TO OBTAIN BLADE LCF LIFE

NEWLIF = LAMDA * LAMTM * BLDLIF (TGAS, HGAS, FAB, FAC, FAD, FAE, FAF, MODER1, RPM, TSUBI, SPEED, SLOPE,
TSTART, FDI1A, FDI1B, FDI1C, FDI1D, FDI1E, FDI1F,
MODER2, FDI2A, FDI2B, FDI2A, FDI2B, FDI2B, FDI2B, LAME, NOMSPD, EMNO, TANAL, LAMA, LAMG, EBEND,
NTIME, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, ZRIG, LNZ, TRSBND, ZRREG, NUMREG, SZERO)

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,
                  NBOUND, BIGK1, 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,')')
903 FORMAT(//,50X,'N( MEAN, STD. DEV.)',&10X,'N(ROTOR SPEED VARIATION (rpm) AT TIME T',II,&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,')')

904 FORMAT(//,2X,'STRAIN DUE TO GAS BENDING (%), 17X,'U(FS.5,','F9.5,')',&2X,'LAMBDA BLADE PULL',29X,'U(FS.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(FS.5,','F9.5,')')

905 FORMAT(//,2X,'DAMAGE MODEL ACCURACY',23X,'U(in',F8.5,','F9.5,')',&2X,'TMF MODEL ACCURACY',6X,'U(In',F8.5,','F9.5,')')

906 FORMAT(///,2X,,'OTHER STRAIN HISTORY INPUT:',&2X,'NOMINAL MECHANICAL STRAIN (%),',23X,F6.4,&2X,'NOMINAL ROTOR SPEED (rpm)',23X,F6.0,&2X,'STRAIN-TIME HISTORY PERIOD (missions)',14X,F5.2,&2X,'STRAIN-TIME HISTORY NOISE FILTER (%),',16X,F7.5,&2X,'NUMBER OF POINTS IN HISTORIES',19X,F5.4,&2X,'WALKER EXPONENT',36X,F5.2)

907 FORMAT(///,6X,'COEFFICIENTS OF ACCELERATION AND DECELERATION',&'FUNCTIONS',//,2X,'THERMAL STRAIN AT STARTUP (%)':,5X,&'Facce1(Tgas, Hgas) = ',E13.6,' + ',E13.6,' * Tgas + ',&15X,E13.6,' * Hgas + ',E13.6,' * Tgas * 2 + ',&15X,E13.6,' * Hgas * 2 + ',E13.6,' * Tgas * Hgas',&2X,'THERMAL STRAIN AT SHUTDOWN (%)':,5X,&'Fdecel1(m, Tstart) = ',E13.6,' + ',E13.6,' * Tstart + ',&15X,E13.6,' * m + ',E13.6,' * Tstart * 2 + ',&15X,E13.6,' * m * 2 + ',E13.6,' * Tstart * m',&2X,'TIME AT SHUTDOWN (sec)':,5X,&'Fdecel2(m, Tstart) = ',E13.6,' + ',(Tstart - ',&13X,E13.6,' ) / m',&2X,'ROTOR SPEED AT SHUTDOWN (rpm)',:5X,'Fdecel3(t, = ',E13.6,' + ',E13.6,' * t',&///,20X,,'STRAIN HISTORY INFORMATION:',&///,5X,,'ROTOR SPEED',:5X,'THERMAL STRAIN',&///,9X,'rpm',:15X,('')/')

908 FORMAT(7X,F7.1,9X,F9.6)

925 FORMAT(///,2X,'B LIVES: WEIBULL LOGNORMAL')

926 FORMAT(2X,F7.5,5X,E13.6,5X,E13.6)

STOP END
C

0.0001

C

676.

............................

B.01

LIFE

c o.ool
.............................
O.Ol..............................
=FE
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
c
C
C
C
C
C
C
c
C
C
C

2730.

C

0.5

0.0

0.0 .... Hgas

(A,B)

(RI,R2)(TI,T2)

0.5
0.5
0.0
0.0
Tgas
(A,B)
(RI,_I
(TI,T2)
0.5
0.5
0.0
0.0. .....DECEL
SLOPE
(A,B)
(R1,
(T1,T2)
0.5
0.5
0.0
0.0 .... Tgas
UNCERTAINTY
FACTOR
507 ..................
ROTOR
SPEED
VARIATION
PARAMETERS:
i, MEAN,
STD. DEV.
(NORMAL
DIST.)
0.0
0.020 ......................
Faccel
MODELING
ERROR
MEAN
& STD.DEV.
1640.0
40.67 ....................
DECEL
Tstart
MEAN
& STANDARD
DEVIATION
975.3
28.6 .....................
STANDARD
RESPONSE
PROBE
MEAN
& STD DEV
0.0
0.003 .......................
Fdecel
MODELING
ERROR
MEAN
& STD DEV
0.0
0.0 ......................
STRAIN
DUE TO GAS BENDING
(%)
0.96
1.04 .....................
LAMBDA
BLADE
PULL
0.80
1.20 .....................
MECHANICAL
ANALYSIS
ACCURACY
FACTOR
0.975
1.025 ....................
COEFFICIENT
OF THERMAL
EXPANSION
0.70
1.30 .....................
THERMAL
ANALYSIS
ACCURACY
FACTOR
-0.693147
0.563283
..............
DAMAGE
ACCUMULATION
MODEL
ACCURACY
0.00
0.00 ..................
TMF MODEL
ACCURACY
0.295
38482 .....................
NOMINAL
MECH.
STRAIN
& ROTOR
SPEED
(%,RPM)
i 0 ...............................
STRAIN-TIME
HISTORY
PERIOD
(MISSIONS)
0 000 .............................
STRAIN-TIME
HISTORY
NOISE
FILTER
(%)
6 .................................
NUMBER
OF POINTS
IN STRAIN-TIME
HISTORY
0.5 ...............................
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
A
B
C
D
E
F
0.00727362
0.000067442
-0.000059109
-3.52929E-08
1.07611E-08
-2.74419E-08
COEFFICIENTS
Fdecell(m,

FOR SHUTDOWN
Tstart)
= A

A
-0.132623

B
0.000227427

COEFFICIENTS
Fdecel2(m,
A
0.20

FOR SHUTDOWN
Tstart)
= A
B
950.0

COEFFICIENTS
Fdecel3(t)
A
30523.07

FOR SHUTDOWN
RESPONSE
= A + B * t
B
-21846.15

RPM(TIME)
225.8
3025.1
6138.8
8309.0
0.0

c 'ia_ _A
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

0.5

800.
2000.
2730.
2730.
0.80
1.20
5
37592.

THERMAL
STRAIN
0.0
-0.196921
0.146025
-0.200128
0.007393

RESPONSE
SURFACE
+ B * Tstart
+ C
+ E * m ** 2 + F
C
-0.000059290

+

RESPONSE
(Tstart

SURFACE
FOR
- B) / m

TIME:

SURFACE

RFM:

(%) ..... STRAIN

1480, 001 DIRECTION'. ....MATERL_

1.57
1
8 ...............
8
-I.0
1 .....................
0.89
6800 ....................
0.89
15000 ....................
0.67
27000 ....................
0.67
43200 ....................
0.56
139300 ....................
0.56
545200 ....................
0.56
147000 ....................
0.39
4344800
....................
0.00 ..............................
1
0 ..............................
1.0E+36
...........................
0.00 ..............................
0
0.00
0.00 ..................
0.0
0.0
0.0 ...................
LIST

OF

FOR THERMAL
STRAIN:
* m + D * Tstart
** 2
* Tstart
* m
D
E
F
0.00
0.00
4. 71714E-08

FOR

HISTORY

INFORMATION

DESC_ION

YIELD
& ULTIMATE
STRENGTHES,
NDIV,
NPTS
# PTS IN DIV,
STRAIN
RATIO,
REGION
S(1)
N(1)
RAW
S(2)
N(2)
STRAIN-LIFE
S(3)
N(3)
(S/N)
S(4)
N(4)
DATA
S(5)
N(5)
POINTS
S(6)
N(6)
FOR THE
S(7)
N(7)
SPECIFIC
S(8)
N(8)
MATERIAL
NO VALUE
OF So SUPPLIED
(%)
NUMBER
OF REGIONS:W/DATA
W/O DATA
LIFE
BOUNDARIES:
REGION
1
CONSTRAINT
ON COEFF.
OF VARIATION
0 PTS IN RANGE,
LOWER
BOUND,
UPPER
BOUND
NORMAL
DIST.
PRIORS:
DELTA,
Mo, SIGMA2

VARIABLES

7 - 184


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 BIGK1 - DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C EXPECTD BLDDLIF REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND LCF
C LIFE CALCULATION
C BLFPER{} 1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS{ } I-D ARRAY CONTAINING POSITION IN LIFE() OF EMPIRICAL BLIVES
C BZERO WEIBULL SHAPE PARAMETER, BETAO, 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, PAF
C FACTR SCALE FACTOR EQUAL TO PHI * KRATIO * Z
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 LIFE() AND LIFEW()
C HGAS SELECTED HOT GAS FILM COEFFICIENT, Hgas
C HGASB HGAS LOWER BOUND
C HGASR SELECTED RHO FOR HGAS
C HGASR1 HGAS - RHO LOWER BOUND
C HGASR2 HGAS - RHO UPPER BOUND
C HGAST SELECTED THETA FOR HGAS
C HGAST1 HGAS - THETA LOWER BOUND
C HGAST2 HGAS - THETA UPPER BOUND
C I OUT CONTROLS DUMP TO FILE IOUTPR
C J CONTROLS DO LOOP FOR EACH BLIFE
C KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C L CONTROLS 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 Damage Accumulation
C LAMDA{ } LAMDA LOWER BOUND
C LAMDB LAMDA UPPER BOUND
C LAMG SELECTED UNCERTAINTY IN Tgas
C LAMGA LAMG LOWER BOUND
C LAMGB LAMG UPPER BOUND
C LAMGR SELECTED RHO FOR LAMG
C LAMGR1 LAMG - RHO LOWER BOUND
C LAMGR2 LAMG - RHO UPPER BOUND
C LAMGT SELECTED THETA FOR LAMG
C LAMGT1 LAMG - THETA LOWER BOUND
C LAMGT2 LAMG - THETA UPPER BOUND
C LAMP SELECTED DEVIATION IN BLADE PULL DUE TO BLADE MASS, Lambda Pull
C LAMPA LAMPA LOWER BOUND
C LAMPB LAMPA UPPER BOUND
C LAMTM SELECTED TMF MODEL ACCURACY FACTOR, Lambda Tmf
C LAMTMA LAMTM LOWER BOUND
C LAMTMB LAMTM UPPER BOUND
C LIFE() 1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
C USING THE LOGNORMAL DISTRIBUTION - SORTED VALUES OF THE
LEFT-HAND TAIL

1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM USING THE WEIBULL DISTRIBUTION – SORTED VALUES OF THE LEFT-HAND TAIL

LIFL

MISSIONS TO FAILURE BASED ON EPSL

LIFW

MISSIONS TO FAILURE BASED ON EPSW

LNA()

1-D ARRAY CONTAINING \( \ln(A) = \ln(BIGK) \ast MM \) FOR EACH REGION

LPHIM()

1-D ARRAY CONTAINING \( \ln(\text{PHI}) \ast MM \) FOR EACH REGION

MANAL

SELECTED MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR

MANALA

MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR LOWER BOUND

MANALB

MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR UPPER BOUND

MAXLIF

MAXIMUM NUMBER OF LIVES TO BE PROVIDED

MAXDAT

MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED

MAXLF

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

MAXREG

MAXIMUM NUMBER OF REGIONS ALLOWED

MCOUNT

NUMBER OF M's TO BE USED TO CALCULATE MEDIAN S/N CURVE

MEDKB

1-D ARRAY CONTAINING THE MEAN K VALUES FOR EACH REGION

(BOOTSTRAP OPTION)

MEDMB

1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION

(BOOTSTRAP OPTION)

MEDM()

1-D ARRAY CONTAINING THE MEDIAN M VALUES FOR EACH REGION

MID

POINTER TO THE MEDIAN M VALUES – EQUAL TO HALF OF MCOUNT

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

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

NBLIFE

NUMBER OF LIVES TO BE PROVIDED

NBND()

1-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 BDLIF

NF()

2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS

NYPER

SIZE OF OUTER LOOP

NLIPE

SIZE OF INNER LOOP

NLIPE

TOTAL NUMBER OF LIVES CALCULATED BY PFM

NMED

CONTROLS MEDIAN CALCULATION FOR THE NORMAL 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(\( \beta_o, \eta_o \)) GENERATED RANDOM VARIATE

PSIG

EQUAL TO SQRT(PVAR) – MATERIALS PROCESS STANDARD DEVIATION

PVAR

MATERIALS PROCESS VARIATION

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

RESID()

1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH POINT IN THE SPECIFIC MATERIAL S/N DATA SET

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

SLOPR2

m - RHO LOWER BOUND

SLOPR2

m - RHO LOWER BOUND

SLOPT

SELECTED THETA FOR m

SLOPT1

m -theta LOWER BOUND

SLOPT2

m -theta UPPER BOUND
FUNCTION BLDLIF PERFORMS THE DRIVER TRANSFORMATION AND CALLS RAINF3 TO CALCULATE THE FATIGUE LIFE.

PROGRAMMER: L. NEWLIN

DATE: CODE: 7JAN92 COMMENTS: 17APR92

VERSION: BLDLIF 3.4 (MATCHR v6.5, RAINF3 v1.1)

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

SUBPROGRAMS: RAINF3

INPUTS: TGAS, HGAS, FAA, FAB, FAC, FAD, FAE, FAF, MODER1, RPM, TSUBI, SPEED, SLOPE, TSTART, FDIA, FDIB, FDIC, FD1D, FD1E, FD1F, MODER2, FD2A, FD2B, FD2D, FD3A, FD3B, ETHNOM, MANAL, LAMP, NOMSPD, EMNOM, TANAL, LAMA, LAMG, EBEND, NTIME, TRUNC, PERIOD, WEXP, MM, LNA, LPHIM, KRATIO, LNZ, TRSBND, ZROREG, NUMREG, SZERO

OUTPUTS: BLDLIF

IMPLICIT NONE

PARAMETER (MAXX = 50, MAXREG = 3)
COMMON IOUT

INTEGER I, IOUT, NTIME, NUMREG, TSUBI, ZROREG

REAL BDLIF, EBEND, EM(MAXM), EMNOM, ETH(MAXM), ETHNOM(MAXM),
& ETOT(MAXM), FA, FAA, FAB, FAC, FAD, FAE, PAF, FD1,
& FD1A, FD1B, FD1C, FD1D, FD1F, FD2, FD2A, FD2B,
& FD3, 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:AXR G), TRUNC, TSTART, WEXP

LIST OF VARIABLES

C I-D ARRAY CONTAINING THE SIMULATED MECHANICAL STRAIN-TIME HISTORY (%)
C I-D ARRAY CONTAINING THE SIMULATED THERMAL STRAIN-TIME HISTORY (%)
C I-D ARRAY CONTAINING THE TOTAL STRAIN-TIME HISTORY
C ORDER POLYNOMIAL AS A FUNCTION OF Tgas and HGAS
C COEFFICIENTS FOR FA, THE ACCELERATION FUNCTION
C ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
C ONE OF THE DECELERATION FUNCTIONS
C ORDER POLYNOMIAL AS A FUNCTION OF m, THE DECELERATION SLOPE
C ONE OF THE DECELERATION FUNCTIONS
C VALUE OF ACCELERATION FUNCTION FOR THERMAL STRAIN -- SECOND ORDER POLYNOMIAL
C THE UNCERTAINTY IN Tgas
C THE UNCERTAINTY IN ln(A) = ln(BIGK)*MM FOR EACH REGION
C SELECTED VALUE FOR MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
C SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C SELECTED VALUE FOR MECHANICAL STRAIN ANALYSIS ACCURACY FACTOR
C SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR
C SELECTED VALUE FOR DEVIATION IN BLADE PULL DUE TO BLADE MASS,
C SELECTED VALUE FOR BENDING STRAIN (%)
C SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR
C SELECTED VALUE FOR HOT GAS TEMPERATURE Tgas (deg R)
C SELECTED VALUE FOR DECELERATION SLOPE, deg R / sec
C SELECTED VALUE FOR STEADY STATE ROTOR SPEED, rpm
C STRAIN TENSILE TEST POINT, So
C SELECTED VALUE FOR CoEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR
C SELECTED VALUE FOR THERMAL STRAIN ANALYSIS ACCURACY FACTOR
C SELECTED VALUE FOR HOT GAS TEMPERATURE Tgas (deg R)
C SELECTED VALUE FOR COEFFICIENT OF THERMAL EXPANSION ACCURACY FACTOR
C VALUE USED TO FILTER OUT NOISE IN THE TIME HISTORY (%)
C THE TIME INDEX FOR WHICH VARIATION IN ROTOR SPEED OCCURS

7 - 188
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 * TGAS + FAC * HGAS + FAD * TGAS ** 2 & + FAE * HGAS ** 2 + FAF * 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. i) .AND. (I .LT. TSUBI))
    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, ' 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, & LPHI, KRATIO, LN2, TRSBND, ZROREG, NUMREG, & SZERO)

RETURN
END

C********************************************************************
C SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C PROGRAMMER: L. NEWLIN
C DATE: 20JUL90
C VERSION: 2.1
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA contract NAS7-918
C is acknowledged.
C
C SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)
C INPUTS: NEWLIF, LIFE, NLIFET
C OUTPUTS: LIFE
C IMPLICIT NONE
INTEGER MAXLIF

C********************************************************************
PARAMETER (MAXLIF = 10000)
COMMON IOUT
INTEGER I, IOUT, NLIFET, NUM, PLACE
REAL LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

LIST OF VARIABLES

I OUT 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()
C TEMP() 1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON 
INSERSION 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

*********************************************************
SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THET1,THET2) 
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 NAS7-918
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
Y = FRAC * (THE2 - THE1) + THE1
C IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ', RHO2 =', RHO2,
& THE1 =', THE1, ', THE2 =', THE2, ', X =', X, ', Y =', Y
RETURN
END

C THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
PROGRAMMER: L. GRONDALSKI, L. NEWLIN
DATE: 9MAR87
SUBPROGRAM: GAM

The random variates are generated using the method described in:
Univariate Distributions - 1, Houghton Mifflin Company, 1970,
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
C IF (IOUT .EQ. 15) WRITE(8,*) 'RAND =', RAND, ', RHO =', RHO,
& 'THETA =', THETA, ', A =', A, ', B =', B, ', X =', X
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”
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)

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,*) '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

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 VBI.2, VBI.3
MATGRM VBI, VBI.1

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 SPECTD 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, ADDREG, CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR, EXPB
FILES: 5:RELATD-OLD; 6:RELATD-NEW

Implicit None

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODET, NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET), NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG
REAL BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG), DELTA(MAXREG), FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
LIST OF VARIABLES

BIGKHT

BZERO

CZERO

DD

DELTA

FTUZ

FTYZ

IOUT

IZERO

JZERO

KRATIO

LAMN

LNNF

LNSTR

MAXDAT

MAXREG

MAXSET

MCHAT

MEDKB

MEDM

MEDMB

MO

MPNT

MPROC

MZERO

NBND

NF

NNODAT

NP

NPPR

RATSTR

RANGEM

SIG

SIGMA2

STR

SUHAT2

SWHAT2

X2

SXY

SY2

SZERO

BIGKHT

EQUAL TO THE MEDIAN VALUE OF K IN REGION 1

BZERO

VALUE OF WEIBULL PARAMETER, \( \beta_0 \), CHARACTERIZING THE S/N DATA SET

CZERO

EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, \( \sigma \)

DD

1-D ARRAY CONTAINING \( \frac{S_Y(L)}{S_X(L)} \), 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

2-D ARRAY CONTAINING \( I_0 \), THE 95% CONFIDENCE INTERVALS ON \( C \) FOR EACH REGION

JZERO

2-D ARRAY CONTAINING \( J_0 \), THE 95% CONFIDENCE INTERVALS ON \( M \) FOR EACH REGION

KRATIO

RATIO OF \( K^*/K \), CONSTANT OVER REGIONS AND COMPONENTS

LAMN

LAMBDA-N -- RATIO OF \( \text{VAR} (\ln N \text{ given } S) / (m**2 C^-2) \), CONSTANT OVER REGIONS AND COMPONENTS

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

MC

2-D ARRAY CONTAINING VALUES OF THE RANGES ON \( M \) FOR EACH REGION CONSISTENT WITH GIVEN VALUE OF \( \sigma \) AND THE DATA

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 \( C \)

MEDKB

1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION

MEDM

1-D ARRAY CONTAINING THE MEAN \( M \) VALUES FOR EACH REGION (BOOTSTRAP OPTION)

MEDMB

1-D ARRAY CONTAINING THE MEAN \( M \) VALUES FOR EACH REGION (BOOTSTRAP OPTION)

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

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)
I-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
MPVAR 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 (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
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
(SY2() (Y = Ln N)
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; 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')

RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET
C INFORMATION
C PERFORM CALCULATIONS COMMON TO UNIFORM, NORMAL, AND BOOTSTRAP
C TYPE OF VARIATION
C INITIALIZE PRIMARY ARRAYS
CALL INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNF,
& NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)
CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR,
& LNNF, REFNF, STR, NF, SZERO, ZROREG, NUMREG, NNODAT,
& NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTY2, DELTA, MO,
& SIGMA2, KRATIO, LAMN)
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
& SUHAT2, SUHAT2, NPPR, MEDMB, MEDKB, RESID)
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, 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
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 = i, NUMREG
         WRITE(7,905) L, IZERO(I, L), IZERO(2, L), JZERO(1, L), JZERO(2, L)
      CONTINUE
   WRITE(7,910)
      DO 50 L = i, NUMREG
         WRITE(7,915) L, MCHAT(2,L), MCHAT(1,L)
      CONTINUE
   IF (CZERO .GT. 0.0) THEN
      DO 150 L = i, NUMREG
         IF (MCPNT(L) .EQ. i) 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 = i, 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, 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, SUHAT2, NPPR, DELTA, MO, SIGMA2, MCHAT, MU, SIG)
    C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
    IF (MPROC .EQ. I) 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
        DO 360 L = 1, NUMREG
            IF (MCPNT(L) .EQ. I) 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(7,960)
    DO 380 L = 1, NUMREG
        WRITE(7,970) L, MU(I,L), SIG(L)
    CONTINUE
    ELSE
        C BOOTSTRAPPING IS REQUIRED
        WRITE(7,900)
    ENDIF
END
FIRST CALCULATE OTHER REGION PARAMETERS BASED ON THE EXPECTED M AND K VALUES

CALL EXPB (MEDMB, MEDKB, SZERO, NUMREG, ZROREG, 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,'Sponsorship under ',
& 'NASA Contract NAS7-918 is acknowledged.',//,7X,'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,')',
& 2X,'Io = (',F12.9,','),F12.9,')')

910 FORMAT(/,7X,'REGION',8X,'E(C)',12X,'E(m)',/)

915 FORMAT(9X,II,8X,FL1.9,5X,FL9.6)

920 FORMAT(/,7X,'POSTERIOR CREDIBILITY RANGES ON m FOR EACH REGION',
& ///,7X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

940 FORMAT(6X,II,8X,FS.4,8X,FS.4)

960 FORMAT(/,2X,'RANGE ON m FOR EACH REGION IMPLIED BY C CONSTRAINT',
& 'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

965 FORMAT(6X,II,8X,FS.4,8X,'INFINITY')

970 FORMAT(6X,II,8X,FS.4,8X,FS.4)

975 FORMAT(2X,'Copyright (C) 1990, California Institute of Technology. U.S. Government.',2X,'Sponsorship under ',
& 'NASA Contract NAS7-918 is acknowledged.',//,7X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
& //,2X,'ESTIMATES OF m FOR EACH REGION',
& //,7X,'REGION',12X,'E(m)',/)

980 FORMAT(9X,II,11X,FL10.6)

985 FORMAT(2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS',
& //,2X,'REGION',5X,'MEAN',8X,'STD DEV',/)

990 FORMAT(5X,II,5X,7.4,5X,E11.5)

995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N CURVE',
& 'WARRANTED BY THE AVAILABLE INFORMATION',/)

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: 5OC787


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: 21JUN88 COMMENTS: 13JUL89

VERSION: MATCHR


SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, & REFNF, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

INPUTS: 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)

C 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), & M(0: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

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(RANSTR()), 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

7 - 198
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 SPECFD AND RELATD. RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS C SPECIFIED BY USER

DO 100 J = 0, MAXSET
   NPTS(J) = 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
      RATSTR(I, J) = 0.0
350 CONTINUE
300 CONTINUE

DO 400 L = 1, MAXREG
   DO 425 K = 1, MAXDAT
      DO 450 J = 0, MAXSET
         LLNF(K, J, L) = 0.0
         LNSTR(K, J, L) = 0.0
450 CONTINUE
425 CONTINUE
400 CONTINUE

DO 500 L = 1, MAXREG
   DO 550 K = 1, MAXDAT
      NF(K, L) = 0.0
      STR(K, L) = 0.0
550 CONTINUE
500 CONTINUE

DO 600 L = 1, MAXREG
   REFNP(L) = 0
   MPNT(L) = 0
   MZERO(L) = 0.0
   MZERO(2, L) = 0.0
   DELTA(L) = 0.0
   MO(L) = 0.0
   SIGMA2(L) = 0.0
600 CONTINUE

RETURN
END
SUBROUTINE RCE

INPUTS:
VARY, MPROC
NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP, STR, NP, ZERO, ZROREG, NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN

OUTPUTS:
C VARY, MPROC
C NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP, STR, NP, ZERO, ZROREG, NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN

COMMON IOUT

INTEGER COUNT, I, IOUT, J, K, M, MPNT(MAXREG), MPROC, NDIV,
NODAT, NP(0 MAXSET, MAXREG), NPTS(0 MAXSET), NSETS,
NUM, NUMREG, REFNP(MAXREG), REG, VARY, ZROREG

REAL CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ,
KRATIO, LAMN, LNNF(MAXDAT, 0 MAXSET, MAXREG), MO(MAXREG),
ZZERO(2, MAXREG), NBND(0 MAXREG), NF(MAXDAT, MAXREG),
RATIO, RATSTR(MAXDAT, 0 MAXSET), RAWNF(MAXDAT, 0 MAXSET),
RAWSTR(MAXDAT, 0 MAXSET), SIGMA2(MAXREG),
STR(MAXDAT, MAXREG), ZZERO

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
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 MATERIAL TESTING STRENGTH 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 OVER ALL REGIONS AND COMPONENTS
LNNF() 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(I,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
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
ZEROREG ZeRO REGION - VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE - 0 - ZERO REGION EXISTS, 1 - NO ZERO
REGION

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), 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)
WRITE(3,905)
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(8,*') 'ERROR: INVALID VALUE FOR RATIO: ', RATIO
   ENDIF
100 CONTINUE
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)
   READ(1,*), RAWSTR(I,0), RAWNF(I,0)
110 CONTINUE
C CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
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, PEG,
   & 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)
130 CONTINUE
C BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C EXPECTED, 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
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
ENDIF

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 = I, (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) THEN
  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 = I, (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 = I, (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) .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 
      ENDIF
 190 CONTINUE
ENDIF
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
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)
DO 300 M = I, 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 + I), (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 + I), (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. I0) 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(8,*) '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',16X,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)
C*****************************************************************************
C THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
STRESS RATIO, R, IS NOT -1.0
PROGRAMMER: L. NEWLIN
DATE: CODE: 60CT87  COMMENTS: 13JUL89
VERSION: MATCHR V6, V6.2, V7, V7.1, V8, V8.1, V8.2,
V8.3, V8.4, V8.5

SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)
C INPUTS: J, NUM1, NUM2, STR, R, FTU, FTY
C OUTPUTS: RSTR
C IMPLICIT NONE
INTEGER MAXDAT, MAXSET
PARAMETER ( AXDAT = 50, MAXSET = 5)
COMMON IOUT
INTEGER I, IOUT, J, NUM1, NUM2
REAL
C 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

7 - 206
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,*)'1RSTR() =',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,*)'2RSTR() =',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,*)'3RSTR() =',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 VBI.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, K, 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),
WHAT2(MAXREG), SX2(MAXREG), SYX(MAXREG), SY2(MAXREG),
WHAT(MAXDAT, 0:MAXSET)

LIST OF VARIABLES

BB() 1-D ARRAY CONTAINING SX(Y)/SY2(L), THE SLOPE OF THE X ON Y
      REGRESSION, FOR EACH REGION
CC() 1-D ARRAY CONTAINING MEANY-DD(L)*MEANX, THE Y-INTERCEPT OF
      THE Y ON X REGRESSION, FOR EACH REGION
DD() 1-D ARRAY CONTAINING SX(Y)/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 LNLF(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
LNRF() 3-D ARRAY CONTAINING LN(RAWF()), 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)
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
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
RESID() 1-D ARRAY CONTAINING THE RESIDUALS OF THE REGRESSION FOR EACH
        POINT IN THE SPECIAL 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
          (X = Ln S)
SXHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
          REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
          (Y = Ln N)
SX() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
     EACH REGION (X = Ln S, Y = Ln N)
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
      (X = Ln S)
SY() 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
  SYX(L) = 0.0
  SWHAT2(L) = 0.0
  SUXHAT2(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
DO 70 K = 1, MAXDAT
  DIFFY(K,J) = 0.0
  DIFFX(K,J) = 0.0
  WHAT(K,J) = 0.0
  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 = I, NP(J,L)
      MEANY(J) = MEANY(J) + LNLF(K,J,L)
      MEANX(J) = MEANX(J) + LNSTR(K,J,L)
      IF (IOUT .EQ. 10) WRITE(8,*) 'LNLF =', LNLF(K,J,L),
        ' LNSTR =', LNSTR(K,J,L)
    CONTINUE
    MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
    MEANY(J) = MEANY(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) = LNLF(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),
        ' SXY(L) =', SXY(L),
      ENDIF
    CONTINUE
    NPPR(L) = NPPR(L) + NP(J,L) - 1
    IF (IOUT .EQ. 10) WRITE(8,*)
      'NPPR(L) =', NPPR(L)
  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
CALCULATE THE REGRESSION PARAMETERS
C
SY2(L) = SY2(L) / FLOAT(NPPR(L))
SX2(L) = SX2(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,*) 'SK(L) =', SY2(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, WHA[T2, 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 = I, NP(J,L)
    WHAT(K,J) = DIFFY(K,J) - DD(L) * DIFFX(K,J)
    SWHAT2(L) = SWHAT2(L) + WHAT(K,J) ** 2
    SUHAT2(L) = SUHAT2(L) + (DIFFY(K,J) - BB(L) * DIFFX(K,J)) ** 2
    IF (IOUT .EQ. 10) THEN
      WRITE(8,*) 'K =', K, ' WHAT(K,J) =', WHAT(K,J)
      WRITE(8,*) ' 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), ' SWHAT2(L) =', SWHAT2(L), ' SUHAT2(L) =', SUHAT2(L)

DO 600 K = 1, NP(0,L)
  RESID(K) = WHAT(K,0) * SQRT(FLOAT(NP(0,L)) / FLOAT(NP(0,L)-2))
  IF (IOUT .EQ. 10) WRITE(8,*) 'K =', K, ' RESID =', RESID(K)
  WRITE(4,*) K, RESID(K)
600 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: CODE: 8OCT87 COMMENTS: 13JUL89
C SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
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)

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) THEN
  WRITE(8,*)'CZERO = ', CZERO, ' CZERO2 = ', CZERO2
END IF

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

THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME COMMON CALCULATIONS

\[
\text{ARG2} = (\text{SXY}(L) \times 2 - \text{SY2}(L) \times \text{ARG1})
\]

IF (\text{ARG2} \gt 0.0) THEN

\text{ARG2 IS NEGATIVE -- IMPLIES M IS COMPLEX}

WRITE(8,*) 'ERROR: CO TOO LOW'

CALL TRMNAT

ELSE

\text{ARG2} = \text{ARG2 \times 0.5}

ENDIF

IF (\text{SX2}(L) \gt \text{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}) \div \text{ARG1}

ELSE

\text{SX2}(L) \lt \text{CZERO2} -- THIS TIME THE M CONSTRAINT IS A RANGE

\text{MCPNT}(L) = 2

\text{MC}(1,L) = (-\text{SXY}(L) - \text{ARG2}) \div \text{ARG1}

\text{MC}(2,L) = (-\text{SXY}(L) + \text{ARG2}) \div \text{ARG1}

ENDIF

ENDIF

100 CONTINUE

IF (\text{IOUT} \geq 10) THEN

DO 200 \text{L} = 1, \text{NUMREG}

WRITE(8,*) 'L = ', \text{L}, ' MCPNT = ', \text{MCPNT}(L)

WRITE(8,*) 'ARG1 = ', \text{ARG1}, ' ARG2 = ', \text{ARG2}

WRITE(8,*) 'MC(1,L) = ', \text{MC}(1,L), ' MC(2,L) = ', \text{MC}(2,L)

200 CONTINUE

ENDIF

RETURN

END

C*****************************************************************************


C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, & JZERO, MCHAT C OUTPUTS: IZERO, JZERO, MCHAT C SUBPROGRAMS: TRMNAT C IMPLICIT NONE

7 - 212
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) /
  0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
  3.237347, 1.68987, 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, 11.6885, 12.4011, 13.1197,
  13.8439, 14.5733, 15.3079, 16.0471, 16.7908,
  17.53, 18.26, 18.99, 19.73, 20.46,
  21.33, 22.10, 22.87, 23.63, 24.351,
  25.21, 25.99, 26.76, 27.53, 28.30,
  29.19, 29.96, 30.73, 31.50, 32.3574,
  33.23, 33.96, 34.74, 35.52, 36.39,
  37.21, 37.94, 38.72, 39.50, 40.2817,
  41.30, 42.12, 42.95, 43.77, 44.60,
  45.43, 46.26, 47.09, 47.92, 48.7536,
  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,
  88.06, 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, 103.85, 104.73,
  105.61, 106.49, 107.37, 108.25, 109.14,
  110.02, 110.90, 111.79, 112.67, 113.56,
  114.44, 115.35, 116.22, 117.10, 117.98 /

DATA (CHI025(I), I = 1, 75) /
  0.502389, 7.37776, 9.34840, 11.1423, 12.8325,
  14.4494, 16.0128, 17.5346, 19.0228, 20.4891,
  21.9200, 23.3367, 24.7356, 26.1190, 27.4884,
  28.8454, 30.1910, 31.5264, 32.8523, 34.1696,
  35.4789, 36.8207, 38.1757, 39.5364, 40.8665,
  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, 65.44,
  66.62, 67.82, 69.02, 70.22, 71.4202,
  72.61, 73.81, 75.00, 76.19, 77.38,
  78.57, 79.75, 80.93, 82.12, 83.2976,
  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, 112.39,
  113.54, 114.69, 115.84, 116.99, 118.136,
  119.28, 120.43, 121.57, 122.72, 123.86,
  124.00, 125.14, 126.28, 127.42, 128.56,
  130.70, 131.84, 132.98, 134.11, 135.241,
  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, 152.22,
  153.34, 154.47, 155.60, 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, 2.303, 3.182, 2.776, 2.571, 2.447,
& 2.365, 2.306, 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
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
IOUT CONTROLS LOOP FOR CHI025() AND CHI975()
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
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1)^2)-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() I-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SWHAT2() I-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)
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 (NPPR(L) .GT. CHITAB) THEN
WRITE(6,*) 'ERROR: EXCEEDED LIMIT ON DEGREES OF FREEDOM ',
7 - 214
& 'IN CHI-SQUARE TABLE, IN REGION ', L

ENDIF

75 CONTINUE

C ASSIGN 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

CALCULATE ESTIMATED VALUES OF M AND C

ARG = T * SWHAT / SX
MCHAT(1,L) = - DD(L)
MCHAT(2,L) = SUHAT

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) =', 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 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
INTEGER MAXREG, MAXSET
PARAMETER (MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG), & TOTAL
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), 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 N^+1 FOR EACH REGION WHERE N^+ 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

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. I0) 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)
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. NEWMAN
DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91
VERSION: MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
V8.4, V8.5
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
C 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
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
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. 0) .AND. (MCPNT(L) .EQ. 0)) THEN
  C THERE IS NO EXOGENOUS INFORMATION
  C ASSUME RANGE TO BE JO
  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),
    ' 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
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
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
ENDIF

7 - 218
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)
  &
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

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. i)) 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(I,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 ((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), 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 ((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 ((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
  ENDIF

ELSE
C  
RANGEM(1,L) = LOWER  
RANGEM(2,L) = UPPER  
ENDIF  

IF (IOUT .EQ. 10) THEN  
  WRITE(*,*) 'JZERO(I,L) = ', JZERO(I,L), '  
  WRITE(*,*) 'JZERO(2,L) = ', JZERO(2,L), '  
  WRITE(*,*) 'MZERO(I,L) = ', MZERO(I,L), '  
  WRITE(*,*) 'MZERO(2,L) = ', MZERO(2,L)  
ENDIF  

ELSE  
  WRITE(*,*) '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(*,*) '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(*,*) 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',  
    'ON m IN REGION ', L  
300 CONTINUE  

RETURN  
END  

***********************************************************************  
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  
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

C IOUT OUTPUT DUMP CONTROLLER
C LL CONTROLS DO LOOP FOR EACH REGION
C MAXREG MAXIMUM NUMBER OF REGIONS 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 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)
C IS UPPER BOUND
C NNODAT NUMBER OF NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
C NUMREG NUMBER OF REGIONS OF INTEREST
C RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND
C 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,
   IF (MPNT(LL) .EQ. 1) THEN
      WRITE(8,*),'RANGEM(1,LL) =', RANGEM(1,LL),
      WRITE(8,*),'MZERO(1,LL) =', MZERO(1,LL)
   ELSE
      WRITE(8,*),'RANGEM(2,LL) =', RANGEM(2,LL),
      WRITE(8,*),'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(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
   ENDIF
   IF (IOUT .EQ. 10) WRITE(8,*),'MCHAT =', MCHAT(1,LL)
   CALL TRNAT
100 CONTINUE
RETURN
END

C******************************************************************************

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: 15SEP89
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

C 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(1,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

C ADJUST RANGE TO INSURE CONCAVITY
DO 100 L = NUMREG, 2, -1
  IF (RANGEM(2,L-1) .LT. 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(*,*) '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(2,L-1)) THEN
      WRITE(*,*) 'ERROR: POSTERIOR INTERVAL IN REGION ',L,
      & IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
      & 'REGION ',L-1
      CALL TRMNAT
    ELSE
      RANGEM(2,L-1) = AMINI(RANGEM(2,L-1), TESTM)
    ENDIF
  ENDIF
  IF (IOUT .EQ. 10) THEN
    WRITE(*,*) 'RANGEM(1,L-1) = ', RANGEM(1,L-1),
    & RANGEM(2,L-1) = ', RANGEM(2,L-1)
    WRITE(*,*) 'RANGEM(1,L) = ', RANGEM(1,L),
    & RANGEM(2,L) = ', RANGEM(2,L)
    WRITE(*,*) '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 OUTPUT: MEDM
C
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
      RANGE IS A POINT
      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
100 CONTINUE

7 - 223
ENDIF

IF (IOUT .NE. 10) THEN
WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
& RANGEM(2,L) = ', RANGEM(2,L)
WRITE(8,*) 'LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
ENDIF

100 CONTINUE
RETURN
END

SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
& ZROREG, NBND, BIGKI, BZHAT)

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
INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXR G = 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),
& SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)

LIST OF VARIABLES

BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
EACH REGION
BIGK1 EQUAL TO BIGK(1)
BZHAT E(BETAo)
FACTR A SCALE FACTOR = PHI * KRATIO * Z
IOUT OUTPUT DUMP CONTROLLER
KHAT E(x)
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 UNAVAILABE – 2 FOR DECOMPOSED STRAIN DATA
NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE

7 - 224
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 NUMBER OF POINTS IN EACH REGION FOR THE SPECIFIC MATERIAL S/N DATA SET

NUMBER OF REGIONS OF INTEREST

1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()

2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

STRESS TENSILE TEST POINT, So

1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE

Zero Region — VALUES CHOSEN TO FACILITATE REGION DO LOOP

BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

1-D ARRAY CONTAINING TRANSFORMED S-N DATA, Z = F(STR, NF, NBND, MM)

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. 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)
200 CONTINUE

7 - 225
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(BETAO)' = ',F8.4,5X,'E(k)' = ',F8.4,5X,
   'REGION':,I4,5X,'m',15X,'K',9X,'LIFE BOUND',7X,
   'STRESS BOUND',/)

910 FORMAT(5X,I1,5X,F9.5,5X,I2.5,5X,E9.3,9X,E13.5)

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,/)
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

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. 10) THEN
      WRITE(8,*), 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ', MCHAT(1,L),
      & SUHAT2 = ', SUHAT2(L), ' MCHAT2 = ', MCHAT(2,L),
      & NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
      & DELTA = ', DELTA(L), ' ARG = ', ARG
      WRITE(8,*), 'MO = ', MO(L), ' MU = ', MU(L),
      & WHAT2 = ', WHAT2(L), ' SIGMA2 = ', SIGMA2(L),
      & SIG = ', SIG(L)
   ENDIF
100 CONTINUE

RETURN
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
C

SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, 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 LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
& RANGEM (2, MAXREG), UPPER

C LIST OF VARIABLES
C IOUT OUTPUT DUMP CONTROLLER
C L CONTROLS DO LOOP FOR EACH REGION
C LOWER LOWER BOUND OF INTERSECTION
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
C - MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C BOUND
C MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C FOR EACH REGION - MCHAT(1,L) = - DD(L), THE ESTIMATE
C FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C MC() FOR EACH REGION
C MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C MZERO() FOR EACH REGION
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)
C IS THE UPPER BOUND
C NUMREG NUMBER OF REGIONS OF INTEREST
C RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C FOR EACH REGION - RANGEM(1,L) IS THE LOWER BOUND AND
C 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
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)
100 CONTINUE

7 - 228
ENDIF

IF (MPNT(L) .EQ. 1) THEN

THERE IS A POINT PRIOR ON M — THIS OVERIDES 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),
&
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

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,*), '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 M0 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 M0 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,*), 'MC(1,L) = ', MC(1,L),
&
WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
&
WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,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(8,*), 'ERROR: NO INTERSECTION BETWEEN M0 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,*), 'MC(1,L) = ', MC(1,L),
&
WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
&
WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
&
ENDIF
& 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. I0) WRITE(8,*) 'RANGEM(I,L) = ', RANGEM(I,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*******************************************************************************************************

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 FORMAT/COMMENTS: 12AUG91
SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
& MZERO, MPNT, MO, SIGMA2)
C
C INPUTS: RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C C OUTPUTS: RANGEM, MCHAT, MU, SIG, NUMREG
C
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)
C
C LIST OF VARIABLES
C IOUT OUTPUT DUMP CONTROLLER
C L CONTROLS DO LOOP FOR EACH REGION

7 - 230
**CC**

**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) = SUSHAT, 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 the 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

---

**C**

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
   ENDIF
   WRITE(8,*), '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)
      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
100 CONTINUE
RETURN
END
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
MATGRM VB1, VB1.1

C 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)
C
C INPUTS: MM, BIGK, SZERO, NUMREG, ZROREG, NBND
C SUBPROGRAMS: FINDSB, KOMO
C
C IMPLICIT NONE
C
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

BIGK() 1-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() 1-D ARRAY CONTAINING 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 THE STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
CONTAINED IN NBND()
SZERO STRESS TENSILE TEST POINT, SO
TRBIGK() 1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
TRBIGK(1) = BIGK(1)
ZROREG ZERO REGION - VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE = 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

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

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:',14,
  & ///,2X,'REGION','m',7X,'K',9X,'LIFE BOUND',7X,
  & 'STRESS BOUND',/)
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, VBI.1
C
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)
  INPUTS: NPTS, NUMREG, ZROREG, RAND, NBND, STR, MEDM, MEDK, RESID
  OUTPUTS: BIGK, BZERO, MM, SBND
  SUBPROGRAMS: PICRES, MREGR, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB
C
C IMPLICIT NONE
C INTEGER MAXDAT, MAXREG
C PARAMETER (MAXDAT = 50, MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, ZROREG
C 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
C
C List of Variables
C
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 THE 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, 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)

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

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 %)

SZ2 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)

C OBTAIN THE VALUES OF THE RANDOMLY SELECTED RESIDUALS FOR EACH DATA POINT
CALL PICRES (RAND, MEDM, MEDK, RESID, NPTS, STR, RESNF)

C BOOTSTRAPPING M IS DESIRED
CALL MREGR (NUMREG, NPTS, STR, RESNF, MM)

C TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)
CALL TRNSFM (NPTS, STR, RESNF, 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
AND VARIANCE
CALL KBETA (MEANZ, SZ2, K, BZERO)

C CALCULATE THE VALUE OF K FOR EACH REGION WHERE A = K ** 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

7 - 234
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

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,
& ' RANT=', RANT, ' RANSUB=', RANSUB, ' RAND=', RAND,
& ' FRAC=', FRAC
RETURN
END

C NOTES: IOUT=2 DUMPS TO SCREEN

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
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 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

C LIST OF VARIABLES
C FRAC UNIFORM(0,1) RANDOM VARIATE
C IOUT OUTPUT DUMP CONTROLLER
C MU MEAN OF NORMAL DISTRIBUTION
C RAND RANDOM NUMBER SEED
C SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION
C X NORMAL RANDOM VARIATE
C U1 UNIFORM RANDOM NUMBER U(0,1)
C U2 UNIFORM RANDOM NUMBER U(0,1)
C Z1 NORMAL RANDOM NUMBER ON N(0,1)
C 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

7 - 236
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15)), &
WRITE(8,*), '21 = ', Z1, ' 22 = ', Z2, ' X = ', X
RETURN
END

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 MAXRN VBI, VBI.I

C SUBROUTINE PICRES (RAND, MEDM, MEDK, RESID, NPTS, STR, RESNF)
C INPUTS: RAND, MEDM, MEDK, RESID, NPTS, STR
C OUTPUTS: RESNF
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

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
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)
UNIFORM(0,1) RANDOM VARIATE

DO 50 K = 1, MAXDAT
RESNF(K,L) = 0.0
75 CONTINUE
50 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. Newlin
**Date:** 13NOV90
**Version:** MATCHR VB1.2, VB1.3

**SUBROUTINE MREGR (NUMREG, NPTS, STR, NF, MM)**

**INPUTS:** NUMREG, NPTS, STR, NF
**OUTPUTS:** MM

**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,
& SXY(MAXREG), SX2(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 MEANY FOR EACH POINT IN REGION L
**IOUT** OUTPUT DUMP CONTROLLER
**K** CONTROLS DO LOOP FOR EACH POINT IN A REGION
**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
(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(8,*) 'MEANY =', MEANY, ' MEANX =', MEANX

C NOW CALCULATE SAMPLE VARIANCES, SX2 AND SXY,
C OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
C 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)
   ENDIF
300 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

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) =', SXY(L), ' MM(L) =', MM(L)
100 CONTINUE

RETURN
END

C*******************************************************************************
SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM THE S/N DATA INTO THE VARIABLE \( Z = \ln(X) \)

PROGRAMMER: L. NEVIN

DATE: CODE: 1JUN88  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 OUT CONTROLS DO LOOP FOR EACH DATA POINT
I OUT OUTPUT DUMP CONTROLLER
K CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
L CONTROLS DO LOOP FOR 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 RAANF() (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)
  IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, '  MM =', MM(L), ' MML =', MML
  & NP = NPTS(L)
DO 200 K = 1, NPTS(L)
  ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
  IF (IOUT .EQ. 10) WRITE(8,*)'K =', K, NP, ' NF =', NF(K,L), ' STR =', STR(K,L), ' ZZ =', ZZ(NP)
200 NP = NP + 1
DO 300 LL = 2, L
300 CONTINUE
SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)
C INPUTS: NP, ZZ
C OUTPUTS: MEANZ, SZ2
C IMPLICIT NONE
C INTEGER MAXDAT
C PARAMETER (MAXDAT = 50)
C COMMON IOUT
C INTEGER I, IOUT, NP
C REAL MEANZ, SZ2, ZZ(MAXDAT)

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
SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)

C INPUTS: NP, ZZ
C OUTPUTS: MEANZ, SZ2
C IMPLICIT NONE
C INTEGER MAXDAT
C PARAMETER (MAXDAT = 50)
C COMMON IOUT
C INTEGER I, IOUT, NP
C REAL MEANZ, SZ2, ZZ(MAXDAT)

C*************************************************************************
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 SZ2 SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C ZZ() 1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C = F(STR,NF,NBND,MM)

C*************************************************************************
C INITIALIZE VARIABLES
C MEANZ = 0.0
C SZ2 = 0.0
C CALCULATE THE MEAN OF ZZ(), MEANZ
C DO 100 I = 1, NP
C MEANZ = MEANZ + ZZ(I)
C IF (IOUT .EQ. 10) WRITE(8,*),'NP =', NP, ' I =', I,
C & ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
C 100 CONTINUE
C MEANZ = MEANZ / FLOAT(NP)
C IF (IOUT .EQ. 10) WRITE(8,*), 'MEANZ =', MEANZ
C*************************************************************************
C*************************************************************************
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, NBNB, 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
LIST OF VARIABLES
C
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, NBNB, MM)
PI  SELF EXPLANATORY CONSTANT
SZ  SZ2 ** 0.5
SZ2  SAMPLE VARIANCE OF THE TRANSFORMED DATA,
Z = F(STR, NF, NBNB, 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, ', SZ2 =', SZ

7 - 242
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

I-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC MATERIAL DATA BASE
E_R'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
I-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
I-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)

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)))
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, 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(L) * NBND(0) ** (-1.0 / MM(L))
ENDIF

C CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES
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: JUN91
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, FACTR
C OUTPUTS: TRBIGK, MM, TRSBND

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, SO
TRBIGK() 1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP SBND(0) < SO 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

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
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. i0) THEN
WRITE(8,*) 'SZERO = ', SZERO, ' BIGKo = ', TRBIGK(0)
WRITE(8,*) 'FACTOR = ', FACTR, ' BIGK1 = ', TRBIGK(1)
WRITE(8,* ) 'MM1 = ', MM(1), ' MMO = ', MM(0)
ENDIF
RETURN
END

C SUBROUTINE WORSTN FINDS THE WORST OF N FOR BOTH THE WEIBULL AND
C LOGNORMAL DISTRIBUTIONS
C PROGRAMMER: L. NEWLIN
C DATE: 14NOV90
C VERSION: MATCHR VB1.2, VB1.3
C MATGRM VB1, VB1.1
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C 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 * T
  T3 = T * T2
  X = T - ((C0 + C1 * T + C2 * T2)
    / (1.0 + D1 * T + D2 * T2 + D3 * T3))
ENDIF

IF (IOUT .EQ. 10) WRITE(8,*) 'X = ', X

RETURN
END

C FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C PROGRAMMER: L. NEWLIN
C DATE: 10FEB89
C VERSION: MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

7 - 247
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
C
C GETLIF VALUE TO BE ASSIGNED TO GTLIFE - CYCLES TO FAILURE FOR
C THE REQUIRED STRESS LEVEL
C IOUT OUTPUT DUMP CONTROLLER
C KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C L CONTROLS DO LOOP FOR EACH REGION
C LNA() 1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C LNZ NORMAL(0,0.01) GENERATED RANDOM VARIATE
C LPHIM() 1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C PHI IS A WEIBULL(BETA, ETA) GENERATED RANDOM VARIATE
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C NUMREG NUMBER OF REGIONS OF INTEREST
C S VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO
C FAILURE) IS REQUIRED
C SBND() 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
C CONTAINED IN NBND()
C SZERO STRESS TENSILE POINT, So
C TEMP TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
C FLOWS
C ZROREG 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 .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
100 CONTINUE
ENDIF
150 CONTINUE
GTLIFE = GETLIF
RETURN
END

C******************************************************************************

7 - 248
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)

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
   IF (INC .GT. 1) THEN
      INC = INC / 2
   ENDIF
   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.
      ENDFI
   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), LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MM(0:MAXREG), PERIOD, RAINF3, S(MAXM), SBND(0:MAXREG), SEFF(MAXM), SEFM(2, MAXM), SEFMAX, SP(MAXM), SRANGE(MAXM), SUMDAM, SZERO, TESTI(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
SEFM(2,N) EFFECTIVE STRAINS AFTER RESEQUENCING/FILTERING/RAINFLOW
SEFM(1,I) = sigma max,eff,i
SEFM(2,I) = sigma min,eff,i
SRANGE(N) SRANGE(I) = EQUIVALENT STRAIN RANGE FOR CYCLE I
REAL FUNCTION THAT CALCULATES FATIGUE LIFE FOR A GIVEN STRAIN

LIFE(I) = CALCULATED LIFE FOR STRAIN LEVEL SRANGE(I);

INVLIF(I) = 1/LIFE(I); DAMAGE FRACTION

SUM OF ALL THE DAMAGE FRACTIONS

CHKFNT DUMMY VARIABLE USED TO PRINT OUT RAINF3 RESULT

IOUT OUTPUT DUMP CONTROLLER

KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS

LNA() 1-D ARRAY CONTAINING VALUES OF LN(A) = M Ln K FOR EACH REGION

LNX 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 VALUES OF M FOR EACH REGION

NUMREG NUMBER OF REGIONS OF INTEREST

SBND() 1-D ARRAY CONTAINING THE STRAIN VALUES (%R = -1.0)

SZERO STRAIN TENSILE POINT, So (%)

ZROREG 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(8,'(m,m)', 'period',period)
write(8,'(m,m)', 'wexp',wexp)
write(8,'(m,m)', 'numreg',numreg,'zroreg',zroreg
write(8,'(m,m)', 'szero',szero,'kratio',kratio,'lnz',lnz
write(8,'(m,m)', 'ina(i), mm(i), lphim(i), sbnd(i)
write(8,'(m,m)', 'szero','Ratio',kratio,'inz',inz
endif

initialize arrays

DO 50 I = i, MAXM
SP(I) = 0.0
S(I) = 0.0
E(I) = 0.0
SEFFM(I,I) = 0.0
SEFFM(2,I) = 0.0
SRANGE(I) = 0.0
LIFE(I) = 0.0
INDEX(I) = 0.0
TEST1(I) = 0.0
TEST2(I) = 0.0
50 CONTINUE

*********** BEGIN RESQUENCE ***********

RESQUENCE effective strains (needed for rainflow analysis);

largest effective strain is placed at beginning and end of SP(M+1)

find SEFMAX, the largest sigma,eff, 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

assign all points from JMAX out, to the beginning of SP()

DO 210 I = 1, M-JMAX+1
J = JMAX-I + I
SP(I) = SEFF(J)
210 CONTINUE

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
SP(M+1) = SEFF(JMAX)
if (iout.eq.20) then
write(8,*),'sefmax,' jmax:
write(8,*),'sp(m+1): ,(sp(i),i=1,m+1)
endif

C*************** E N D  R E S O U R C E  ********************************
C*************** B E G I N  F I L T E R  ********************************
C FILTER the resequenced effective strains, leaving only peaks and valleys
C (excursions larger than TRUNC are deleted during rainflow counting) in
C S(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
if (iout .eq. 20) then
do 305 i = 2, m
write(8,*),'test1 = ', test1(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
K = K + 1
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*************** E N D  F I L T E R  ********************************
C*************** B E G I N  R A I N F L O W  ****************************************
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+1
C check J to avoid reading beyond end of filtered strain data
IF ( J .GT. NEWTOT ) GOTO 499
C read strain point into a holding array to be checked for cycles
E(K) = S(J)
410 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
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
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****************************** E N D R A I N F L O W **************************
C calculate equivalent strain range
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
C calculate lives and damage fractions: LIFE(N) and INVLIF(N)
DO 600 I=I,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(8,*),life(i),''
14 continue
endif
C Miner's Rule -- sum the damage fractions
SUMDAM = 0.0
DO 700 I=1,N
SUMDAM = SUMDAM + INVLIF(I)
700 CONTINUE
C calculate fatigue life (time to failure)
RAINF3 = PERIOD / SUMDAM
if (iout.eq.15) then
chkft=period/sumdam
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>
<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>ADDRGN¹</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>BLDHFLF</td>
<td>5.3.2.4</td>
<td>Performs the calculations of the driver transformation and the fatigue life.</td>
</tr>
<tr>
<td>CONCAV²</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>CONVRT³</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>EXPCTD⁴</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>FINDM⁵</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>FINDMN⁵</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 stress 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 ( \beta_o ) and ( k ).</td>
</tr>
<tr>
<td>FNDRNG⁶</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>
</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>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 $I_o$ 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 $J_o$ 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 $m_<em>$ and standard deviation $\sigma_</em>$, for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NORMGN</td>
<td>4.4.3</td>
<td>Generates Normal($\mu$, $\sigma^2$) random variates.</td>
</tr>
<tr>
<td>NORGNG</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 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>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>SORTM$^{13}$</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>TRNSFM$^{14}$</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.¹</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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>BZERO</td>
<td>RE</td>
<td>Estimate of Weibull distribution shape parameter $\beta_0^\prime$, 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>FIFTY</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>LAMDB</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>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 $\ln(A) = \ln(BIGK) * MM$ 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($0$, 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 $\ln(PHI) * MM$ 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>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<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 M_COUNT.</td>
</tr>
<tr>
<td>MINPHI</td>
<td>RE</td>
<td>Value of min(PHI), the minimum of 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>$m_d f_d \omega^2$, 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>$N_{i, i+1}$ 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 BLDHLF.</td>
</tr>
<tr>
<td>NF</td>
<td>RE</td>
<td>$N_f$, the fatigue life in cycles.</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>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>
</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>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>RPMMS</td>
<td>RE</td>
<td>Mean, $\mu$, of Normally distributed $\omega$, the steady state rotor speed (rpm).</td>
</tr>
<tr>
<td>RPMMS</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 BLDHC (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(^7) 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_{MAX}) (psi) in Equation 4-4, the maximum or peak stress.</td>
</tr>
<tr>
<td>SMEAN</td>
<td>RE</td>
<td>(\sigma_{MEAN}) (psi) in Equation 4-2, the mean stress.</td>
</tr>
<tr>
<td>SMIN</td>
<td>RE</td>
<td>(\sigma_{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_{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>

7 - 267
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.
### 7.3.4 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>INTVL</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>ADDRGN</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>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>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>DRVRI</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>ADDRIG</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>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>
<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>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>7-280</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-329</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-327</td>
</tr>
<tr>
<td>PRYAV</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>BLDHFL</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>DRVIRIN</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>7-319</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-322</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-323</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-326</td>
</tr>
<tr>
<td>PAREST</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>RANDOM</td>
<td>7-330</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-331</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-332</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-333</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-334</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-335</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-336</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-337</td>
</tr>
<tr>
<td>SELECT</td>
<td>7-338</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-339</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-340</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-341</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-342</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-343</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-344</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-345</td>
</tr>
<tr>
<td>BLDHNF</td>
<td>7-346</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-347</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-348</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-349</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-350</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-351</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-352</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-353</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-354</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-355</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-356</td>
</tr>
</tbody>
</table>

7 - 271
**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** Copyright (C) 1990, California Institute of Technology.
**C** U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

**C** PROGRAM BLDHCF

**C** SUBPROGRAMS: DRVRIN, INFAGG, PAREST, NORMGN, SELECT, WEIBGN,
**C** TRMNAT, BLDHLP, INSORT, SORTM, EXPTCD

**C** FILES: I:BLDHCD-OLD; 3:BLDHCO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
**C** 7:DUMP-NEW; 8: IOUTPR-NEW; 9:LOWLIF-NEW;
**C** NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

**C** IMPLICIT NONE

**C** INTEGER MAXBLF, MAXDAT, MAXLIF, MAXMM, MAXREG

**C** PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLIF = 10000,
**C** MAXMM = 20001, MAXREG = 3)

**C** COMMON IOUT

**C** INTEGER BLFPOS(MAXBLF), I, IOUT, J, K, L, M, MCOUNT, MID,
**C** & MPROC, NS, NLFIFE, NHYPF, NLIFE, NLIFET, NMED,
**C** & NPTS(MAXREG), NSYM, NUMREG, VARY, ZROREG

**C** DOUBLE PRECISION RAND

**C** REAL A0, AI, ALLM(MAXMM, MAXREG), B0, B1, BIGK(0:MAXREG),
**C** & BIGK1, BLDHLP, BLPFPER(MAXBLF), BZERO, C, C0, C10, C11,
**C** & C20, C21, CH, CS, FACTR, FIFTY, FTU, FTU, HIM, KRATIO,
**C** & LAMB, LAMDA, LAMDB, LAMDA, LAMDB, LIFE(MAXLIF),
**C** & LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MD, MEDI(MAXREG),
**C** & MINPHI, MM(0:MAXREG), MJ(MAXREG), MW, MWA, MWB,
**C** & NBDN(0:MAXREG), NEWLIF, NF(MAXMM, MAXREG), PHI, PSIG,
**C** & PVAR, RANGEM(2, MAXREG), RAVG, RAVGM, RAVGS, RD, RDM,
**C** & RDS, RPM, RPM, RROOT, RROOT, RROOTS,
**C** & SBND(0:MAXREG), SIG(MAXREG), STR(MAXDAT, MAXREG), SZERO,
**C** & TRBIGK(0:MAXREG), TRSBND(0:MAXREG), Z

**C** **SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

**C** OPEN (1, FILE = 'BLDHCD', STATUS = 'OLD')
**C** OPEN (3, FILE = 'BLDHCO', STATUS = 'NEW')
**C** OPEN (7, FILE = 'DUMP', STATUS = 'NEW')
**C** OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
**C** OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')

**C** READ(1,*) RAND
**C** READ(1,*) IOUT
**C** READ(1,*) NLFIFE
**C** READ(1,*) NHYPF
**C** READ(1,*) NSYM
**C** READ(1,*) VARY
**C** READ(8,*) ' RANDOM NUMBER SEED = ', RAND
**C** READ(8,*) ' IOUT (MATCHR = 10, BLDHCF = 15) = ', IOUT
**C** READ(8,*) ' INNER LOOP SIZE = ', NLFIFE
**C** READ(8,*) ' OUTER LOOP SIZE = ', NHYPF
**C** READ(8,*) ' SYMMETRY NUMBER = ', NSYM

**C** WRITE(8,*) ' TYPE OF S/N VARIATION DESIRED '
**C** WRITE(8,*) ' (0-NONE; 1-INTRINSIC; 2-UNIFORM; 3-NORMAL) = ', VARY
**C** WRITE(8,*) ' NORMAL MEDIAN CURVE (0 - NO, 1 - YES) = ', NMED

**C** READ(1,*) MPROC
**C** WRITE(8,*) ' MATERIALS PROCESS VARIATION DESIRED'
**C** 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)
CALL DRVRIN TO READ DATA FROM BLDHCD AND ECHO DATA TO BLDHCO
CALL DRVRIN (RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS, CM, CS, 
  & RDM, RDS, LAMBA, LAMBB, LAMDA, LAMDB, MWA, MBW, 
  & IMIN, MD, NB, NS, A0, A1, B0, B1, C0, C10, C11, 
  & C20, C21)
C ** CALL INFAGG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT 
C ** OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, 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 
C ** MATERIALS CHARACTERIZATION MODEL CALCULATIONS
  CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, 
  & RAND, NBND, STR, BIGK, BZERO, MM, SBND)
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, NBLIFE
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, RPMX, RROOT, RROOTM,
  & RROOTS, RAVG, RAVGM, RAVGS, C, CM, CS, RD,
  & RDM, RDW, LAMB, LAMBX, LAMMB, Lambda, LAMDB, LambdaM,
  & LAMDB, MW, MWA, MB)

MINPHI = 1.0E+36
DO 230 M = l, 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

IF (ZROREG .EQ. 0) CALL KOMOM (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)
    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
250 CONTINUE

C ** CALL BLDHLF TO OBTAIN BLADE HCF LIFE

NEWLIF = BLDHLF (RPM, RROOT, RAVG, C, RD, LAMB, Lambda, 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

150 CONTINUE

IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES TO FILE LOWLIF

DO 300 J = l, (NLIFET / 100)
  WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
300 CONTINUE

C ** INITIALIZE VARIABLE BLFPOS()
DO 325 J = I, MAXBLF
   BLFPOS(J) = 0
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)
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)
   CONTINUE
   CALL EXPCTD (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******************************************************************************
C SAMPLE 'BLDHC'D' INPUT FILE
C******************************************************************************
C 675 RANDOM NUMBER SEED
C 0.............................. INNER LOOP SIZE
C 1.............................. OUTER LOOP SIZE
C 20000.......................... SYMMETRY NUMBER
C 54.............................. UNIFORM S/N VARIATION
C 0.............................. NORMAL MEDIAN NOT REQUIRED
C 0.............................. MAT. PROC. VAR. NOT REQUIRED
C 3.0001.......................... NUMBER OF BLIVES REQUESTED
C 0.001.......................... B.01 LIFE
C 0.01.......................... B1 LIFE
C 26161.......................... ROTOR SPEED VARIATION PARAMETERS:
C 600.............................. BLADE ROOT RADIUS MEAN & STD DEV
C 4.700 0.0035.................... BLADE AVERAGE RADIUS MEAN & STD DEV
C 5.117 0.0035.................... DISTANCE FROM NEUTRAL AXIS MEAN & STD DEV
C 4.445 0.010..................... DAMPER RADIUS MEAN & STD DEV
C 0.00 0.00.......................UNCERT. IN PERFORMANCE BALANCE
C 0.50 1.50....................... UNCERT. IN DAMPER COEFFICIENT OF FRICTION
C 0.40 0.60....................... WALKER EXponent m
C 0.0004769........................ MINIMUM MOMENT OF INERTIA
C 0.0010733........................ DAMPER MASS
C 54.---------------------------- NUMBER OF ROTOR BLADES

7 - 275
78. ................................ NUMBER OF STATOR VANES

COEFFICIENTS OF RESPONSE SURFACE FUNCTIONS

FLOW RATE:
\[ F_{\text{mdot}}(w) = A + B \cdot w \]
\[ A = -24.41242623 \quad B = 0.3307822E-02 \]

ENTHALPY CHANGE:
\[ F_{\text{delta h}}(w) = A + B \cdot w \]
\[ A = -29.65037673 \quad B = 0.6433368E-02 \]

BLADE DAMPER EFFECTIVENESS:
\[ F_{\text{eff}}(m, r, w) = \begin{cases} 
A + B \cdot m^2 & \text{if } m^2 < A \\
C + D \cdot m^2 & \text{if } m^2 > A 
\end{cases} \]
\[ A = 26 \quad B = 1.0 \quad C = -0.03750 \quad D = 5.883003E-3 \quad E = 7.429614E-4 \]

'MATERIAL DESCRIPTION'

YIELD & ULTIMATE STRENGTHS, NDIV, NPTS

<table>
<thead>
<tr>
<th>Material Strength</th>
<th>NDIV</th>
<th>NPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>137000.</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>142000.</td>
<td>-1.0</td>
<td>1</td>
</tr>
</tbody>
</table>

# PTS IN DIV, STRESS RATIO, REGION

<table>
<thead>
<tr>
<th>Material Strength</th>
<th>Number of Regions: W/ Data W/O Data</th>
<th>Constraint on Coeff. of Variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>80000.</td>
<td>1.0E+36</td>
<td></td>
</tr>
<tr>
<td>50000.</td>
<td>1.0E+36</td>
<td></td>
</tr>
</tbody>
</table>

VALUE OF S0 SUPPLIED (PSI)

NORMAL DIST. PRIORS: DELTA, MO, SIGMA2

LIST OF VARIABLES

A0, A1 COEFFICIENTS OF THE FLOW RATE, m-dot, RESPONSE SURFACE
ALLM() 2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
B0, B1 COEFFICIENTS OF THE ENTHALPY CHANGE, delta-h, RESPONSE SURFACE
BIGK(1) 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
BIGK1 EQUAL TO BIGK(1) - DUMMY PARAMETER FOR CALLS TO SUBROUTINE EXPCTD
BLDFHLF REAL FUNCTION PERFORMING THE DRIVER TRANSFORMATION AND HCF LIFE
CLAMBA 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)
FACR SCALE FACTOR EQUAL TO PHI * KRAITO * Z
FIFTY EQUAL TO 50% POINT IN LIFE()
FYU MATERIAL YIELD STRENGTH (psi)
JYU MATERIAL YIELD STRENGTH (psi)
K CONTROLS OUTER DO LOOP
KOUT CONTROLS DUMP TO FILE IOUTPR
L CONTROLS DO LOOP FOR EACH BLIFE
LAMB SELECTED UNCERTAINTY IN PERFORMANCE BALANCE MODEL, LAMBA
LAMBA SELECTED UNCERTAINTY IN PERFORMANCE BALANCE MODEL, LAMBA

7 - 276
DISTRIBUTION LOWER BOUND, LAMBdAB, UNIFORM
DISTRIBUTION UPPER BOUND, LAMBD
SELECTED UNCERTAINTY IN DAMPER COEFFICIENT OF FRICTION, LAMBdAD
SELECTED UNCERTAINTY IN DAMPER COEFFICIENT OF FRICTION, LAMBdAD, UNIFORM
DISTRIBUTION LOWER BOUND, LAMDB
DISTRIBUTION UPPER BOUND, LIFE(1)
1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
- SORTED VALUES OF THE LEFT-HAND TAIL
LNA(1)
1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
LN2 NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
LPHM() 1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
M CONTROLS SYMMETRY DO LOOP
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
MAXMM MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCOUNT NUMBER OF M's TO BE USED TO CALCULATE THE TRUNCATED NORMAL
MEDIAN S/N CURVE
MDAMPER MASS (lbm)
MEDM() 1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
MINPHI EQUAL TO MIN(PHI) - THE MINIMUM OF NSYM DRAWS OF PHI
MM() 1-D ARRAY CONTAINING VALUES OF M 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
MW SELECTED WALKER M
MWA WALKER M UNIFORM DISTRIBUTION LOWER BOUND
MBW WALKER M UNIFORM DISTRIBUTION UPPER BOUND
NB NUMBER OF ROTOR BLADES
NBLIFE NUMBER OF BLIVES TO BE PROVIDED
NBND() 1-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 BLDHLF
NF() 2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL S/N DATA
SET BROKEN INTO LIFE REGIONS
NHYPRE SIZE OF OUTER LOOP
NLIFE SIZE OF INNER LOOP
NLIFET TOTAL NUMBER OF LIVES CALCULATED BY PFM
NAMED 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 SYMMETRY NUMBER
NUMREG NUMBER OF REGIONS OF INTEREST
PHI WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
PSIG EQUAL TO SQRT(PVAR) - MATERIALS PROCESS STANDARD DEVIATION
PVAR MATERIALS PROCESS VARIATION
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
RAVG SELECTED BLADE AVERAGE RADIUS (in)
RAVGM MEAN OF AVERAGE BLADE RADIUS (in)
RAVGMS STANDARD DEVIATION OF AVERAGE BLADE RADIUS (in)
RD SELECTED DAMPER RADIUS (in)
RDMS MEAN OF DAMPER RADIUS (in)
RDS STANDARD DEVIATION OF DAMPER RADIUS (in)
RPM SELECTED ROTOR SPEED (rpm)
RPMM MEAN OF ROTOR SPEED (rpm)
RPMMS STANDARD DEVIATION OF ROTOR SPEED (rpm)
ROOT SELECTED BLADE ROOT RADIUS (in)
ROOTM MEAN OF BLADE ROOT RADIUS (in)
ROOTS STANDARD DEVIATION OF BLADE ROOT RADIUS (in)
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

STRAIN() 2-D ARRAY CONTAINING STRESS POINTS (STRESS RATIO = -1.0) FOR

S-N SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS

SZERO STRESS TENSILE TEST POINT, SC

B1GK() 1-D ARRAY CONTAINING VALUES OF B1GK() 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

Z0REG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP

BEGINNING VALUE = 0 ZERO REGION EXISTS, 1 -- NO ZERO REGION EXISTS

C***********************************************

SUBROUTINE DRVWIN READS AND ECHOES THE INPUT DATA

PROGRAMMER: L. NEWLIN

DATE: 31OCT90 COMMENTS: 20APR92

VERSION: BLDHCW V1, V1.1

Copyright (C) 1990, California Institute of Technology.

U.S. Government Sponsorship under NASA Contract NAS7-918

is acknowledged.

SUBROUTINE DRVWIN (RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS,

CM, CS, EDM, RDS, LAMBA, LAMBD, LAMDA,

& LAMDB, MWA, MBW, IMIN, MD, NB, NS, A0, A1,

& B0, B1, C0, C10, C11, C20, C21)

OUTPUT: RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS, CM, CS, EDM, RDS,

LAMBA, LAMBB, LAMDA, LAMDB, MWA, MBW, IMIN, MD, NB, NS,

A0, A1, B0, B1, C0, C10, C11, C20, C21

IMPLICIT NONE

COMMON IOUT

INTEGER IOUT, NB, NS

REAL A0, A1, B0, B1, C0, C10, C11, C20, C21, CM, CS, IMIN,

& LAMBA, LAMBB, LAMDA, LAMDB, MWA, MBW, RAVGM, RAVGS,

& EDM, RDS, RPMM, RPMS, 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)

IOUT Output dump controller

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)

MWA Walker m Uniform distribution lower bound

MBW Walker m Uniform distribution upper bound

NB Number of rotor blades

NS Number of stator vanes

7 - 278
C RAVGM Mean of average blade radius (in)
C RAVGS Standard deviation of average blade radius (in)
C RDM Mean of damper radius (in)
C RDS Standard deviation of damper radius (in)
C RPMM Mean of rotor speed (rpm)
C RPMS Standard deviation of rotor speed (rpm)
C RROOTM Mean of blade root radius (in)
C RROOTS Standard deviation of blade root radius (in)

READ(1,*) 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

WRITE(3,900) RPMM, RPMS, RROOTM, RROOTS, RAVGM, RAVGS,
& CM, CS, RDM, RDS,
& LAMBA, LAMBB, LAMDA, LAMDB, MWA, MWB
WRITE(3,903) IMIN, MD, NB, NS
WRITE(3,904) 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,,',F8.1, ',),',
& //,2X,'BLADE ROOT RADIUS (in)',25X,
& 'N(',F6.3,',',E10.3,'),',
& //,2X,'BLADE AVERAGE RADIUS (in)',22X,
& 'N(',F6.3,',',E10.3,'),',
& //,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,12
& //,2X,'NUMBER OF STATOR VANES',29X,12)

904 FORMAT(///,13X,'COEFFICIENTS OF RESPONSE SURFACE FUNCTIONS',
& //,2X,'FLOW RATE',
& //,5X,'Fndot(w) = ',F12.8,' + ',E14.7,' * w',
& //,2X,'ENTHALPY CHANGE',
& //,5X,'Fdelh(w) = ',F12.8,' + ',E14.7,' * w',
& //,2X,'BLADE DAMPER EFFECTIVENESS:',
& //,5X,'IF mrw**2 < ',F4.1,
& //,10X,'Feff(m, r, w) = ',E14.7,' + ',E14.7,' * mrw**2',
& //,5X,'IF mrw**2 > ',F4.1,
& //,10X,'Feff(m, r, w) = ',E14.7,' + ',E14.7,' * mrw**2')

RETURN
END
SUBROUTINE SELECT (RAND, RPM, RPMM, RPM, RROOT, RROOTM, & RROOTS, RAVG, RAVGM, RAVGS, C, CH, CS, RD, & RDM, RDS, LAMB, LAMBA, LAMBB, LAMDB, LAMDA, & LAMDB, MW, MWA, MWB)

INPUT: RAND, RPMM, RPM, RROOT, RROOTM, RROOTS, RAVGM, RAVGS, CH, CS, RD, RDM, RDS, LAMB, LAMBA, LAMBB, LAMDB, LAMDA, MW, MWA, MWB

OUTPUT: RPM, RROOT, RAVG, C, RD, LAMB, LAMDA, MW

IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL C, CM, CS, DUM, LAMB, LAMBA, LAMBB, LAMDB, MW, MWA, MWB, RAVG, RAVGM, RAVGS, RD, RDM, RDS, RPM, RPMM, RPMS, 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, LAMbdaB
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
LAMDB Uncertainty in damper coefficient of friction, LAMbdaD, Uniform distribution lower bound
LAMDB 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
RN Random number seed
RAVG Selected blade average radius (in)
RAVGM 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 Selected 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, RPMM, RPM, RPM)
CALL NORMGN (RAND, RROOTM, RROOTS, RROOT)
CALL NORMGN (RAND, RAVGM, RAVGS, RAVG)
CALL NORMGN (RAND, CM, CS, C)
CALL NORMGN (RAND, RDM, RDS, RD)

CALL PRYRV (RAND, LAMB, LAMBB, LAMDA, LAMDB, LAMDA, MW, MWA, MWB)
CALL PRYRV (RAND, MWA, MWB, MWA, MWB, MW, DUM)
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,*)
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 (MATCHR 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 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, ZZERO)
 C INPUT: 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, ZZERO
 C OUTPUT: BLDHLF
 C SUBPROGRAMS: GTLIFE

C IMPLICIT NONE

COMMON IOUT
INTEGER IOUT, MAXREG, NB, NS, NUMREG, ZZERO
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, ZZERO

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
BLDHLF Real function performing the driver transformation and HCF life calculation
C Selected distance from neutral axis (in)
C0, C10, C11, C20, C21 Coefficients 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 LPHIM() 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 SBND() 1-D array containing the stress values (psi, R = -1.0)
corresponding to the "life boundary" values for each region
C 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 SMEN 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
C 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

C CALCULATE FLOW CONDITIONS
MDOT = LAMB * (A0 + A1 * RPM)
DELTAH = LAMB * (B0 + B1 * RPM)

C CALCULATE BLADE ROOT MEAN STRESS
SBRM = (MDOT * DELTAH / RPM) * (C / (IMIN * FLOAT(NB)))
& * (1.0 - (RROOT / RAVG)) * 9336

C OBTAIN BLADE UNDAMPED VIBRATORY STRESS
SUD = (8.55300181 + 34.06551173 * (SBRM / 9336)) * 1000.0

C CALCULATE DAMPER NORMAL LOAD
MRW2 = (MD * RD * (RPM ** 2)) * 2.83805E-5

C 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

C CALCULATE ALTERNATING & MEAN STRESSES, MAX & MIN STRESSES, AND THE STRESS RATIO
SALT = SUD * (SDSUD)
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, ZZERO)

C TRANSFORM LIFE FROM CYCLES TO SECONDS
LIFE = (60.0 / RPM) * NF / FLOAT (NS)
BLDKLF = LIFE

IF (IOUT .EQ. 15)
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
C PROGRAMMER: L. NEWLIN
C DATE: 20JUL90
C VERSION: 2.1

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

SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)
C INPUTS: NEWLIF, LIFE, NLIFET
C OUTPUTS: LIFE
C
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(1)

7 - 283
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
C INSERTION OF NEWLIF

NUM = NLIFET / 2
C
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
C STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()
DO 200 I = (PLACE + i), NUM
   TEMP(I) = LIFE(I-i)
200 CONTINUE
C
C INSERT NEWLIF
LIFE(PLACE) = NEWLIF
C
C SHIFT VALUES OF LIFE() FOLLOWING NEWLIF
DO 300 I = (PLACE + i), NUM
   LIFE(I) = TEMP(I)
300 CONTINUE
C
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.
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)
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
\[ Y = \text{FRAC} \times (\text{THE2} - \text{THE1}) + \text{THE1} \]

\[
\text{IF (IOUT .EQ. 15) WRITE(8,*) 'RHOI =', RHOI, ' RHO2 =', RHO2, ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y}
\]

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 FORMAT/COMMENTS: 12AUG91
C VERSION: MATCHR V8.4, V8.5 MATGRM V4.4, 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 INFAGG (RANGEM, MU, SIG, NF, REFPN, SZERO, ZROREG, & NUMREG, NBBND, STR, FTUZ, FTYZ, VARY, MPROC, & KRATIO, PVAR)
C
INPUTS: READS DATA FROM SPECIFIED AND RELATD; VARY, MPROC
C
OUTPUTS: RANGEM, MU, SIG, NF, REFPN, SZERO, ZROREG, NUMREG, & NBBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C
SUBPROGRAMS: INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDRGE, & CONCAV, MEDIAN, EXPCTD, MUSIG, NORMN, ADDRGN, GTPVAR
C
FILES: 5:RELATD-OLD; 6:RELATO-NEW
C
IMPLICIT NONE
C
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
C
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT, & NF(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET), & NSETS, NUMREG, REFPN(MAXREG), VARY, ZROREG
REAL 
BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG), & FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG), & KRATIO, LAMN, LNP(MAXDAT, 0:MAXSET, MAXREG), & LNSTR(MAXDAT, 0:MAXREG, MAXREG), MC(2, MAXREG), & MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG), & MZERO(2, MAXREG), NBBND(0:MAXREG), NF(MAXDAT, MAXREG), & PVAR, RANGEM(2, MAXREG), RAWTR(MAXDAT, 0:MAXSET), & RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET), & SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT, MAXREG), & SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(MAXREG), & SXY(MAXREG), SY2(MAXREG), SZERO
C
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
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

7 - 285
2-D ARRAY CONTAINING $J_0$, THE 95% CONFIDENCE INTERVALS ON $M$ FOR EACH REGION

RATIO OF $K^*/K$, CONSTANT OVER REGIONS AND COMPONENTS

L

CONTROLS DO LOOP FOR EACH REGION

LAMBDA-N = RATIO OF Var(Ln N given S) / (m**2 C**2), CONSTANT OVER REGIONS AND COMPONENTS

3-D ARRAY CONTAINING Ln(RAWNF()), ALSO INDEXED FOR REGION

3-D ARRAY CONTAINING Ln(RATSTR()), ALSO INDEXED FOR REGION

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

MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON $M$ FOR EACH REGION CONSISTENT WITH GIVEN VALUE OF $C_0$ AND THE DATA -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND

2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF $M$ AND $C$ FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) = $-\bar{D}$, THE ESTIMATE FOR $M$ AND MCHAP(2,L) = $\sqrt{\bar{D}}$, THE ESTIMATE FOR $C$

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

NUMBER OF NO DATA REGIONS (REGIONS WITHOUT ANY S/N DATA)

2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION

1-D ARRAY CONTAINING VALUES OF $\left(\sum \left(\frac{\text{NF}() - 1}{\text{NF}()}\right) - 1\right)$ OVER ALL DATA SETS IN A REGION (NUMBER OF POINTS PER REGION)

1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS

NUMBER OF RELATED MATERIAL S/N DATA SETS

NUMBER OF REGIONS OF INTEREST

MATERIALS PROCESS VARIATION

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

2-D ARRAY CONTAINING STRESs DATA (PSI) CORRECTED FOR STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS

2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS

2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS

1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION

1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION

2-D ARRAY CONTAINING RANGES ON $M$ FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM $X$ ON $Y$

REGRESSION FOR EACH REGION ($X = \ln S$, $Y = \ln N$)

1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM $Y$ ON $X$

REGRESSION FOR EACH REGION ($X = \ln S$, $Y = \ln N$)

1-D ARRAY CONTAINING SAMPLE $X$ VARIANCE FOR EACH REGION ($X = \ln S$, $Y = \ln N$)

1-D ARRAY CONTAINING SAMPLE $Y$ VARIANCE FOR EACH REGION ($X = \ln S$, $Y = \ln N$)

STRESS TENSILE TEST POINT, SO
OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
OPEN(6, FILE = 'RELATD', STATUS = 'NEW')

C RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET
C 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, 
                  SWHAT2, SUHAT2, NPPR)
C CALCULATE M CONSTRAINT BASED ON CO
    CALL FINDMC (NUMREG, CZERO, SX2, SY2, 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 ADDREG (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(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)
END!
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
  CONTINUE
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 100 L = 1, NUMREG
  WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
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, 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

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(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(L)
    ELSEIF (MCPNT(L) .EQ. 2) THEN
      WRITE(7,970) L, MC(L), MC(2,L)
    ENDIF
  360 CONTINUE
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 370 L = 1, NUMREG
  WRITE(7,940) L, RANGEM(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
P NT REs=Ts
oF MATE ALs PR ESS V ATIO. T IF (MPROC .EQ. 1) THEN
  WRITE(7,995)
ENDIF
 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', & ///,'2X,' 95% CONFIDENCE INTERVALS ON C AND m ', & 'FOR EACH REGION',/)
905 FORMAT(7X,'REGION: ',I1,7X,'Io = ('.FI2 , ' ',F12.9,')', & '/,'F12.9,)')
910 FORMAT(///,'REGION',8X,'E(C)',I2X,'E(m)',/)
915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)
920 FORMAT(///,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)
930 FORMAT(6X,I1,8X,F8.4,8X,F8.4)
940 FORMAT(///)
950 FORMAT(//,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)
960 FORMAT(///,'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, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR,
& REFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

C INPUTS: ---
C OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
& 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),
LIST OF VARIABLES

DELTA()  1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
IOUT   OUTPUT DUMP CONTROLLER
J     CONTROLS DO LOOP FOR EACH DATA SET
K     CONTROLS DO LOOP FOR EACH REGION
L     CONTROLS DO LOOP FOR EACH REGION
LNNF()  3-D ARRAY CONTAINING LN(RAWN()) (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
LNSTR()  3-D ARRAY CONTAINING LN(RATSTR()) (TOTAL STRAIN DATA (%)) FOR ALL S/N DATA SETS BROKEN INTO REGIONS
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 BROKEN INTO REGIONS
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 BROKEN INTO REGIONS
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 425 K = 1, MAXDAT
      NF(K, L) = 0.0
      STR(K, L) = 0.0
425 CONTINUE
400 CONTINUE
C** SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATO; "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 
C PROGRAMMER: L. NEWLIN 
C DATE: 21JUN88 FORMAT/COMMENTS: 12AUG91 
C INPUTS: VARY, MPROC 
C OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, 
&
C SUBPROGRAMS: TRMNAT, CONVRT 
C IMPLICIT NONE 
INTEGER MAXDAT, MAXREG, MAXSET 
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5) 
COMMON IOUT 
INTEGER COUNT, I, IOUT, J, K, L, M, MPNT(MAXREG), MPROC, NDIV, 
&
& REAL CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ, 
&
& 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 
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND 
SIG() CALCULATION 
DESCRP() 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
OUTPUT DUMP CONTROLLER
CONTROLS DO LOOP FOR EACH DATA SET
CONTROLS DO LOOP FOR EACH POINT IN A REGION
RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
CONTROLS DO LOOP FOR EACH REGION
LAMBDA-N — RATIO OF VARI (LN N given S) / (m**2 C**2), CONSTANT OVER ALL REGIONS AND COMPONENTS
3-D ARRAY CONTAINING LN(RAWNF( )!), ALSO INDEXED FOR REGION
3-D ARRAY CONTAINING LN(RATSTR( ), ALSO INDEXED FOR REGION
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 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
2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
NUMBER OF REGIONS OF INTEREST
STRESS RATIO (R = -1.0 IS DESIRED)
2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS
2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
REGION OF INTEREST IN A PARTICULAR DIVISION
1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
2-D ARRAY CONTAINING RATSTR( ) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
STRESS TENSILE TEST POINT, SO
STRESS TENSILE TEST POINT, SO
CONTROLS CURVE VARIATION DESIRED- 0 - NO VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE — 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

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), 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), FTU, FTU, NPTS(0)
IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(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(8,*), 'ERROR: INVALID VALUE FOR RATIO: ', RATIO
      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)
      READ(1,*), RAWSTR(I,0), RAWNF(I,0)
      CONTINUE
   C CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
   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

INSTR(K, 0, REG) = ALOG(RATSTR(I, 0))
LNSTR(K, 0, REG) = ALOG(RATSTR(I, 0))
SMF(K, REG) = RATSTR(I, 0)
NF(K, REG) = RAWNF(I, 0)

CONTINUE

IF (K .GT. MAXDAT) THEN
WRITE(8, *) 'ERROR: OVER NUMBER OF POINTS LIMIT IN ',
& 'SPECIFIC MATERIAL'
CALL TRMNAT
ENDIF
NP(0, REG) = K
NP(MFREG) = K
COUNT = COUNT + NUM

CONTINUE

IF (NPTS(0) .NE. COUNT) THEN
WRITE(8, *) 'ERROR: NUMBER OF POINTS PER DIVISION ',
& 'INCORRECTLY SPECIFIED'
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, NNOGAT
IF ((NUMREG + NNOGAT) .GT. MAXREG) THEN
WRITE(8, *) 'ERROR: EXCEEDED LIMIT ON NUMBER OF REGIONS'
CALL TRMNAT
ENDIF
DO 150 L = ZROREG, (NUMREG + NNOGAT)
READ(1, *) NBND(L)
150 CONTINUE
READ(1, *) CZERO
DO 160 L = 1, (NUMREG + NNOGAT)
READ(1, *) MFPN(L), MZERO(1, L), MZERO(2, L)
160 CONTINUE
WRITE(3, 913)
IF (ZROREG .EQ. 0) WRITE(3, 914) SZERO
IF (IOUT .EQ. 10) THEN
WRITE(8, 913)
IF (ZROREG .EQ. 0) WRITE(8, 914) SZERO
ENDIF
WRITE(3, 915) NUMREG, NNOGAT
IF (IOUT .EQ. 10) WRITE(8, 915) NUMREG, NNOGAT
DO 170 L = ZROREG, (NUMREG + NNOGAT)
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 + NNOGAT)
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 + NNODAT)
  IF (IOUT .EQ. 10)
    WRITE (8, 940) L, DELTA(L), MO(L), SIGMA2(L)
  IF (IOUT .EQ. 10)
    WRITE (8, 940) 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,'
& 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 RELATD
C AND THEN ECHO TO RELATO
READ (5, *) NSETS
IF (NSETS .GT. MAXSET) THEN
  WRITE (8, 940) '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, NPTS(J)
  IF (NPTS(J) .GT. MAXDAT) THEN
    WRITE (8, 940) 'ERROR: OVER LIMIT ON NUMBER OF POINTS IN','
    'SET ', J
    CALL TRMNAT
  ENDIF
WRITE (6, 940) DESCRP(J), FTU, FTY, NPTS(J)
WRITE (6, 945) DESCRP(J), FTU, FTY, NPTS(J)
DO 300 M = 1, NDIV
  READ (5, *) NUM, RATIO, REG
  IF (ABS(RATIO) GT. 1.0) THEN
    WRITE (8, 940) 'ERROR: INVALID VALUE OF RATIO: ', RATIO
    CALL TRMNAT
  ENDIF
300 CONTINUE
IF (REG .GT. MAXREG) THEN
WRITE(*,*) 'ERROR: OVER REGION LIMIT IN RELATED MATERIAL ', J
CALL TRMNAT
ENDIF

IF (IOUT .EQ. 10) THEN
WRITE(*,*) 'NUM = ', NUM, ' COUNT = ', COUNT
WRITE(*,*) '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
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)
& 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(*,*) '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)
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):',2X)
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,II,12X,F7.4,gX,F7.4)
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,'mo',10X,'SIGMA2',///)
945 FORMAT(///,2X,'MATERIALS PROCESS VARIATION INFORMATION',
 & ///,2X,'MEDK*/MEDK:',5X,E11.5,///,5X,'LAMBDAN:',5X,E11.5)
950 FORMAT(///)
955 FORMAT(/,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS',
 & /,2X,'REGION',5X,'DELTA',8X,'mo',10X,'SIGMA2',///)
960 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, 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
7 - 298
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)
C RSTR() STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
C 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 SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C AND, SUBAT2, 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
SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR)

C INPUTS: NUMREG, NSETS, NP, LNSTR
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), MEANX(0:MAXSET),
& MEANY(0:MAXSET), SUHAT2(MAXREG), SWHAT2(MAXREG),
& SX2(MAXREG), SXY(MAXREG), SY2(MAXREG)

LIST OF VARIABLES

I-D ARRAY CONTAINING SXY(L)/SY2(L) FOR EACH REGION
I-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L)
AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L)
AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
OUTPUT DUMP CONTROLLER
CONTROLS DO LOOP FOR EACH DATA SET
CONTROLS DO LOOP FOR EACH POINT IN A REGION
CONTROLS DO LOOP FOR EACH REGION
3-D ARRAY CONTAINING LN(RAWNFT()), ALSO INDEXED FOR REGION
3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
MAXIMUM NUMBER OF REGIONS ALLOWED
MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION L
AND DATA SET J (X = Ln S)
1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION L
AND DATA SET J (Y = Ln N)
2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMBER OF REGIONS OF INTEREST
1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
(X = Ln S)
1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(X = Ln S, Y = Ln N)
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
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
   CONTINUE
50 CONTINUE

MEAN(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

DO 200 J = 0, NSETS
   FIRST CALCULATE SAMPLE X AND Y MEANS
   FOR DATA SET J IN REGION L
   MEAN(J) = 0.0
   MEANX(J) = 0.0
   IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' J =', J,
      ' NP = ', NP(J,L)
   DO 250 K = I, NP(J,L)
      MEAN(J) = MEAN(J) + LNNF(K,J,L)
      MEANX(J) = MEANX(J) + LNSTR(K,J,L)
   CONTINUE
   MEANY(J) = MEAN(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)
   C NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY,
   C OF X AND Y FOR EACH REGION BY SUMMING OVER EACH
   C DATA SET IN REGION L
   DO 300 K = 1, NP(J,L)
      DIFFY(K,J) = LNNF(K,J,L) - MEAN(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),
            ' SXY(L) = ', SXY(L),
            ' SX2(L) = ', SX2(L),
            ' SY2(L) = ', SY2(L),
            ' MEANX(J) = ', MEANX(J),
            ' MEAN(J) = ', MEAN(J)
      ENDIF
300 CONTINUE

NPPR(L) = NPPR(L) + NP(J,L) - 1
IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) = ', NPPR(L)
200 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. 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
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 5OCT87 COMMENTS: 15SEP89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
V8.4, V8.5
C
SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
 & MCHAT)
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 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)
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

7 - 303
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 SX/(SX2/L) FOR EACH REGION
I\ CONTROLS LOOP FOR CHI025() AND CHI975()
IOUT\ OUTPUT DUMP CONTROLLER
IZERO()\ 2-D ARRAY CONTAINING I0, THE 95% CONFIDENCE INTERVALS ON C
FOR EACH REGION
JZERO()\ 2-D ARRAY CONTAINING J0, 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 NPPL(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
SIX\ REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SWHAT\ EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
SWHAT2()\ 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
SX2()\ EQUAL TO (NPPL(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
SX2()\ 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
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 (NPPL(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
C CONFIDENCE INTERVALS FOR EACH REGION
DO 100 L = 1, NUMREG
  NUM = NPPL(L)
  IF (NUM .LT. 31) THEN
    " ...
  ELSE
    " ...
  ENDIF
100 CONTINUE

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(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,* 'T =', T, ' DD =', DD(L), ' ARG =', ARG
WRITE(8,* 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) ='
& WRITE(8,* 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) ='
& WRITE(8,* 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(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: CODE: 8OCT87  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 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)

7 - 305
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 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) = - SY2(L) / (2.0 * SXY(L))
    ELSE
        ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)
        IF (ARG2 .LT. 0.0) THEN
            WRITE(8,*), 'ERROR: Co TOO LOW'
            CALL TRMMAT
        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

\[ \text{SXY}(L) \cdot \text{GT. CZERO2} - \text{THIS TIME THE M CONSTRAINT IS A RANGE} \]

\[ \text{MCPNT}(L) = 2 \]
\[ \text{MC}(1,L) = (- \text{SXY}(L) - \text{ARG2}) / \text{ARG1} \]
\[ \text{MC}(2,L) = (- \text{SXY}(L) + \text{ARG2}) / \text{ARG1} \]

ENDIF

ENDIF

100 CONTINUE

IF (IOUT .EQ. 10) THEN

DO 200 L = 1, NUMREG

WRITE(8,*) 'L = ', L, ' MARGT = ', MARGT(L)
WRITE(8,*) 'ARGL = ', ARGL, ' 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
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, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG), &
NUMREG, TOTAL
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),
C

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         SUM OF NUM() OVER ALL REGIONS
TOTAL       SUM OF NUM() OVER ALL REGIONS

C

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
    DO 300 L = 1, NUMREG
        WRITE(8,*),'L = ', L, ' NUM = ', NUM(L)
        WRITE(8,*),'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
    300 CONTINUE
ENDIF

PVAR = SUM / FLOAT(TOTAL)

RETURN
END

C***************************************************************

C SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
M AND C WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C PROGRAMER: J. NEWLIN
C DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91
            V8.4, V8.5
            SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
                MCHAT, RANGEH)
INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
OUTPUTS: RANGEM
SUBPROGRAMS: TRMNAT

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
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() 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

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(*,*) '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(1,L) = ', JZERO(1,L)
  ENDIF
100 CONTINUE

7 - 309
WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L), ' RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN
C
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 < 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(1,L), ' JZERO(2,L) = ', JZERO(2,L)
WRITE(8,*) '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. 0) .AND. (MCPNT(L) .EQ. 2)) THEN
C
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 < 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(1,L), ' JZERO(2,L) = ', JZERO(2,L)
WRITE(8,*) '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
C
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
C
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 < LOWER) THEN
WRITE(8,*) 'ERROR: NO INTERSECTION BETWEEN JO AND MC'

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
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

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
C IOUT OUTPUT DUMP CONTROLLER
C L CONTROLS DO LOOP FOR EACH REGION
C MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
C MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
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 MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C MZERO() FOR EACH REGION
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)
SUBROUTINE CONCAV ADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY RANGES ON M TO BE CONSISTENT WITH CONCAVITY CONSTRAINTS

PROGRAMMER: L. NEWLIN

DATE: 29E88 FORMAT/COMMENTS: 15SEP89


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

C**************************************************************

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 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
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(1, 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

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) .LT. 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) = AMINI(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

C******************************************************************************

C SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR CO
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 5OCT87 COMMENTS: IDEC87
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 MEDIAN (NUMREG, RANGEM, MEDM)
C INPUTS: NUMREG, RANGEM
C OUTPUT: MEDM
C IMPLICIT NONE

7 - 314
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

C BEGIN CALCULATIONS FOR EACH REGION
DO 100 L = 1, NUMREG
   IF (RANGEM(2,L) .EQ. 0.0) THEN
      RANGE IS A POINT
      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) = (LOWERL + RANGEM(2,L)) / 2.0
   ENDIF
   IF (IOUT .EQ. 10) THEN
      WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
      WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
      & ' RANGEM(2,L) = ', RANGEM(2,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

SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, & ZROREG, NBND, BIGK1, BZHAT)

INPUTS: NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
OUTPUTS: BIGK1, 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, & REAL & MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG), & & NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG), & & SZ2, SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)

LIST OF VARIABLES

BIGK() 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
KHAu 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)
    BIGK1 = 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
        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

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) = ',
            F8.4,5X,'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)
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)

LIST OF VARIABLES

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()-I))-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)
C SX2()  1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C       (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 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)
      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 SUBPROGRAMS: TRMNAT

SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT

7 - 319
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
C 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
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
    WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,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
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),
  & 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. (MCPNT(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

IF (IOUT .EQ. 10) THEN
  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(1,L),
  & RANGEM(2,L) = ', RANGEM(2,L)
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 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

IF (IOUT .EQ. 10) THEN
  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(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
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 ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C 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,
& MO, SIGMA2

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
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
C SIGM( )
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

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
      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)
         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
C INPUTS: VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND, NBOUND, STR
C OUTPUTS: BIGK, BZERO, MM, SBND
C SUBPROGRAMS: FINDM, FINDMN, TRANSFM, SMNVAR, KBETA, FINDK, FINDSB
C
C IMPLICIT NONE
C
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), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

DOUBLE PRECISION RAND

C
C
C
C BIGK()
C 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
C
C BZERO
VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING S/N DATA SET
C
C IOUT
OUTPUT DUMP CONTROLLER
C
C K
VALUE OF K -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C
C MAXD
MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C
C MAXREG
MAXIMUM NUMBER OF REGIONS ALLOWED
C
C MEANZ
SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C
C MM()
1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C
C MU()
1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
C
C NBND()
1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
C
C NF()
2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C
C NP
TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C
C NPTS()
1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET
C
C NUMREG
NUMBER OF REGIONS OF INTEREST
C
C 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
C RAND
RANDOM NUMBER SEED
C
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
C SIG()
1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C
C STR()
2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C
C SZ2
SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C
C VARY
CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
C
C ZROREG
ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C
C ZZ()
1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
Z = F(STR, NF, NBND, MM)
C
C
C OBTAIN THE VALUES OF M FOR EACH REGION
C
IF (VARY .LE. 2) THEN
C
UNIFORM OR NO VARIATION IN M IS DESIRED
CALL FINDM (RAND, NUMREG, RANGEM, MM)
ELSE

7 - 324
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
 WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)

200 CONTINUE

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,11,5X,F9.5,5X,E12.5,5X,E9.3,6X,E11.5)

920 FORMAT(///)

RETURN

END

************************************************************************************************

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOUT</td>
<td>OUTPUT DUMP CONTROLLER</td>
</tr>
<tr>
<td>L</td>
<td>CONTROLS DO LOOP FOR EACH REGION</td>
</tr>
<tr>
<td>MAXREG</td>
<td>MAXIMUM NUMBER OF REGIONS ALLOWED</td>
</tr>
<tr>
<td>NUMREG</td>
<td>NUMBER OF REGIONS OF INTEREST</td>
</tr>
<tr>
<td>PICK()</td>
<td>1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM</td>
</tr>
<tr>
<td>RAND</td>
<td>RANDOM NUMBER SEED</td>
</tr>
<tr>
<td>RANGEM()</td>
<td>2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M</td>
</tr>
<tr>
<td>X</td>
<td>UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED OFF THE RANGE ON M</td>
</tr>
</tbody>
</table>

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), '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), '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

C


C PROGRAMMER: L. GRONDALSKI, L. NEWLIN

C DATE: IDEC87


SUBROUTINE RANDOM (FRAC, RAND)
IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL FRAC
DOUBLE PRECISION RANA, RANC, RAND, RANDIV, RANM, RANSUB, RANT, RANX

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
SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION

PROGRAMMER: L. NEWLIN
DATE: CODE: 7JUN88 COMMENTS: 13FEB89

VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5


SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)

INPUTS: RAND, NUMREG, MU, SIG, RANGEM
OUTPUTS: MM
SUBPROGRAMS: NORMGN, TRMNAT

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

C LIST OF VARIABLES

OUTPUT DUMP CONTROLLER
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 THE MEAN OF M FOR EACH REGION
NUMBER OF REGIONS OF INTEREST
1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RANDOM NUMBER SEED
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 THE STANDARD DEVIATION OF M FOR EACH
REGION
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
M IS SPECIFIED AS A POINT VALUE
MM(L) = RANGEM(1,L)
ELSEIF (L .EQ. I) THEN
WRITE(8,*) 'RANGEM(I,L) =', RANGEM(I,L),
', MM(L) =', MM(L)
ENDIF
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
C 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
ELSE
C SAMPLE ON ADJUSTED RANGE
CALL NORMGN (RAND, MU(L), SIG(L), X)
IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
MM(L) = X
ENDIF
END}

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 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 N(0,1)
Z2    NORMAL RANDOM NUMBER N(0,1)

IF ((IOUT .EQ. I0) .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. I0) .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. I0) .OR. (IOUT .EQ. 15))
&   WRITE(8,*) 'Z1 = ', Z1, ' Z2 = ', Z2, ' X = ', X
RETURN
END

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
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)
C INPUTS:   NPTS, STR, NF, NUMREG, MM, NBND
C OUTPUTS:  NF, ZZ
C
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)

LIST OF VARIABLES
C 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 NF() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
C NUMREG TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C NF() 1-D ARRAY CONTAINING THE NUMBER OF REGIONS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET
C STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL 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,'(A,F9.1)') 'L = ', L, ' MM = ', MM(L), ' MML = ', MML," NPTS = ', NPTS(L)
   DO 200 K = 1, NPTS(L)
      ZZ(NP) = ALOG(STR(K,L)) + ALOG(NF(K,L)) * (1.0 / MML)
      IF (IOUT .EQ. 10) WRITE(8,'(A,F9.1)') 'K = ', K, ' NF = ', NF(K,L), ' ZZ = ', ZZ(NP)
   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 SUBROUTINE SMNVAR (NP, ZZ, MEANZ, S22)
C INPUTS: NP, ZZ
C OUTPUTS: MEANZ, SZ2
C
C IMPLICIT NONE
INTEGER MAXDAT
PARAMETER (MAXDAT = 50)
COMMON IOUT
INTEGER I, IOUT, NP
REAL MEANZ, SZ2, ZZ(MAXDAT)
C
C I OUT PUTS DUMP CONTROLLER
C MAXD AT 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 SZ2 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 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.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
C VARIANCE OF Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87 COMMENTS: 13JUL89
C V8.4, V8.5
C SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)
C INPUTS: MEANZ, SZ2
C OUTPUTS: K, BZERO

7 - 332
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 SAMPLE VARIANCE OF THE TRANSFORMED DATA,
SZ2 ** 0.5

PERFORM CALCULATIONS
SZ = SZ2 ** 0.5
BZERO = PI / (SZ * (6.0 ** 0.5))
K = MEANZ

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

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,
   & ' Gamalm = ', 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 ', L, ', K = ', BIGK(L)
   IF (IOUT .EQ. 10) WRITE(8,*)'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*****************************************************************************

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
INTEGER MAXREG

C*****************************************************************************
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

   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: 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
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

LIST OF VARIABLES
ARG INTERMEDIATE CALCULATION VARIABLE
BETA WEIBULL DISTRIBUTION SHAPE PARAMETER
ETA WEIBULL DISTRIBUTION LOCATION PARAMETER
FRAC UNIFORM (0,1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
RAND RANDOM NUMBER SEED
WEIB WEIBULL(BETA, ETA) GENERATED RANDOM VARIATE

CALCULATE CONSTRAINED ETA
ETA = 1.0 / ( ALOG(2.0) ** (1.0 / BETA))

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,'(A,A,F20.14)')'BETA = ', BETA, ' ETA = ', ETA,
& ' FRAC = ', FRAC, ' ARG = ', ARG, ' WEIB = ', WEIB
RETURN
END

******************************************************************************

SUBROUTINE KOMO CALCULATES KO AND MO FOR THE ZERO REGION (NO DATA
REGION TO THE LEFT). IT ACCOUNTS FOR TYPING 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
INTEGER 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 SBND(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
   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

IF (IOUT .EQ. 1) THEN
   WRITE(8, *) 'SZERO = ', SZERO, ' BIGK0 = ', TRBIGK(0)
   WRITE(8, *) 'FACTOR = ', FACTR, ' BIGK1 = ', TRBIGK(1)
   WRITE(8, *) 'MM1 = ', MM(1), ' 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, ZROREG, NUMREG, SZERO)

C INPUTS: S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C OUTPUTS: GTLIFE
C IMPLICIT NONE

7 - 337
INTEGER IOUT, L, MAXREG, NUMREG, ZROREG
PARAMETER (MAXREG = 3)
COMMON IOUT, GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
& MM(0:MAXREG), S, SBND(0:MAXREG), ZERO, 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
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()
ZERO -- STRESS TENSILE POINT, SO TEMP -- TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
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. ZERO) .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)
     ENDIF
  ENDIF
100 CONTINUE
ENDIF
150 CONTINUE

GTLIFE = GETLIF

RETURN

C*******************************************************************************

C SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
M FOR EACH REGION
PROGRAMMER: L. NEWLIN
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 BESORTED 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
10   IF (INC .GT. 1) THEN
      INC = INC / 2
20   INC = (NUM - INC) / 2
      INORDR = .TRUE.
      DO 300 I = 1, NUM
         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
