An Improved Approach for Flight Readiness Certification—Methodology for Failure Risk Assessment and Application Examples
Volume III: Structure and Listing of Programs

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

June 1, 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 the methodology for evaluating flight readiness developed by the Jet Propulsion Laboratory (JPL) under NASA RTOP 553-02-01 sponsored by the Office of Space Flight (OSF), NASA Headquarters. This methodology was developed 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.

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 engineering analysis 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 engineering analyses 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 engineering analysis and operating experience is needed to make effective risk management decisions and utilize financial resources efficiently.

The PFA methodology is applicable to failure modes that can be characterized by analytical or empirical modeling of failure phenomena 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.

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

---

1 See [3] of Section 1.0 references.
formulated probabilistic engineering analysis methods and models. M. Creager made major contributions to defining and formulating the probabilistic modeling approach and engineering analysis procedures used in this work. Present or former JPL personnel who made substantial contributions in early phases of this work include D. L. Schwartz, W. E. Edmiston, and L. J. Grondalski. D. Goode and J. Ramsay 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, 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 the above-listed organizations and by 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 three 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 user's guides and flowcharts for the computer software used to implement PFA in the application examples. Volume III presents the structure and listings of the computer programs.

\[2\] Currently of Structural Integrity Engineering, Chatsworth, CA.
Acknowledgments

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

The application examples of this report were performed in collaboration with Rocketdyne, Canoga Park, California, and Pratt & Whitney, West Palm Beach, Florida. Several individuals at each organization contributed generously to this work, including E. P. Fox and C. G. Annis of Pratt & Whitney, and K. J. O'Hara and D. O'Connor of Rocketdyne. 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.

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.

The authors appreciate the technical comments provided by L. Beitch and A. Coles of the General Electric Co., Cincinnati, Ohio, on flight readiness certification approaches and fatigue life modeling. The effort of G. R. Halford of the NASA Lewis Research Center in providing information on materials fatigue life modeling is appreciated.

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 engineering analysis 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 engineering analyses of failure phenomena, is expensive or difficult to acquire.

The PFA methodology is a prescribed statistical structure in which engineering analysis 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. Conventional engineering analysis models currently employed for design or failure prediction 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 useful when models or information used in analysis are uncertain or approximate. Such failure modes include, but are not limited to, fatigue, flaw propagation, rupture, degradation and wear, and malfunction of mechanical or electrical 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 engineering analyses 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 engineering analysis is needed to make effective risk management decisions.
This report describes the PFA methodology and presents examples of its application. Conventional approaches to failure risk evaluation for spaceflight systems are discussed, and the rationale for the approach taken in the PFA methodology is presented. The statistical methods, engineering models, and computer software used in fatigue failure mode 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 .......................................... xxiii
- List of Tables ........................................... xxviii

## 1.0 Introduction ........................................ 1-1

### 1.1 Flight Readiness Assessment .......................... 1-3
  - 1.1.1 The Flight Readiness Assessment Problem .......... 1-3
  - 1.1.2 Conventional Approaches to Flight Readiness Assessment
    - 1.1.2.1 Testing to Establish Flight Readiness .......... 1-5
    - 1.1.2.2 Deterministic Engineering Analysis .......... 1-9

### 1.2 An Improved Approach to Flight Readiness Assessment 1-11
  - 1.2.1 Failure Risk Assessment .......................... 1-11
  - 1.2.2 The Probabilistic Failure Assessment Methodology 1-12
  - 1.2.3 Probabilistic Failure Modeling ..................... 1-14
  - 1.2.4 Driver Characterization ............................ 1-15
  - 1.2.5 Computational Methods ............................ 1-17

### 1.3 Implementing the PFA Methodology ...................... 1-19

### 1.4 Report Organization .................................. 1-21

## References ............................................. 1-24

## Appendix 1.A The Limitations of Testing for Reliability Demonstration 1-27

## Appendix 1.B List of Acronyms ............................ 1-31

## 2.0 Methodology .......................................... 2-1

### 2.1 Statistical Analysis .................................. 2-3
  - 2.1.1 Failure Simulation Statistics ..................... 2-3
  - Reference ............................................. 2-6
  - 2.1.2 Materials Fatigue Life Characterization .......... 2-6
    - 2.1.2.1 Stress/Life Characterization of Fatigue Failure of Materials 2-6
    - References ........................................... 2-18
    - 2.1.2.2 Strain/Life Characterization of Fatigue Failure of Materials 2-19
    - 2.1.2.3 Process Variation in Materials ................. 2-20
  - 2.1.3 Driver Characterization ............................ 2-22
3.0 Application Examples

3.1 Elbow Duct HCF Analysis
   3.1.1 HPOTP Main Discharge Duct Description
   3.1.2 LPFTP Turbine Drive Duct Description
   3.1.3 Driver Description for Elbow Ducts
   3.1.4 HPOTP Main Discharge Duct Analysis
   3.1.5 Results for HPOTP Main Discharge Duct
   3.1.6 LPFTP Turbine Drive Duct Analysis
   3.1.7 Results for LPFTP Turbine Drive Duct

3.2 HPOTP Heat Exchanger Coil HCF Analysis
   3.2.1 Component Description
   3.2.2 Driver Description
   3.2.3 Analysis
   3.2.4 HEX Coil Results

3.3 ATD-HPFTP Second Stage Turbine Disk LCF Analysis
   3.3.1 Component Description
   3.3.2 Driver Description
   3.3.3 Analysis
   3.3.4 Results

References

Appendix 3.A Probabilistic Failure Assessment Details
   3.A.1 Introduction
   3.A.2 HPOTP Heat Exchanger Coil HCF Analysis Details
      3.A.2.1 Selecting the Component, Failure Mode, and Critical Location
      3.A.2.2 Preliminary Deterministic Analysis
      3.A.2.3 Driver Characterization
      Weld Offset
      Stress Concentration Factor
      Wall Temperature and Internal Pressure
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1.2</td>
<td>MATCHR Program</td>
</tr>
<tr>
<td>4.1.2.1</td>
<td>Stress Formulation</td>
</tr>
<tr>
<td>4.1.2.2</td>
<td>Strain Formulation</td>
</tr>
<tr>
<td>4.1.3</td>
<td>INFAGG Routine</td>
</tr>
<tr>
<td>4.1.3.1</td>
<td>Routine INIT</td>
</tr>
<tr>
<td>4.1.3.2</td>
<td>Routine RCE</td>
</tr>
<tr>
<td>4.1.3.3</td>
<td>Routine CONVRT</td>
</tr>
<tr>
<td>4.1.3.4</td>
<td>Routine SW2SU2</td>
</tr>
<tr>
<td>4.1.3.5</td>
<td>Routine FINDMC</td>
</tr>
<tr>
<td>4.1.3.6</td>
<td>Routine INTRVL</td>
</tr>
<tr>
<td>4.1.3.7</td>
<td>Routine GTPVAR</td>
</tr>
<tr>
<td>4.1.3.8</td>
<td>Routine FNDRNG</td>
</tr>
<tr>
<td>4.1.3.9</td>
<td>Routine ADDREG</td>
</tr>
<tr>
<td>4.1.3.10</td>
<td>Routine CONCAV</td>
</tr>
<tr>
<td>4.1.3.11</td>
<td>Routine MEDIAN</td>
</tr>
<tr>
<td>4.1.3.12</td>
<td>Routine EXPCTD</td>
</tr>
<tr>
<td>4.1.3.13</td>
<td>Routine MUSIG</td>
</tr>
<tr>
<td>4.1.3.14</td>
<td>Routine NORRNG</td>
</tr>
<tr>
<td>4.1.3.15</td>
<td>Routine ADDRGN</td>
</tr>
<tr>
<td>4.1.4</td>
<td>Routine DECOMP</td>
</tr>
<tr>
<td>4.1.4.1</td>
<td>Routine INITD</td>
</tr>
<tr>
<td>4.1.4.2</td>
<td>Routine RDECHO</td>
</tr>
<tr>
<td>4.1.4.3</td>
<td>Routine PREP</td>
</tr>
<tr>
<td>4.1.4.4</td>
<td>Routine PECOMP</td>
</tr>
<tr>
<td>4.1.5</td>
<td>PAREST Routine</td>
</tr>
<tr>
<td>4.1.5.1</td>
<td>Routine FINDM</td>
</tr>
<tr>
<td>4.1.5.2</td>
<td>Routine FINDMN</td>
</tr>
<tr>
<td>4.1.5.3</td>
<td>Routine TRNSFM</td>
</tr>
<tr>
<td>4.1.5.4</td>
<td>Routine SMNVAR</td>
</tr>
<tr>
<td>4.1.5.5</td>
<td>Routine KBETA</td>
</tr>
<tr>
<td>4.1.5.6</td>
<td>Routine FINDK</td>
</tr>
<tr>
<td>4.1.5.7</td>
<td>Routine FINDSB</td>
</tr>
<tr>
<td>4.1.6</td>
<td>Routine KOMO</td>
</tr>
<tr>
<td>4.1.7</td>
<td>Routine ADJSTM</td>
</tr>
<tr>
<td>4.1.8</td>
<td>Routine GTLIFE</td>
</tr>
<tr>
<td>4.1.9</td>
<td>Routine GTLIF2</td>
</tr>
<tr>
<td>4.1.9.1</td>
<td>Routine NEWTON</td>
</tr>
<tr>
<td>4.1.9.2</td>
<td>Routine FCT</td>
</tr>
<tr>
<td>4.1.10</td>
<td>Routine SORTM</td>
</tr>
<tr>
<td>4.1.11</td>
<td>Routine TRMNAT</td>
</tr>
<tr>
<td>4.2</td>
<td>Prior Distribution Parameter Estimation Software</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Introduction</td>
</tr>
</tbody>
</table>
4.2.2 BFIT Program .................................. 4-73
   4.2.2.1 LLS Routine .................................. 4-73
4.2.3 ABTFIT Program .................................. 4-76
   4.2.3.1 ABT Routine .................................. 4-80
   4.2.3.2 JABT Routine .................................. 4-80
4.2.4 LZERO Program .................................. 4-80
   4.2.4.1 Routine GAMMA .................................. 4-83
   4.2.4.2 Routine DLGAM .................................. 4-83
   4.2.4.3 Routine MUELLR .................................. 4-83
   4.2.4.4 Routine FCT .................................. 4-83
   4.2.4.5 Routine TRMNAT .................................. 4-83

4.3 Bayesian Statistical Procedure Software .................. 4-85
   4.3.1 Introduction ..................................... 4-85
   4.3.2 BAYES Program ..................................... 4-85

4.4 Random Number Generation Software .................... 4-87
   4.4.1 Introduction ...................................... 4-87
   4.4.2 RANDOM Routine ................................... 4-87
   4.4.3 NORMGN Routine ................................... 4-87
   4.4.4 GAM Routine ...................................... 4-87
   4.4.5 BETAGN Routine ................................... 4-87
   4.4.6 WEIBGN Routine ................................... 4-88

4.5 Reference Time History Generation Software ............ 4-89
   4.5.1 Introduction ...................................... 4-89
   4.5.2 NBSIN Program ..................................... 4-89

References ............................................. 4-95

5.0 Fatigue Analysis Software ................................ 5-1

5.1 High Cycle Fatigue Analysis Software ................. 5-3
   5.1.1 Introduction ....................................... 5-3
   5.1.2 DCTHCF Program ..................................... 5-4
       5.1.2.1 Main Routine ..................................... 5-4
       5.1.2.2 ELWELD Routine ..................................... 5-8
       5.1.2.3 M2L1 Routine ..................................... 5-8
       5.1.2.4 CALCS Routine ..................................... 5-11
       5.1.2.5 NARBN1 Routine ..................................... 5-11
       5.1.2.6 RAINFO1 Routine ..................................... 5-11
       5.1.2.7 PGETSM Routine ..................................... 5-21
   5.1.3 HEXHCF Program ..................................... 5-21
       5.1.3.1 Main Routine ..................................... 5-21
       5.1.3.2 THWELD Routine ..................................... 5-26
       5.1.3.3 M4L1 Routine ..................................... 5-28
5.1.3.4 NARB2 Routine ........................................ 5-28
5.1.3.5 RAINF2 Routine ........................................ 5-28
5.1.3.6 NEUBER Routine ....................................... 5-38

5.2 Low Cycle Fatigue Analysis Software .......................... 5-41
5.2.1 Introduction ............................................... 5-41
5.2.2 TRBPWA Program .......................................... 5-41
5.2.2.1 Main Routine .......................................... 5-41
5.2.2.2 Driver Transformation .................................. 5-46

Appendix 5.A Program Flowchart Symbols ......................... 5-49
Appendix 5.B INSORT Routine .................................... 5-51

6.0 Software User's Documentation ................................ 6-1
6.1 High Cycle Fatigue Analysis User's Guides .................... 6-3
6.1.1 DCTHCF Program .......................................... 6-3
6.1.2 How To Use Program DCTHCF ................................ 6-3
6.1.3 Description of Input Data Files .......................... 6-4
  6.1.3.1 Input File DCTHCD .................................. 6-4
        Analysis Parameters Block .............................. 6-10
        Driver Information Block .............................. 6-12
        Load and Geometry Block .............................. 6-18
        Materials Information Block ......................... 6-22
  6.1.3.2 Input File RELATD ................................... 6-25
  6.1.3.3 Reference Time History Files ......................... 6-27

6.1.4 Options and Capabilities ................................ 6-27
6.1.5 Code Execution Example ................................ 6-28
  Input File - DCTHCD ........................................ 6-30
  Input File - RELATD ......................................... 6-32
  Input File - NBP ............................................ 6-32
  Input File - NBM3 ........................................... 6-32
  Input File - SIN10 .......................................... 6-32
  Output File - DCTHCO ....................................... 6-33
  Output File - RELATO ........................................ 6-36
  Output File - DUMP .......................................... 6-36
  Output File - IOUTPR ........................................ 6-37
  Output File - LOWLIF ....................................... 6-38

6.1.6 Error Messages and Possible Remedies ..................... 6-38
6.1.7 Summary of Input/Output Files ........................... 6-43
  Input Files .................................................. 6-43
  Output Files .................................................. 6-43
6.1.8 HEXHCF Program ......................................... 6-44
6.1.9 How To Use Program HEXHCF ................................ 6-44
6.1.10 Description of Input Data Files ................................... 6-45
  6.1.10.1 Input File HEXHCD ........................................... 6-45
      Analysis Parameters Block ................................... 6-49
      Driver Information Block ................................ 6-51
      Load and Geometry block .................................. 6-58
      Materials Information Block ............................... 6-62
  6.1.10.2 Input File RELATD ........................................... 6-65
  6.1.10.3 Reference Time History Files .............................. 6-67
6.1.11 Options and Capabilities ........................................ 6-67
6.1.12 Code Execution Example .......................................... 6-68
  Input File - HEXHCD .............................................. 6-70
  Input File - RELATD .............................................. 6-72
  Input File - NBM3 ................................................. 6-72
  Input File - SIN1 .................................................. 6-72
  Input File - AERO1 ................................................. 6-72
  Output File - HEXHCO ............................................. 6-73
  Output File - RELATO ............................................. 6-77
  Output File - DUMP ................................................. 6-77
  Output File - IOUTPR .............................................. 6-78
  Output File - LOWLIF .............................................. 6-78
6.1.13 Error Messages and Possible Remedies .......................... 6-78
6.1.14 Summary of Input/Output Files ................................ 6-83
  Input Files ....................................................... 6-83
  Output Files ..................................................... 6-83

6.2 Low Cycle Fatigue Analysis User's Guide ......................... 6-85
  6.2.1 TRBPWA Program .............................................. 6-85
  6.2.2 How To Use Program TRBPWA ................................ 6-85
  6.2.3 Description of Input Data Files ............................ 6-86
    6.2.3.1 Input File TRBPWD ...................................... 6-86
      Analysis Parameters Block ................................ 6-89
      Driver Information Block ................................ 6-91
      Load and Geometry Block ................................ 6-93
      Materials Information Block ............................... 6-94
    6.2.3.2 Input File RELATD ...................................... 6-97
  6.2.4 Options and Capabilities ................................... 6-99
  6.2.5 Code Execution Example .................................... 6-100
    Input File - TRBPWD ........................................... 6-101
    Input File - RELATD ........................................... 6-102
    Output File - TRBPWO .......................................... 6-102
    Output File - RELATO .......................................... 6-104
    Output File - DUMP ............................................. 6-104
    Output File - IOUTPR .......................................... 6-105
xvi
Output File - LOWUF .................................................. 6-106
6.2.6 Error Messages and Possible Remedies ...................... 6-110
6.2.7 Summary of Input/Output Files
Input Files .................................................................. 6-114
Output Files .............................................................. 6-114

6.3.1 MATCHR Program ................................................. 6-117
6.3.2 How To Use the Stress Formulation Option of Program MATCHR . 6-117
6.3.3 Description of the Stress Formulation Input Data Files ........ 6-118
   6.3.3.1 Input File SPECFD ............................................. 6-118
       Analysis Parameters Block ........................................ 6-118
       Materials Information Block ....................................... 6-123
   6.3.3.2 Input File RELATD .............................................. 6-127
6.3.4 Options and Capabilities of the Stress Formulation .......... 6-128
6.3.5 Code Execution Example for the Stress Formulation ......... 6-129
   Input File - SPECFD .................................................. 6-129
   Input File - RELATD .................................................. 6-130
   Output File - SPECFO ................................................. 6-130
   Output File - RELATO ............................................... 6-131
   Output File - DUMP ................................................... 6-132
   Output File - IOUTPR ................................................ 6-133
6.3.6 How To Use the Strain Formulation Option of Program MATCHR . 6-133
6.3.7 Description of the Strain Formulation Input Data Files ...... 6-134
   6.3.7.1 Input File SPECFD ............................................. 6-134
       Analysis Parameters Block ........................................ 6-134
       Materials Information Block ....................................... 6-138
   6.3.7.2 Input File RELATD .............................................. 6-142
6.3.8 Options and Capabilities of the Strain Formulation .......... 6-143
6.3.9 Code Execution Example for the Strain Formulation ........ 6-144
   Input File - SPECFD .................................................. 6-144
   Input File - RELATD .................................................. 6-145
   Output File - SPECFO ................................................. 6-145
   Output File - RELATO ............................................... 6-146
   Output File - DUMP ................................................... 6-146
   Output File - IOUTPR ................................................ 6-149
6.3.10 Error Messages and Possible Remedies ....................... 6-150
6.3.11 Summary of Input/Output Files
Input Files .................................................................. 6-156
Output Files .............................................................. 6-156

6.4.1 BFIT Program ....................................................... 6-159
6.5.1 BAYES Program ........................................ 6-181
6.5.2 How To Use Program BAYES .............................. 6-181
6.5.3 Description of the Input Data File ........................ 6-181
   6.5.3.1 Input File BAYESD ................................... 6-182
      Prior Failure Distribution Parameters Block .............. 6-183
      Operating Experience Block .............................. 6-183
6.5.4 Options and Capabilities ................................ 6-183
6.5.5 Code Execution Example ................................. 6-184
   Input File - BAYESD ......................................... 6-184
   Output File - BAYEso ........................................ 6-184
   Output File - UBAYES ....................................... 6-185
6.5.6 Summary of Input/Output Files .......................... 6-185
   Input Files .................................................. 6-185
   Output Files ................................................ 6-186

6.6.1 NBSIN Program .......................................... 6-187
6.6.2 How To Use Program NBSIN ............................... 6-187
6.6.3 Description of the Input Data File .................... 6-187
   6.6.3.1 Input File NBSIN ................................... 6-189
      Generation Parameters Block ............................. 6-189
      Narrow-band Process Information Block ................. 6-191
      Sinusoidal Process Block ................................ 6-192
6.6.4 Options and Capabilities ................................ 6-192
6.6.5 Code Execution Example ................................. 6-192
   Input File - NBSIN .......................................... 6-193
   Output File - IOUTPR ....................................... 6-193
   Output File - AXIAL ........................................ 6-193
   Output File - MOMENT ...................................... 6-194
   Output File - SIN .......................................... 6-194
6.6.6 Summary of Input/Output Files .......................... 6-194
   Input Files .................................................. 6-194
   Output Files ................................................ 6-194

VOLUME III – Structure and Listing of Programs

7.0 Structure and Listing of Programs .......................... 7-1
7.1 High Cycle Fatigue Failure Programs ...................... 7-3
   7.1.1 DCTHCF Program ....................................... 7-3
      7.1.1.1 Program Tree Structure ......................... 7-3
      7.1.1.2 List of Subprograms ........................... 7-3
      7.1.1.3 Description of Variables ....................... 7-9
      7.1.1.4 Program DCTHCF Listing ......................... 7-26
         Program DCTHCF Listing Temporal Order, Uniform Distribution 7-28
7.2 Low Cycle Fatigue Failure Program ........................................ 7-241
  7.2.1 TRBPWA Program ....................................................... 7-241
     7.2.1.1 Program Tree Structure ...................................... 7-241
     7.2.1.2 List of Subprograms .......................................... 7-241
     7.2.1.3 Description of Variables .................................... 7-247
     7.2.1.4 Program TRBPWA Listing ..................................... 7-255
     Program TRBPWA Listing Temporal Order, Uniform Distribution ... 7-256
     Program TRBPWA Listing Temporal Order, Truncated Normal
     Distribution .............................................................. 7-257

7.3 Materials Characterization Program ...................................... 7-325
  7.3.1 MATCHR Program ..................................................... 7-325
     7.3.1.1 Program Tree Structure ...................................... 7-325
     7.3.1.2 List of Subprograms .......................................... 7-325
     7.3.1.3 Description of Variables .................................... 7-335
     7.3.1.4 Program MATCHR Listing ..................................... 7-359
     Program MATCHR Listing Temporal Order, Stress Formulation,
     Uniform Distribution ................................................... 7-361
     Program MATCHR Listing Temporal Order, Stress Formulation,
     Truncated Normal Distribution ....................................... 7-362
     Program MATCHR Listing Temporal Order, Strain Formulation,
     Uniform Distribution ................................................... 7-363
     Program MATCHR Listing Temporal Order, Strain Formulation,
     Truncated Normal Distribution ....................................... 7-365

7.4 Prior Distribution Parameter Estimation Program .................... 7-451
  7.4.1 BFIT Program ......................................................... 7-451
     7.4.1.1 List of Subprograms .......................................... 7-451
     7.4.1.2 Description of Variables .................................... 7-451
     7.4.1.3 Program BFIT Listing ......................................... 7-452
  7.4.2 ABTFIT Program ....................................................... 7-455
     7.4.2.1 Program Tree Structure ...................................... 7-455
     7.4.2.2 List of Subprograms .......................................... 7-455
7.4.2.3 Description of Variables .................................. 7-456
7.4.2.4 Program ABTFIT Listing ................................ 7-457
7.4.3 LZERO Program .................................................. 7-461
   7.4.3.1 Program Tree Structure .................................. 7-461
   7.4.3.2 List of Subprograms ...................................... 7-461
   7.4.3.3 Description of Variables ................................. 7-462
   7.4.3.4 Program LZERO Listing .................................. 7-463

7.5 Bayesian Statistical Procedure Program .......................... 7-471
   7.5.1 BAYES Program ............................................. 7-471
       7.5.1.1 Description of Variables ............................. 7-471
       7.5.1.2 Program BAYES Listing ................................ 7-473

7.6 Random Number Generation Subprograms .......................... 7-477
   7.6.1 RANDOM Subprogram ........................................ 7-477
       7.6.1.1 Description of Variables ............................. 7-477
       7.6.1.2 Subprogram RANDOM Listing .......................... 7-478
   7.6.2 NORMGN Subprogram ......................................... 7-479
       7.6.2.1 Description of Variables ............................. 7-479
       7.6.2.2 Subprogram NORMGN Listing .......................... 7-479
   7.6.3 BETAGN Subprogram ......................................... 7-480
       7.6.3.1 Description of Variables ............................. 7-480
       7.6.3.2 Subprogram BETAGN Listing .......................... 7-481
   7.6.4 GAM Subprogram ............................................ 7-482
       7.6.4.1 Description of Variables ............................. 7-482
       7.6.4.2 Subprogram GAM Listing ............................. 7-482
   7.6.5 WEIBGN Subprogram ......................................... 7-483
       7.6.5.1 Description of Variables ............................. 7-483
       7.6.5.2 Subprogram WEIBGN Listing .......................... 7-483
       7.6.6 PRYRV Subprogram ....................................... 7-484
       7.6.6.1 Description of Variables ............................. 7-484
       7.6.6.2 Subprogram PRYRV Listing ........................... 7-485

7.7 Reference Time History Generation Program ...................... 7-487
   7.7.1 NBSIN Program ............................................. 7-487
       7.7.1.1 Program Tree Structure ............................... 7-487
       7.7.1.2 List of Subprograms ................................. 7-487
       7.7.1.3 Description of Variables ............................. 7-488
       7.7.1.4 Program NBSIN Listing ............................... 7-491
List of Figures

Figure 1-1 Reliability Demonstrated by Zero-Failure Operating Experience for \( N = 2 \) at 95% Confidence ............................................. 1-7
Figure 1-2 Information Sources for Failure Risk Assessment ........................... 1-11
Figure 1-3 The Probabilistic Failure Assessment Methodology ......................... 1-13
Figure 1-4 The Probabilistic Failure Modeling Procedure ................................ 1-14
Figure 1-5 Options for Controlling Failure Risk ........................................... 1-19
Figure 1-6 Reliability Demonstrated By Nonfailure Operating Experience for the Powerhead Assembly ........................................ 1-29
Figure 1-7 Reliability Demonstrated By Nonfailure Operating Experience for the HPOTP Main Discharge Duct ............................................. 1-29
Figure 1-8 Reliability Demonstrated By Nonfailure Operating Experience for the LPFTP Turbine Drive Duct ............................................. 1-30
Figure 1-9 Reliability Demonstrated By Nonfailure Operating Experience for the HPOTP Heat Exchanger ............................................. 1-30
Figure 2-1 Component Failure Mode Monte Carlo Simulation Structure .................. 2-3
Figure 2-2 S/N Curves ................................................................................. 2-18
Figure 2-3 Uniform Distribution .................................................................... 2-24
Figure 2-4 Normal Distribution ..................................................................... 2-24
Figure 2-5 Beta Distributions with \( \rho = 0.5 \) and Different Values of \( \theta \) ....................... 2-25
Figure 2-6 Beta Distributions with \( \rho = 0.7 \) and Different Values of \( \theta \) ....................... 2-26
Figure 2-7 Procedure for Specifying Beta Distribution .................................... 2-26
Figure 2-8 Load Scale Factors ..................................................................... 2-27
Figure 2-9 Superposition of Narrow-Band Gaussian and Sinusoidal Stress-Time Histories ............................................................. 2-30
Figure 2-10 Cumulative Distribution for a Situation with a Lot of Small Damage Effects ...................................................... 2-34
Figure 2-11 Cumulative Distribution for a Situation with a Rare Large Damage Effect ...................................................... 2-34
Figure 2-12 Component with 10-element Symmetry ........................................ 2-35
Figure 2-13 High Cycle Fatigue Failure Modeling Approach ......................... 2-39
Figure 2-14 Duct Structural Analysis Procedure ............................................ 2-41
Figure 2-15 Geometry of an Elbow Duct ....................................................... 2-43
Figure 2-16 Schematic of Duct Stress Analysis ............................................... 2-50
Figure 2-17 Procedure for Rainflow Counting ................................................ 2-52
Figure 2-18 Mean Stress Calculation Assuming an Elastic Perfectly Plastic Stress-Strain Curve ...................................................... 2-53
Figure 2-19 Mean Stress Calculation Using Neuber's Rule ............................... 2-54
Figure 2-20 Calculation Procedure for the High Cycle Fatigue Failure Simulation ...................................................... 2-56
Figure 2-21 Low Cycle Fatigue Failure Modeling Approach ................................ 2-59
Figure 2-22 Calculation Procedure for the Low Cycle Fatigue Failure Simulation ...................................................... 2-60
Figure 2-23 Axial Cross Section of the ATD-HPFTP Turbine Showing the Monolithic Disk ...................................................... 2-62
Figure 2-24 Stylized Radial Cross Section of a Turbine Disk ............................. 2-62
Figure 2-25 Reference Stress Method for ATD Disk LCF Life ........................................ 2-63
Figure 2-26 Radial Cross Section of the Disk Blade Attachment Area Illustrating the Thermal Gradient Between the Lobes .................................................. 2-67
Figure 2-27 Axial Cross Section of the Disk Illustrating K_d, the Stress Factor to Adjust Axially for Two-Dimensional Analyses ........................................ 2-67
Figure 2-28 Illustration of the Stress-Time History for the ATD Disk ........................................ 2-68
Figure 2-29 Structure of the Probabilistic Failure Model for the ATD-HPFTP Second Stage Turbine Disk ................................................................. 2-70
Figure 2-30 Structure of the LCF Failure Simulation for the ATD-HPFTP Second Stage Turbine Disk ................................................................. 2-72
Figure 2-31 Overall Procedure for Fatigue Failure Modes .................................................. 2-74
Figure 2-32 Procedure for Significant Load Identification .................................................. 2-81
Figure 2-33 Preprocessing ................................................................................................. 2-85
Figure 2-34 Cycle Identification ......................................................................................... 2-86
Figure 3-1 Location of the HPOTP Main Discharge Duct .................................................. 3-3
Figure 3-2 Detail of HPOTP Main Discharge Duct Near Weld 6 ........................................ 3-4
Figure 3-3 Inconel 718 Weld Data ......................................................................................... 3-4
Figure 3-4 Location of the LPFTP Turbine Drive Duct ....................................................... 3-5
Figure 3-5 Detail of the LPFTP Turbine Drive Duct Near Weld 32 ........................................ 3-6
Figure 3-6 Incoloy 903 Weld Data ......................................................................................... 3-6
Figure 3-7 Finite Element Discretization of HPOTP Main Discharge Duct Forces Extracted from Node 24 ................................................................. 3-11
Figure 3-8 HPOTP Main Discharge Duct Impact of Weld Offset on Failure Life Distribution ................................................................................... 3-13
Figure 3-9 HPOTP Main Discharge Duct Failure Life Distribution and Driver Sensitivities ................................................................. 3-14
Figure 3-10 HPOTP Main Discharge Duct Risk Equivalent Life Limiting Procedure ................................................................................... 3-15
Figure 3-11 Finite Element Discretization of LPFTP Turbine Drive Duct Forces Extracted from Node 61 ................................................................. 3-16
Figure 3-12 LPFTP Turbine Drive Duct Impact of Weld Offset on Failure Life Distribution ................................................................................... 3-18
Figure 3-13 LPFTP Turbine Drive Duct Failure Life Distribution and Driver Sensitivities ................................................................................... 3-19
Figure 3-14 HPOTP Heat Exchanger .................................................................................. 3-21
Figure 3-15 Detail of the HPOTP Heat Exchanger Coil Near Weld 3 ........................................ 3-22
Figure 3-16 316L and 321 Stainless Steel Parent Material Test Data ........................................ 3-23
Figure 3-17 321 Weld Data Used as Proxy ........................................................................... 3-23
Figure 3-18 Stress Concentration Factor K_T Distribution .................................................. 3-25
Figure 3-19 Finite Element Discretization of HPOTP Heat Exchanger Coil-Forces Extracted from Node 27 ................................................................ 3-27
Figure 3-20 HPOTP Heat Exchanger Impact of Weld Offset on Failure Life Distribution ................................................................................... 3-29
Figure 3-21 HPOTP Heat Exchanger Failure Life Distribution and Driver Sensitivities ................................................................................... 3-30
Figure 3-22 Axial Cross Section of the ATD-HPFTP Turbine Showing the Monolithic Disk ................................................................................... 3-32
Figure 3-23 Stylized Radial Cross Section of a Turbine Disk .................................................. 3-32
Figure 3-24 Axial Cross Section of the Disk ........................................................................... 3-33
Figure 3-25  Inconel 100 Fatigue Life for Notched Specimens .......................... 3-33
Figure 3-26  Driver Distribution for $\Delta T$ .................................................. 3-34
Figure 3-27  ATD-HPFTP Second Stage Turbine Disk Failure Life Distribution .... 3-36
Figure 3-28  Driver Sensitivities for the ATD-HPFTP Second Stage Turbine Disk  
LCF Failure Life ................................................................................................. 3-37
Figure 3-29  Stress Concentration Due to Different Height Welds .................... 3-42
Figure 3-30  $F_x$ Values From Different Sources, Comparison with Curve in Use  
and Accuracy Factor $\lambda_{OFF}$ Impact ............................................................... 3-43
Figure 3-31  Steps of the Probability of Failure Curve Parameter Estimation  
for the HEX HCF ............................................................................................... 3-49
Figure 3-32  Steps of the Probability of Failure Curve Parameter Estimation  
for the ATD-HPFTP Second Stage Turbine Disk LCF Problem ....................... 3-55
Figure 4-1  Main Flowchart for the Materials Characterization Model Program MATCHR . 4-4
Figure 4-2  Flowchart for Subprogram INFAGG, Stress Formulation .............. 4-12
Figure 4-3  Flowchart for Subprogram RCE, Stress Formulation ..................... 4-19
Figure 4-4  Flowchart for Subprogram CONVRT, Stress Formulation ............. 4-23
Figure 4-5  Flowchart for Subprogram SW2SU2 ............................................... 4-24
Figure 4-6  Flowchart for Subprogram FINDMC, Stress Formulation .............. 4-29
Figure 4-7  Flowchart for Subprogram INTRVL, Uniform Distribution .......... 4-31
Figure 4-8  Flowchart for Subprogram GTPVAR ............................................ 4-32
Figure 4-9  Flowchart for Subprogram FNDRNG, Uniform Distribution .......... 4-33
Figure 4-10 Flowchart for Subprogram MEDIAN, Stress Formulation, Uniform Distribution .... 4-37
Figure 4-11 Flowchart for Subprogram EXPCTD ............................................. 4-39
Figure 4-12 Flowchart for Subprogram MUSIG, Truncated Normal Distribution .. 4-41
Figure 4-13 Flowchart for Subprogram NORRNG, Truncated Normal Distribution . 4-43
Figure 4-14 Flowchart for Subprogram DECOMP, Strain Formulation .......... 4-45
Figure 4-15 Flowchart for Subprogram RDECHO, Strain Formulation .......... 4-52
Figure 4-16 Flowchart for Subprogram PECOMP, Strain Formulation .......... 4-55
Figure 4-17 Flowchart for Subprogram PAREST ........................................... 4-57
Figure 4-18 Flowchart for Subprogram FINDM, Uniform Distribution .......... 4-60
Figure 4-19 Flowchart for Subprogram FINDMN, Truncated Normal Distribution .... 4-61
Figure 4-20 Flowchart for Subprogram TRNSFM .......................................... 4-64
Figure 4-21 Flowchart for Subprogram SMNVAR .......................................... 4-65
Figure 4-22 Flowchart for Subprogram FINDK .............................................. 4-66
Figure 4-23 Flowchart for Subprogram SORTM ............................................. 4-70
Figure 4-24 Flowchart for the Prior Failure Distribution Parameter $\beta$ Estimation  
Program BFIT ................................................................................................... 4-74
Figure 4-25 Flowchart for Subprogram LLS ...................................................... 4-75
Figure 4-26 Flowchart for the Prior Failure Distribution Parameter Estimation  
Program ABTFIT .............................................................................................. 4-77
Figure 4-27 Flowchart for the Assurance Calculation Program LZERO ............. 4-81
Figure 4-28  Flowchart for Subprogram GAMMA ................................................. 4-84
Figure 4-29  Flowchart for the Bayesian Statistical Procedure Program BAYES .......... 4-86
Figure 4-30  Flowchart for the Time History Generation Program NBSIN .................. 4-90
Figure 5-1   Structure of the Probabilistic Failure Model for the Elbow Ducts with Welds . 5-5
Figure 5-2   Main Flowchart for the Duct Analysis Program DCTHCF ....................... 5-6
Figure 5-3   Flowchart for the Subprogram ELWELD ........................................... 5-9
Figure 5-4   Flowchart for the Subprogram M2L1 ................................................. 5-10
Figure 5-5   Flowchart for the Subprogram CALCS ............................................. 5-12
Figure 5-6   Flowchart for the Subprogram NARBN1 ........................................... 5-13
Figure 5-7   Flowchart for Subprogram RAINF1 ............................................... 5-15
Figure 5-8   Flowchart for Subprogram PGETSM ............................................... 5-22
Figure 5-9   Structure of the Probabilistic Failure Model for Straight Ducts with Welds and Temperature Differences Across the Wall ............ 5-23
Figure 5-10  Main Flowchart for the HEX Coil Analysis Program HEXHCF .............. 5-24
Figure 5-11  Flowchart for Subprogram THWELD .............................................. 5-27
Figure 5-12  Flowchart for Subprogram M4L1 .................................................. 5-29
Figure 5-13  Flowchart for the Subprogram NARBN2 ......................................... 5-30
Figure 5-14  Flowchart for Subprogram RAINF2 .............................................. 5-32
Figure 5-15  Flowchart for Subprogram NEUBER ............................................. 5-39
Figure 5-16  Structure of the Probabilistic Failure Model for the ATD-HPFTP Second Stage Turbine Disk .......................................................... 5-42
Figure 5-17  Main Flowchart for the ATD Disk LCF Analysis Program TRBPWA ........ 5-43
Figure 5-18  Flowchart of Driver Transformation ............................................... 5-47
Figure 5-19  Program Flowchart Symbols .......................................................... 5-49
Figure 5-20  Flowchart for Subprogram INSORT .............................................. 5-51
Figure 6-1   Format for File DCTHCD ............................................................... 6-5
Figure 6-2   Format for File RELATD ............................................................... 6-9
Figure 6-3   Data Blocks for Input File ............................................................ 6-9
Figure 6-4   Detail of the HPOTP Main Discharge Duct, Near Weld 6 .................... 6-28
Figure 6-5   Format for File HEXHCD ............................................................. 6-46
Figure 6-6   Detail of the HPOTP Heat Exchanger Coil Small Tube Outlet Near Weld 3 ... 6-68
Figure 6-7   Format for File TRBPWD ............................................................ 6-87
Figure 6-8   Format for File SPECFD ............................................................. 6-119
Figure 6-9   Format for File RELATD ............................................................ 6-121
Figure 6-10  Data Blocks for Input File ......................................................... 6-121
Figure 6-11  Format for File SPECFD ............................................................ 6-135
Figure 6-12  Format for File RELATD ............................................................ 6-136
Figure 6-13  Format for File BFITD .............................................................. 6-160
Figure 6-14  Format for File LOWLIF ............................................................ 6-160
Figure 6-15  Format for File PARAMS ............................................................ 6-168
Figure 6-16 Format for File BAYESD .................................................. 6-177
Figure 6-17 Format for File LDAT .................................................. 6-177
Figure 6-18 Format for File BAYESD .................................................. 6-182
Figure 6-19 Data Blocks for Input File ........................................ 6-182
Figure 6-20 Format for File NBSIN ................................................ 6-188
Figure 6-21 Data Blocks for the NBSIN Input File ......................... 6-189
Figure 7-1 Tree Structure for Program DCTHCF for the Uniform Variation In the Materials Shape Parameter $m$ .................................................. 7-4
Figure 7-2 Tree Structure for Program DCTHCF for the Truncated Normal Variation in the Materials Shape Parameter $m$ ........................................ 7-5
Figure 7-3 Tree Structure for Program HEXHCF for the Uniform Variation in Materials Shape Parameter $m$ .................................................. 7-122
Figure 7-4 Tree Structure for Program HEXHCF for the Truncated Normal Variation in Materials Shape Parameter $m$ ........................................ 7-123
Figure 7-5 Tree Structure for Program TRBPWA for the Uniform Variation in Materials Shape Parameter $m$ .................................................. 7-242
Figure 7-6 Tree Structure for Program TRBPWA for the Truncated Normal Variation in Materials Shape Parameter $m$ ........................................ 7-243
Figure 7-7 Tree Structure for the Stress Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameter $m$ ........................................ 7-326
Figure 7-8 Tree Structure for the Stress Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameter $m$ ........................................ 7-327
Figure 7-9 Tree Structure for the Strain Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameters $m_p$ and $m_E$ ........................................ 7-328
Figure 7-10 Tree Structure for the Strain Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameters $m_p$ and $m_E$ ........................................ 7-330
Figure 7-11 Tree Structure For Program ABTFIT ............................. 7-455
Figure 7-12 Tree Structure For Program LZERO ............................ 7-461
Figure 7-13 Tree Structure For Program NBSIN ............................. 7-487
List of Tables

Table 1-1  Index of Topics Contained in the Report ............................................. 1-22
Table 1-2  Index of Software Documentation Contained in the Report .................... 1-23
Table 2-1  The Required Values of $k$ and the Computed Coverage for Sample Sizes of $N$ . 2-29
Table 2-2  Summary of Load Sources, Critical Locations and Significant Force Components ................................................... 2-58
Table 2-3  Driver Distributions for the ATD-HPFTP Second Stage Turbine Disk ....... 2-72
Table 3-1  Driver Distributions for the HPOTP Main Discharge Duct ...................... 3-8
Table 3-2  Driver Distributions for the LPFTP Turbine Drive Duct ......................... 3-9
Table 3-3  HPOTP Main Discharge Duct Beam-End Forces Near Weld 6 ................... 3-12
Table 3-4  LPFTP Turbine Drive Duct Beam-End Forces Near Weld 32 ..................... 3-17
Table 3-5  Driver Distributions for the HPOTP Heat Exchanger Coil .................. 3-24
Table 3-6  HPOTP Heat Exchanger Coil Beam-End Forces Near Weld 3 ................. 3-28
Table 3-7  Driver Distributions for the Turbine Disk ........................................... 3-34
Table 3-8  Scanning Circumference for Critical Angle Causing Minimum Life ........ 3-40
Table 3-9  Weld Offset Measurements .................................................................. 3-41
Table 3-10 Wall Temperature and Internal Pressure at Weld 3 From Engine Balance Model ................................................................. 3-43
Table 3-11 321 SS Welded S/N Data .................................................................. 3-44
Table 3-12 Summary of Materials Characterization Study of 321 Weld Data .......... 3-44
Table 3-13 Lives for Different Random Number Seeds and History Length .......... 3-45
Table 3-14 Von Mises Stress and Damage Indices Due to Each Load Component ..... 3-47
Table 3-15 Load Components Rank Ordered by Damage Indices for Contributing Stress Components ......................................................... 3-48
Table 3-16 Significant Load Identification Checks ............................................... 3-48
Table 3-17 Probability of Failure Curve Parameter Estimates for 6%, 10% and 20% Weld Offset ................................................................. 3-50
Table 3-18 Driver Sensitivity Analysis for 10% Weld Offset .................................. 3-51
Table 3-19 Inconel 100 Notched S/N Data ......................................................... 3-53
Table 3-20 Summary of Materials Characterization Study of IN100 Notched Data .... 3-54
Table 3-21 Driver Sensitivity Analysis for the Turbine Disk ................................ 3-57
Table 4-1 The Seven Cases Considered by Subprogram FNDRNG ....................... 4-35
Table 4-2 The Four Cases Considered by Subprogram NORRNG ....................... 4-42
Table 7-1 List of Subprograms For Program DCTHCF ..................................... 7-3
Table 7-2 List of Variables for Program DCTHCF ............................................ 7-9
Table 7-3 List of Subprograms for Program HEXHCF ....................................... 7-121
Table 7-4 List of Variables for Program HEXHCF ........................................... 7-127
Table 7-5 List of Subprograms For Program TRBPWA .................................... 7-244
Table 7-6 List of Variables For Program TRBPWA .......................................... 7-247
Table 7-7 List of Subprograms for Program MATCHR ..................................... 7-325
| Table 7-8 | List of Variables for Program MATCHR | 7-335 |
| Table 7-9 | Routine/Variable Chart | 7-352 |
| Table 7-10 | List of Subprograms for Program BFIT | 7-451 |
| Table 7-11 | List of Variables for Program BFIT | 7-451 |
| Table 7-12 | List of Subprograms for Program ABTFIT | 7-455 |
| Table 7-13 | List of Variables for Program ABTFIT | 7-456 |
| Table 7-14 | List of Subprograms for Program LZERO | 7-461 |
| Table 7-15 | List of Variables for Program LZERO | 7-462 |
| Table 7-16 | List of Variables for Program BAYES | 7-471 |
| Table 7-17 | List of Variables for Subprogram RANDOM | 7-477 |
| Table 7-18 | List of Variables for Subprogram NORMGN | 7-479 |
| Table 7-19 | List of Variables for Subprogram BETAGN | 7-481 |
| Table 7-20 | List of Variables for Subprogram GAM | 7-482 |
| Table 7-21 | List of Variables for Subprogram WEIBGN | 7-483 |
| Table 7-22 | List of Variables for Subprogram PRYRV | 7-485 |
| Table 7-23 | List of Subprograms for Program NBSIN | 7-487 |
| Table 7-24 | List of Variables for Program NBSIN | 7-488 |
7.0 Structure and Listing of Programs
Section 7.1
High Cycle Fatigue Failure Programs

The program tree structures, list of subprograms, descriptions of the key variables, and the FORTRAN source listings for the two HCF analysis codes DCTHCF and HEXHCF are given here. The pertinent HCF methodology is given in Section 2.2.1. The overall description of the programs and the flowcharts are given in Section 5.1. The user's guides for running DCTHCF and HEXHCF are given in Section 6.1.

7.1.1 DCTHCF Program

7.1.1.1 Program Tree Structure
The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for DCTHCF using Uniform variation on the materials shape parameter $m$ is given in Figure 7-1, while the tree structure for the truncated Normal case is given in Figure 7-2. 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.

7.1.1.2 List of Subprograms
A list of subprograms and their purposes is given in Table 7-1. The section numbers where the subprograms are described by means of flowcharts are given next to the names.

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDREG</td>
<td>4.1.3.9</td>
<td>Adds the $m$ ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>4.1.3.15</td>
<td>Adds the $m$ ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.</td>
</tr>
<tr>
<td>BETAGN</td>
<td>4.4.5</td>
<td>Generates Beta($a$, $b$, $p$, $\theta$) random variates.</td>
</tr>
<tr>
<td>CALCS</td>
<td>5.1.2.4</td>
<td>Performs the stress component calculations in Equations 2-68, 2-69, 2-71 and 2-72 by using the loads, stress concentration factors, and geometric information.</td>
</tr>
<tr>
<td>CONCAV</td>
<td>4.1.3.10</td>
<td>Adjusts the upper bound of the posterior ranges on $m$ to be consistent with concavity constraints.</td>
</tr>
<tr>
<td>CONVRT</td>
<td>4.1.3.3</td>
<td>Transforms stress data to equivalent zero-mean stresses with stress ratio of $-1.0$.</td>
</tr>
</tbody>
</table>
Figure 7-1  Tree Structure for Program DCTHCF for the Uniform Variation in the Materials Shape Parameter $m$
Figure 7-2  Tree Structure for Program DCTHCF for the Truncated Normal Variation in the Materials Shape Parameter $m$

7 - 5
### Table 7-1 List of Subprograms For Program DCTHCF (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCTHCF</td>
<td>5.1.2.1</td>
<td>The main routine that controls the logical flow of the high cycle fatigue elbow welded duct program.</td>
</tr>
<tr>
<td>ELWELD</td>
<td>5.1.2.2</td>
<td>Controls the logical flow for the driver transformation and fatigue life calculations.</td>
</tr>
<tr>
<td>EXPCTD$^5$</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.</td>
</tr>
<tr>
<td>FINDM$^6$</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 for each life region implied by the constraint on the coefficient of variation of fatigue strength $C$ by using Equations 2-28 through 2-32.</td>
</tr>
<tr>
<td>FINDMN$^6$</td>
<td>4.1.5.2</td>
<td>Obtains the value of $m$ for each life region by sampling from the appropriate truncated Normal distribution on $m$.</td>
</tr>
<tr>
<td>FINDSB</td>
<td>4.1.5.7</td>
<td>Calculates the life region &quot;tie-points&quot; or stress values which correspond to the &quot;life boundaries&quot; conditional on the randomly selected $m$ for each region. Also calculates $K$, characterizing the specific material S/N data set, which is a function of $\beta_0$ and $k$.</td>
</tr>
<tr>
<td>FNDRNG$^7$</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>GAM</td>
<td>4.4.4</td>
<td>Generates Gamma($\alpha, 1$) random variates.</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.</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>4.1.3.7</td>
<td>Calculates $\sigma^2$, Equation 2-49, the extent of departures from the multiple heat median S/N curve warranted by the information available.</td>
</tr>
<tr>
<td>INFAGG$^8$</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 and 2-26.</td>
</tr>
</tbody>
</table>

---

7 - 6
Table 7-1  List of Subprograms For Program DCTHCF (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBETA</td>
<td>4.1.5.5</td>
<td>Calculates ( k ) and ( \beta_o ) from the sample mean and variance of ( Z ), where ( Z ) is a function of stress, life, the life region boundaries, and the ( m )'s by using Equation 2-42.</td>
</tr>
<tr>
<td>KOMO\textsuperscript{9}</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 ). Disabled for this application.</td>
</tr>
<tr>
<td>M2L1</td>
<td>5.1.2.3</td>
<td>Performs the calculations, Equations 2-73 through 2-80, necessary to find the stress at location 1, the exterior surface of the duct.</td>
</tr>
<tr>
<td>M2L2</td>
<td>5.1.2.3</td>
<td>Performs the calculations, Equations 2-73 through 2-80, necessary to find the stress at location 2, the interior surface of the duct.</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.</td>
</tr>
<tr>
<td>MUSIG\textsuperscript{10}</td>
<td>4.1.3.13</td>
<td>Calculates the posterior Normal distribution parameters: mean ( m_o ) and standard deviation ( \sigma_o ) for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NARBN1</td>
<td>5.1.2.5</td>
<td>Calculates the composite stress-time history by using Equations 2-73 and 2-74 and then calls RAINF1 to calculate the fatigue life.</td>
</tr>
<tr>
<td>NORMGN\textsuperscript{11}</td>
<td>4.4.3</td>
<td>Generates ( \text{Normal}(\mu, \sigma^2) ) random variates.</td>
</tr>
<tr>
<td>NORRNG\textsuperscript{7}</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\textsuperscript{12}</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>PGETSM</td>
<td>5.1.2.7</td>
<td>Calculates the equivalent mean stress from the maximum stress by using Equation 2-87.</td>
</tr>
<tr>
<td>PRYRV\textsuperscript{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>RAINF1\textsuperscript{14}</td>
<td>5.1.2.6</td>
<td>Performs rainflow cycle counting, Miner's rule damage accumulation, and calls GTLIFE to calculate the fatigue life.</td>
</tr>
<tr>
<td>RANDOM\textsuperscript{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 DCTHCD and RELATD; calls CONVRT to transform the stress data to a stress ratio of (-1.0); and echoes the data to DCTHCO 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 stress, life, the life region boundaries, and the ( m )'s by using Equation 2-42.</td>
</tr>
<tr>
<td>SORTM\textsuperscript{15}</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>
</tbody>
</table>
Table 7-1  List of Subprograms For Program DCTHCF (Cont’d)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2SU2</td>
<td>4.1.3.4</td>
<td>Calculates the residual variances from the $Y$ on $X$ and $X$ on regressions for each life region where $Y = \ln(\text{Endurance cycles})$ and $X = \ln(\text{Stress})$ by using Equations 2-20 and 2-21; 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</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>

---

1. No data regions to the right are discussed on Page 2-17.
2. The Beta distribution is discussed on Page 2-25.
3. Concavity constraints are discussed on Pages 2-13 through 2-14.
4. The stress transformation is discussed on Page 2-7.
5. The median S/N curve parameter estimation calculations are described on Pages 2-15 through 2-18.
6. Selection of the $\{m_j\}$ parameters is discussed on Page 2-15.
7. Combining information to obtain the posterior credibility ranges on $m$ is discussed on Page 2-13.
8. The information aggregation calculations are discussed on Pages 2-6 through 2-14.
9. Extension of the S/N curve to the left is discussed on Page 2-17.
10. Calculation of the truncated Normal distribution parameters is discussed on Page 2-14.
11. The Normal distribution is discussed on Page 2-23.
12. The parameter estimation calculations are discussed on Pages 2-15 through 2-18.
13. The Uniform distribution is discussed on Page 2-23.
14. Rainflow cycle counting is discussed on Page 2-51 and in Appendix 2.A.
15. The need for saving $m$'s is discussed on Page 2-15.
16. The S/N data transformation is discussed on Page 2-16.
7.1.1.3 Description of Variables

A list of variables used in the elbow welded duct HCF code, DCTHCF, is given in Table 7-2. 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, MAXLD, 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 to be used in the truncated Normal median S/N curve calculation. 1</td>
</tr>
<tr>
<td>ANGLE</td>
<td>RE</td>
<td>$\phi$ (rad) in Equation 2-68, the angle measured counterclockwise from Z-direction to the critical circumferential location.</td>
</tr>
<tr>
<td>AREA</td>
<td>RE</td>
<td>$A$ (in.$^2$) in Equation 2-68, the cross-sectional area of the duct wall.</td>
</tr>
<tr>
<td>B</td>
<td>RE</td>
<td>$\beta$ in Equation 2-79, the stress increase due to the torus effect.</td>
</tr>
<tr>
<td>BIGK(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the materials model location parameter $K$, Equation 2-12, where $A = K^m$.</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>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>BNRD</td>
<td>RE</td>
<td>$R_B$ (in.) in Equation 2-74, the elbow BeNd RaDius. (see RB in ELWELD)</td>
</tr>
</tbody>
</table>

Table 7-2 List of Variables for Program DCTHCF
(Footnotes are at the end of the table)
<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BZERO</td>
<td>RE</td>
<td>Estimate of Weibull distribution shape parameter $\beta_0$, \textit{Equation 2-11}, which characterizes the intrinsic variation of the S/N data set.</td>
</tr>
<tr>
<td>CCY</td>
<td>RE</td>
<td>$C_{cy}$ in \textit{Equation 2-77}, the out-of-plane circumferential stress carryover factor.</td>
</tr>
<tr>
<td>CCYA</td>
<td>RE</td>
<td>$C_{cy}$ Uniform distribution lower bound.</td>
</tr>
<tr>
<td>CCYB</td>
<td>RE</td>
<td>$C_{cy}$ Uniform distribution upper bound.</td>
</tr>
<tr>
<td>CCZ</td>
<td>RE</td>
<td>$C_{cz}$ in \textit{Equation 2-75}, the in-plane circumferential stress carryover factor.</td>
</tr>
<tr>
<td>CCZA</td>
<td>RE</td>
<td>$C_{cz}$ Uniform distribution lower bound.</td>
</tr>
<tr>
<td>CCZB</td>
<td>RE</td>
<td>$C_{cz}$ Uniform distribution upper bound.</td>
</tr>
<tr>
<td>CLY</td>
<td>RE</td>
<td>$C_{ly}$ in \textit{Equation 2-76}, the out-of-plane axial stress carryover factor.</td>
</tr>
<tr>
<td>CLYA</td>
<td>RE</td>
<td>$C_{ly}$ Uniform distribution lower bound.</td>
</tr>
<tr>
<td>CLYB</td>
<td>RE</td>
<td>$C_{ly}$ Uniform distribution upper bound.</td>
</tr>
<tr>
<td>CLZ</td>
<td>RE</td>
<td>$C_{lz}$ in \textit{Equation 2-74}, the in-plane axial stress carryover factor.</td>
</tr>
<tr>
<td>CLZA</td>
<td>RE</td>
<td>$C_{lz}$ Uniform distribution lower bound.</td>
</tr>
<tr>
<td>CLZB</td>
<td>RE</td>
<td>$C_{lz}$ Uniform distribution upper bound.</td>
</tr>
<tr>
<td>CULPRT</td>
<td>INT</td>
<td>Location about duct circumference responsible for failure. See variable \textit{LOCAT} for the possible locations.</td>
</tr>
<tr>
<td>DI</td>
<td>RE</td>
<td>$D_l$ (in.) the duct inner diameter at the weld, used to calculate $R_i$ in \textit{Equation 2-68}. (see IDWE in DCTHCF)</td>
</tr>
<tr>
<td>DSTR</td>
<td>RE</td>
<td>$\lambda_{DYNstr}$ in \textit{Equation 2-81}, the randomly selected dynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>DSTRA</td>
<td>RE</td>
<td>Dynamic stress analysis accuracy factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>DSTRB</td>
<td>RE</td>
<td>Dynamic stress analysis accuracy factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>ELWELD</td>
<td>RE</td>
<td>Real function that controls the logical flow for the driver transformation and fatigue life calculations of a duct at a weld near an elbow, and then returns the fatigue life (sec).</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>EM</td>
<td>RE</td>
<td>$E$ (psi) in Equation 2-70, Young's modulus of elasticity for the material. (see EMOD in DCTHCF)</td>
</tr>
<tr>
<td>EMOD</td>
<td>RE</td>
<td>$E$ (psi) in Equation 2-70, Young's modulus of elasticity for the material. (see EM in ELWELD)</td>
</tr>
<tr>
<td>FATLIF</td>
<td>RE</td>
<td>Value of FATigue LIFe calculated (sec).</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>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 2-73, used to find stress concentration due to weld eccentricity, $K_{OFF}$.</td>
</tr>
<tr>
<td>FTEST</td>
<td>LOG</td>
<td>File TEST. Used to test for the existence of a reference time history file before attempting to open it.</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>GAM</td>
<td>RE</td>
<td>$\lambda_{dam}$ in Equation 2-91, the randomly selected damage accumulation model accuracy factor. See Section 2.2.1.4 for a discussion of the damage calculations.</td>
</tr>
<tr>
<td>GAMA</td>
<td>RE</td>
<td>Damage accumulation model accuracy factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>Gamb</td>
<td>RE</td>
<td>Damage accumulation model accuracy factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>GCY</td>
<td>RE</td>
<td>$\gamma_{cy}$ in Equation 2-77, the out-of-plane circumferential ovality effect coefficient.</td>
</tr>
<tr>
<td>GCZ</td>
<td>RE</td>
<td>$\gamma_{cz}$ in Equation 2-75, the in-plane circumferential ovality effect coefficient.</td>
</tr>
<tr>
<td>GLY</td>
<td>RE</td>
<td>$\gamma_{y}$ in Equation 2-76, the out-of-plane axial ovality effect coefficient.</td>
</tr>
<tr>
<td>GLZ</td>
<td>RE</td>
<td>$\gamma_{z}$ in Equation 2-74, the in-plane axial ovality effect coefficient.</td>
</tr>
<tr>
<td>GNBI</td>
<td>RE</td>
<td>$\gamma_{nbi}$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>GNBO</td>
<td>RE</td>
<td>$\gamma_{nbo}$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>GTMI</td>
<td>RE</td>
<td>$\gamma_{tmi}$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>GTMO</td>
<td>RE</td>
<td>( \gamma_{\text{mo}} ) in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls inner DO loop.</td>
</tr>
<tr>
<td>IDWE</td>
<td>RE</td>
<td>( D_i ) (in.) the duct Inner Diameter at the WELD, used to calculate ( R_i ) in Equation 2-68. (see DI in ELWELD)</td>
</tr>
<tr>
<td>II</td>
<td>INT</td>
<td>Controls DO loop for narrow-band random and superimposed sinusoidal loads.</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. [2]</td>
</tr>
<tr>
<td>K</td>
<td>INT</td>
<td>Controls outer DO loop.</td>
</tr>
<tr>
<td>K(2, 2)</td>
<td>RE</td>
<td>2-D array containing the fatigue stress concentration factors required for the stress analysis. ( K(1,<em>) ) is ( K_{T1} ) in Equation 2-68 and ( K(2,</em>) ) is ( K_{T2} ) in Equation 2-69. ( K(1,1) ) is the outer diameter axial stress concentration factor, the value of ( KGOD \times KWOD ); ( K(1,2) ) is the inner diameter axial stress concentration factor, the value of ( KGID \times KWID ); ( K(2,1) ) is the outer diameter hoop stress concentration factor; and ( K(2,2) ) is the inner diameter hoop stress concentration factor. (see KT(2,2) in DCTHCF)</td>
</tr>
<tr>
<td>KGID</td>
<td>RE</td>
<td>Axial stress concentration factor due to geometry for the duct inner diameter used to calculate ( K_{T1} ) in Equation 2-68.</td>
</tr>
<tr>
<td>KGOD</td>
<td>RE</td>
<td>Randomly selected axial stress concentration factor due to geometry for the duct outer diameter used to calculate ( K_{T1} ) in Equation 2-68.</td>
</tr>
<tr>
<td>KGODA</td>
<td>RE</td>
<td>Outer diameter geometric axial stress concentration factor lower bound of Beta distribution.</td>
</tr>
<tr>
<td>KGODB</td>
<td>RE</td>
<td>Outer diameter geometric axial stress concentration factor upper bound of Beta distribution.</td>
</tr>
<tr>
<td>KGODR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for the outer diameter geometric axial stress concentration factor.</td>
</tr>
<tr>
<td>KGODR1</td>
<td>RE</td>
<td>( \rho ) Uniform distribution lower bound of Beta distribution of the outer diameter geometric axial stress concentration factor.</td>
</tr>
<tr>
<td>KGODR2</td>
<td>RE</td>
<td>( \rho ) Uniform distribution upper bound of Beta distribution of the outer diameter geometric axial stress concentration factor.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>KGODT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter ( \theta ) for the outer diameter geometric axial stress concentration factor.</td>
</tr>
<tr>
<td>KGODT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution of the outer diameter geometric axial stress concentration factor.</td>
</tr>
<tr>
<td>KGODT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution of the outer diameter geometric axial stress concentration factor.</td>
</tr>
<tr>
<td>KOFF</td>
<td>RE</td>
<td>( K_{OFF} ) in Equation 2-73, the stress concentration factor due to eccentricity of the weld.</td>
</tr>
<tr>
<td>KRATIO</td>
<td>RE</td>
<td>Ratio of MED ( K^* )/MED ( K ) in Equation 2-48. KRATIO is constant over life regions for the materials model.</td>
</tr>
<tr>
<td>KT(2, 2)</td>
<td>RE</td>
<td>2-D array containing the fatigue stress concentration factors required for the stress analysis. KT(1,<em>) is ( K_{T1} ) in Equation 2-68 and KT(2,</em>) is ( K_{T2} ) in Equation 2-69. KT(1,1) is the outer diameter axial stress concentration factor, the value of KGOD * KWOD; KT(1,2) is the inner diameter axial stress concentration factor, the value of KGID * KWID; KT(2,1) is the outer diameter hoop stress concentration factor; and KT(2,2) is the inner diameter hoop stress concentration factor. (see K(2,2) in ELWELD)</td>
</tr>
<tr>
<td>KT1</td>
<td>RE</td>
<td>( K_{T1} ) in Equation 2-68, the stress concentration factor for the axial stress.</td>
</tr>
<tr>
<td>KT2</td>
<td>RE</td>
<td>( K_{T2} ) in Equation 2-69, the stress concentration factor for the hoop stress.</td>
</tr>
<tr>
<td>KWID</td>
<td>RE</td>
<td>Randomly selected axial stress concentration factor due to the weld for the duct inner diameter used to calculate ( K_{T1} ) in Equation 2-68.</td>
</tr>
<tr>
<td>KWIDA</td>
<td>RE</td>
<td>Inner diameter weld axial stress concentration factor lower bound of Beta distribution.</td>
</tr>
<tr>
<td>KWIDB</td>
<td>RE</td>
<td>Inner diameter weld axial stress concentration factor upper bound of Beta distribution.</td>
</tr>
<tr>
<td>KWIDR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>KWIDR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWIDR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWIDT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ for the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWIDT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWIDT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWOD</td>
<td>RE</td>
<td>Randomly selected axial stress concentration factor due to the weld for the duct outer diameter used to calculate $K_{T1}$ in Equation 2-68.</td>
</tr>
<tr>
<td>KWODA</td>
<td>RE</td>
<td>Outer diameter weld axial stress concentration factor lower bound of Beta distribution.</td>
</tr>
<tr>
<td>KWODB</td>
<td>RE</td>
<td>Outer diameter weld axial stress concentration factor upper bound of Beta distribution.</td>
</tr>
<tr>
<td>KWODR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODR1</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>KWODT2</td>
<td>RE</td>
<td>( \theta ) Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.</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>L</td>
<td>RE</td>
<td>( \lambda ) in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>LAMN</td>
<td>RE</td>
<td>( \lambda_{D\text{RANDOM}} ) in Equation 2-81, the randomly selected load scale factor for the narrow-band random loads. See Section 2.1.3.2 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>LAMNMU</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>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 2-81, the randomly selected load scale factor for the superimposed sinusoidal loads. See Section 2.1.3.2 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>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</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>LAMST</td>
<td>RE</td>
<td>( \lambda ) in Equation 2-81, the randomly selected load scale factor for the static loads.</td>
</tr>
<tr>
<td>LAMSTA</td>
<td>RE</td>
<td>Uniform distribution lower bound for the static load scale factor.</td>
</tr>
<tr>
<td>LAMSTB</td>
<td>RE</td>
<td>Uniform distribution upper bound for the static load scale factor.</td>
</tr>
<tr>
<td>LAMW</td>
<td>RE</td>
<td>( \lambda ) in Equation 2-73, 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.</td>
</tr>
<tr>
<td>LAMWB</td>
<td>RE</td>
<td>Uniform distribution upper bound.</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 DCTHCF. The lives are sorted values for the left-hand tail simulated failure distribution.</td>
</tr>
<tr>
<td>LIMPRI</td>
<td>RE</td>
<td>( p_l ) (psi) in Equation 2-68, the LIMIT or internal PRESSure. (see PSUBI in ELWELD)</td>
</tr>
<tr>
<td>LNA(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of ( \ln(A) = \ln(BIGK) \times MM ) for each life region of the S/N curve.</td>
</tr>
<tr>
<td>LNZ</td>
<td>RE</td>
<td>( \ln(Z) ) in Equation 2-48, the Normal(0, PVAR) random variable for the materials process variation aspect of the materials model.</td>
</tr>
<tr>
<td>LOCAT</td>
<td>INT</td>
<td>Critical location of interest on the duct 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>LPHIM(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of $\ln(PHI) \times MM$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>M(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 2-68, the moment load components about the y axis; and $M(2,</em>)$ is $M_z$ (in.-lbs) in Equation 2-68, 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 10.2.</td>
</tr>
<tr>
<td>MAXDAT</td>
<td>INT</td>
<td>Maximum number of points per data set per region allowed for S/N curve. The maximum number of data points per set allowed is 50.</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 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 24,000.</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 median S/N curve for the truncated Normal distribution case.</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>MI</td>
<td>RE</td>
<td>$I$ (in.$^4$) in Equation 2-68, the cross-sectional Moment of Inertia.</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>MIIB</td>
<td>RE</td>
<td>$I$ (in.$^4$) in Equation 2-68, the cross-sectional moment of inertia calculated by using the wall thickness at the inner bend.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>MIOB</td>
<td>RE</td>
<td>( I ) (in.(^4)) in Equation 2-68, the cross-sectional moment of inertia calculated by using the wall thickness at the outer bend.</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 ) and ( LAMS ) or ( LAMN ), as appropriate, according to variable ( \text{TYPE}() ). ( \text{MLAM}(1,<em>) ) is ( M_y ) (in.-lbs) in Equation 2-68, the moment load components about the ( y ) axis; and ( \text{MLAM}(2,</em>) ) is ( M_z ) (in.-lbs) in Equation 2-68, the moment load components about the ( z ) axis.</td>
</tr>
<tr>
<td>MM(0[MAXREG])</td>
<td>RE</td>
<td>( m_j ) in Equation 2-12, 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>MNWT</td>
<td>RE</td>
<td>( t_w ) (in.) the duct Minimun Wall Thickness at the weld outer diameter. (see TOB in ELWELD)</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.(^4)</td>
</tr>
<tr>
<td>MSLAM(2)</td>
<td>RE</td>
<td>1-D array containing static moment load components scaled by ( SSTR ) and ( \text{LAMST} ). ( \text{MSLAM}(1) ) is ( M_y ) (in.-lbs) in Equation 2-68, the moment load component about the ( y ) axis; and ( \text{MSLAM}(2) ) is ( M_z ) (in.-lbs) in Equation 2-68, the moment load component about the ( z ) axis.</td>
</tr>
<tr>
<td>MSTAT(2)</td>
<td>RE</td>
<td>1-D array containing the static moment load components. ( \text{MSTAT}(1) ) is ( M_y ) (in.-lbs) in Equation 2-68, the moment load component about the ( y ) axis; and ( \text{MSTAT}(2) ) is ( M_z ) (in.-lbs) in Equation 2-68, the moment load component about the ( z ) axis.</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_{i,i+1} ) in Equation 2-35, 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>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NEWLIF</td>
<td>RE</td>
<td>Fatigue life value (sec) returned from call to function ELWELD.</td>
</tr>
<tr>
<td>NF(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values from the array RAWNF( ) for the specific material S/N data set partitioned into life regions.</td>
</tr>
<tr>
<td>NHYPER</td>
<td>INT</td>
<td>The outer loop size.</td>
</tr>
<tr>
<td>NLIFE</td>
<td>INT</td>
<td>The inner loop size.</td>
</tr>
<tr>
<td>NLIFFET</td>
<td>INT</td>
<td>Total number of lives calculated by program DCTHCF. Value of NHYPER * NLIFFET.</td>
</tr>
<tr>
<td>NLOAD</td>
<td>INT</td>
<td>NLOAD in Equation 2-81, the number of dynamic or time-varying loads.</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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>a median calculation or that the Uniform distribution case is being used; while a value of 1 indicates that the user desires the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>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 Equations 2-74 through 2-77, the Poisson's ratio for the duct material.</td>
</tr>
<tr>
<td>NUMREG</td>
<td>INT</td>
<td>$R$ in Equation 2-11, the number of life regions of interest in the S/N curve.</td>
</tr>
<tr>
<td>OVAL</td>
<td>RE</td>
<td>$\lambda_{oval}$ in Equations 2-74 through 2-77, the randomly selected ovality effect analysis accuracy factor.</td>
</tr>
<tr>
<td>OVALA</td>
<td>RE</td>
<td>Uniform distribution lower bound for the ovality effect analysis accuracy factor.</td>
</tr>
<tr>
<td>OVALB</td>
<td>RE</td>
<td>Uniform distribution upper bound for the ovality effect analysis accuracy factor.</td>
</tr>
<tr>
<td>P(MAXLD)</td>
<td>RE</td>
<td>1-D array containing $P$ (lbs) in Equation 2-68, the dynamic or time-varying axial load components.</td>
</tr>
<tr>
<td>PERIOD</td>
<td>RE</td>
<td>$T$ (sec) in Equation 2-91, the length of time in seconds of the reference 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>PHI</td>
<td>RE</td>
<td>$\varphi$ in Equation 2-11, the materials intrinsic variation, or scatter, given by a Weibull($\beta_o$, $\eta_o$) random variate.</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 2-68, the dynamic or time-varying axial load components scaled by DSTR and LAMN or LAMS, as appropriate, according to variable TYPE().</td>
</tr>
<tr>
<td>PSI</td>
<td>RE</td>
<td>$\psi$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>PSIG</td>
<td>RE</td>
<td>$\sigma$ in Equation 2-48, the value of SQRT(PVAR).</td>
</tr>
<tr>
<td>PSLAM</td>
<td>RE</td>
<td>$P$ (lbs) in Equation 2-68, the static axial load component scaled by SSTR and LAMST.</td>
</tr>
<tr>
<td>PSTAT</td>
<td>RE</td>
<td>$P$ (lbs) in Equation 2-68, the static axial load component.</td>
</tr>
<tr>
<td>PSUBI</td>
<td>RE</td>
<td>$p_i$ (psi) in Equation 2-68, the limit or internal pressure. (see LIMPR in DCTHCF)</td>
</tr>
<tr>
<td>PVAR</td>
<td>RE</td>
<td>$\sigma^2$ in Equation 2-48, characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.</td>
</tr>
<tr>
<td>QO</td>
<td>RE</td>
<td>$Q_o$ in Equation 2-78, the decay factor for the ovality effect.</td>
</tr>
<tr>
<td>QT</td>
<td>RE</td>
<td>$Q_T$ in Equation 2-80, the decay factor for the torus effect.</td>
</tr>
<tr>
<td>R</td>
<td>RE</td>
<td>$R$ in Equation 2-68, the radius where the stress is to be found.</td>
</tr>
<tr>
<td>RAINF1</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.</td>
</tr>
<tr>
<td>RB</td>
<td>RE</td>
<td>$R_B$ (in.) in Equation 2-74, the elbow bend radius. (see BNRD in DCTHCF)</td>
</tr>
<tr>
<td>RI</td>
<td>RE</td>
<td>$R_I$ (in.) in Equation 2-68, the duct inner radius.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>RM</td>
<td>RE</td>
<td>$R_m$ (in.) in Equations 2-74 through 2-77, the mean duct radius.</td>
</tr>
<tr>
<td>RO</td>
<td>RE</td>
<td>$R_o$ (in.) in Equation 2-68, the duct outer radius.</td>
</tr>
<tr>
<td>ROIB</td>
<td>RE</td>
<td>$R_o$ (in.) in Equation 2-68, the duct outer radius calculated by using the wall thickness at the inner bend.</td>
</tr>
<tr>
<td>ROOB</td>
<td>RE</td>
<td>$R_o$ (in.) in Equation 2-68, the duct outer radius calculated by using the wall thickness at the outer bend.</td>
</tr>
<tr>
<td>ROT</td>
<td>RE</td>
<td>$R$ Over $T$, the value of the ratio $R/L$.</td>
</tr>
<tr>
<td>ROVERI</td>
<td>RE</td>
<td>$R$ OVER $I$, the value of the ratio $R/l$.</td>
</tr>
<tr>
<td>RT(10)</td>
<td>RE</td>
<td>1-D array containing values of $R/t$ used in conjunction with $F_k$, Equation 2-73, to find stress concentration due to weld eccentricity, $K_{OFF}$.</td>
</tr>
<tr>
<td>S(4, MAXM)</td>
<td>RE</td>
<td>2-D array containing the total component stress-time histories $\sigma_k(t)$ (psi), Equation 2-82, resulting from the combination of static, narrow-band random, and sinusoidal loads. $S(1,<em>)$ is the axial stress-time history $\sigma_1(t)$; $S(2,</em>)$ is the hoop stress-time history $\sigma_2(t)$; $S(3,<em>)$ is the radial stress-time history $\sigma_3(t)$; and $S(4,</em>)$ is the shear stress-time history $\sigma_4(t)$.</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>SEFF(MAXM)</td>
<td>RE</td>
<td>1-D array containing the EFFective or uni-axial stress-time history $\sigma(t)$ (psi), Equation 2-84, resulting from the combination of static, narrow-band random, and sinusoidal loads for all four stress components.</td>
</tr>
<tr>
<td>SIG(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution standard deviation of the materials model shape parameter $m$, for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>SSTR</td>
<td>RE</td>
<td>$\lambda ST_{str}$ in Equation 2-81, the randomly selected static stress analysis accuracy factor.</td>
</tr>
<tr>
<td>SSTRA</td>
<td>RE</td>
<td>Static stress analysis accuracy factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SSTRB</td>
<td>RE</td>
<td>Static stress analysis accuracy factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>STATIC(4)</td>
<td>RE</td>
<td>1-D array containing values of the static stresses ( \sigma_{Stk} ) (psi), Equation 2-82. STATIC(1) is the axial stress ( \sigma_{St1} ); STATIC(2) is the hoop stress ( \sigma_{St2} ); STATIC(3) is the radial stress ( \sigma_{St3} ); and STATIC(4) is the shear stress ( \sigma_{St4} ).</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>STRAMP(4, MAXLD)</td>
<td>RE</td>
<td>2-D array containing values of the amplitudes of the dynamic or time-varying stresses ( \sigma_{Dik} ) (psi), Equation 2-82. STRAMP(1,1) is the amplitude of the ( i^{\text{th}} ) axial stress; STRAMP(2,1) is ( \sigma_{D2i} ), the amplitude of the ( i^{\text{th}} ) hoop stress; STRAMP(3,1) is ( \sigma_{D3i} ), the amplitude of the ( i^{\text{th}} ) radial stress; and STRAMP(4,1) is ( \sigma_{D4i} ), the amplitude of the ( i^{\text{th}} ) shear stress.</td>
</tr>
<tr>
<td>STRHIS(MAXLD, MAXM)</td>
<td>RE</td>
<td>2-D array containing ( \sigma(t) ), Equation 2-82, the reference time histories for the dynamic or time-varying load components.</td>
</tr>
<tr>
<td>SZERO</td>
<td>RE</td>
<td>Stress tensile test point, ( S_o ) (psi).</td>
</tr>
<tr>
<td>T(MAXLD)</td>
<td>RE</td>
<td>1-D array containing ( M_x ) (in.-lbs) in Equation 2-72, the dynamic or time-varying torsional load components.</td>
</tr>
<tr>
<td>TEST</td>
<td>RE</td>
<td>Uniform(0, 1) random variate used to determine Beta distribution for ( W_{OFF} ).</td>
</tr>
<tr>
<td>TIB</td>
<td>RE</td>
<td>( t_{w2} ) (in.) the wall thickness at the bend inner diameter at the weld. (see WTID in DCTHCF)</td>
</tr>
<tr>
<td>TLAM(MAXLD)</td>
<td>RE</td>
<td>1-D array containing ( M_x ) (in.-lbs) in Equation 2-72, the dynamic or time-varying torsional load components scaled by DSTR and LAMN or LAMS, as appropriate according to variable TYPE().</td>
</tr>
<tr>
<td>TM</td>
<td>RE</td>
<td>( t_m ) (in.) in Equations 2-74 through 2-77, the mean wall thickness at the weld.</td>
</tr>
<tr>
<td>TOB</td>
<td>RE</td>
<td>( t_{w1} ) (in.) the duct minimum wall thickness at the weld outer diameter. (see MNWT in DCTHCF)</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</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 &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 $S_o$.</td>
</tr>
<tr>
<td>TRUNC</td>
<td>RE</td>
<td>Value used to filter out noise in the composite stress-time history during rainflow cycle counting. See Section 2.2.1.4 for a discussion of rainflow cycle counting.</td>
</tr>
<tr>
<td>TSLAM</td>
<td>RE</td>
<td>$M_x$ (in.-lbs) in Equation 2-72, the static torsional load component scaled by SSTR and LAMST.</td>
</tr>
<tr>
<td>TSTAT</td>
<td>RE</td>
<td>$M_x$ (in.-lbs) in Equation 2-72, the static torsional load component.</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(*) = 1, use the narrow-band random load scale factor; and TYPE(+) = 2, use the superimposed sinusoidal load scale factor.</td>
</tr>
<tr>
<td>V(2, MAXLD)</td>
<td>RE</td>
<td>2-D array containing the dynamic or time-varying shear load components. $V(1,<em>)$ is $V_y$ (lbs) in Equation 2-72, the shear load components along the y axis; and $V(2,</em>)$ is $V_z$ (lbs) in Equation 2-72, the shear load components along the z axis.</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; 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>VLAM(2, MAXLD)</td>
<td>RE</td>
<td>2-D array containing the dynamic or time-varying shear load components scaled by DSTR and LAMN or LAMS, as appropriate, according to variable TYPE( ). $VLAM(1,<em>)$ is $V_y$ (lbs) in Equation 2-72, the shear load components along the y axis; and $VLAM(2,</em>)$ is $V_z$ (lbs) in Equation 2-72, the shear load components along the z 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>VSLAM(2)</td>
<td>RE</td>
<td>1-D array containing the static shear load components scaled by SSTR and LAMST. VSLAM(1) is $V_y$ (lbs) in <em>Equation 2-72</em>, the shear load component along the y axis; and VSLAM(2) is $V_z$ (lbs) in <em>Equation 2-72</em>, the shear load component along the z axis.</td>
</tr>
<tr>
<td>VSTAT(2)</td>
<td>RE</td>
<td>1-D array containing the static shear load components. VSTAT(1) is $V_y$ (lbs) in <em>Equation 2-72</em>, the shear load component along the y axis; and VSTAT(2) is $V_z$ (lbs) in <em>Equation 2-72</em>, the shear load component along the z axis.</td>
</tr>
<tr>
<td>WD</td>
<td>RE</td>
<td>$W_D$ (in.) in <em>Equation 2-78</em>, the Weld Distance from elbow tangency line. (see WEDS in DCTHCF)</td>
</tr>
<tr>
<td>WEDS</td>
<td>RE</td>
<td>$W_D$ (in.) in <em>Equation 2-78</em>, the Weld Distance from elbow tangency line. (see WD in ELWELD)</td>
</tr>
<tr>
<td>WEOF</td>
<td>RE</td>
<td>$W_{OFF}$ in <em>Equation 2-73</em>, the randomly selected Weld Offset (%). (see WOFF in ELWELD)</td>
</tr>
<tr>
<td>WEOFA</td>
<td>RE</td>
<td>$W_{OFF}$ lower bound of Beta distribution 1.</td>
</tr>
<tr>
<td>WEOFB</td>
<td>RE</td>
<td>$W_{OFF}$ upper bound of Beta distribution 1.</td>
</tr>
<tr>
<td>WEOFC</td>
<td>RE</td>
<td>$W_{OFF}$ lower bound of Beta distribution 2.</td>
</tr>
<tr>
<td>WEOFD</td>
<td>RE</td>
<td>$W_{OFF}$ upper bound of Beta distribution 2.</td>
</tr>
<tr>
<td>WEOFE</td>
<td>RE</td>
<td>Decimal equivalent percentage weight occurring in Beta distribution 1 of the weld offset $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFHI</td>
<td>RE</td>
<td>Upper bound of the randomly selected Beta distribution for the weld offset $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFLO</td>
<td>RE</td>
<td>Lower bound of the randomly selected Beta distribution for the weld offset $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for the weld offset $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFR1</td>
<td>RE</td>
<td>$\rho$ Uniform distribution lower bound of Beta distribution 1 of $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFR2</td>
<td>RE</td>
<td>$\rho$ Uniform distribution upper bound of Beta distribution 1 of $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFR3</td>
<td>RE</td>
<td>$\rho$ Uniform distribution lower bound of Beta distribution 2 of $W_{OFF}$.</td>
</tr>
</tbody>
</table>
### Table 7-2  List of Variables for Program DCTHCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEOFR4</td>
<td>RE</td>
<td>$\rho$ Uniform distribution upper bound of Beta distribution 2 of $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ for the weld offset $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFT1</td>
<td>RE</td>
<td>$\theta$ Uniform distribution lower bound of Beta distribution 1 of $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFT2</td>
<td>RE</td>
<td>$\theta$ Uniform distribution upper bound of Beta distribution 1 of $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFT3</td>
<td>RE</td>
<td>$\theta$ Uniform distribution lower bound of Beta distribution 2 of $W_{OFF}$.</td>
</tr>
<tr>
<td>WEOFT4</td>
<td>RE</td>
<td>$\theta$ Uniform distribution upper bound of Beta distribution 2 of $W_{OFF}$.</td>
</tr>
<tr>
<td>WOFF</td>
<td>RE</td>
<td>$W_{OFF}$ in Equation 2-73, the randomly selected Weld Offset (%). (see WEOF in DCTHCF)</td>
</tr>
<tr>
<td>WTID</td>
<td>RE</td>
<td>$tw_{2}$ (in.) the Wall Thickness at the bend Inner Diameter at the weld. (see TIB in ELWELD)</td>
</tr>
<tr>
<td>X1</td>
<td>RE</td>
<td>$X_1$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>X2</td>
<td>RE</td>
<td>$X_2$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>X3</td>
<td>RE</td>
<td>$X_3$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>X4</td>
<td>RE</td>
<td>$X_4$ in Equations 2-74 through 2-77.</td>
</tr>
<tr>
<td>Z</td>
<td>RE</td>
<td>$Z$ in Equation 2-48, 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.</td>
</tr>
</tbody>
</table>

---

1. The need for saving $m$'s is discussed on 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 on Page 2-15.
4. See Section 2.1.2.3 for a discussion on process variation in materials.
5. $m_*$ of the posterior density of $m$ is discussed on Page 2-14.
6. The posterior credibility ranges $\pi(m)$ are discussed on Page 2-13.
7. $\sigma_*$ of the posterior density of $m$ is discussed on Page 2-14.
8. Extension of the S/N curve to the left using the tensile point is discussed on Page 2-17. Disabled for this application.
## 7.1.1.4 Program DCTHCF Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program DCTHCF Listing Temporal Order, Uniform Distribution</td>
<td>7-28</td>
</tr>
<tr>
<td>Program DCTHCF Listing Temporal Order, Truncated Normal Distribution</td>
<td>7-30</td>
</tr>
<tr>
<td>DCTHCF</td>
<td>7-32</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-44</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-45</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-46</td>
</tr>
<tr>
<td>GAM</td>
<td>7-46</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-47</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-52</td>
</tr>
<tr>
<td>INIT</td>
<td>7-52</td>
</tr>
<tr>
<td>RCE</td>
<td>7-54</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-61</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-62</td>
</tr>
<tr>
<td>INTRV</td>
<td>7-64</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-68</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-69</td>
</tr>
<tr>
<td>FINDRNG</td>
<td>7-71</td>
</tr>
<tr>
<td>ADDRREG</td>
<td>7-74</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-76</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-77</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-78</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-80</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-82</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-84</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-86</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-88</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-90</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-92</td>
</tr>
<tr>
<td>TRANSPM</td>
<td>7-93</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-94</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-95</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-96</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-97</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-98</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-99</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-100</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-101</td>
</tr>
<tr>
<td>ELWELD</td>
<td>7-102</td>
</tr>
<tr>
<td>M2L1</td>
<td>7-105</td>
</tr>
<tr>
<td>M2L2</td>
<td>7-108</td>
</tr>
<tr>
<td>CALCS</td>
<td>7-112</td>
</tr>
<tr>
<td>NARBN1</td>
<td>7-114</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
</tr>
<tr>
<td>RAINF1</td>
<td>7-116</td>
</tr>
<tr>
<td>PGETSM</td>
<td>7-120</td>
</tr>
</tbody>
</table>

DCTHCF Version 3.4
## Program DCTHCF Listing Temporal Order, Uniform Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCTHCF</td>
<td>7-32</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-47</td>
</tr>
<tr>
<td>INIT</td>
<td>7-52</td>
</tr>
<tr>
<td>RCE</td>
<td>7-54</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-61</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-62</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-68</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-64</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-69</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-71</td>
</tr>
<tr>
<td>ADDREG</td>
<td>7-74</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-76</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-77</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-78</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-93</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-94</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-95</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-96</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-97</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-99</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-45</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-86</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-88</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-93</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-94</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-95</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-96</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-97</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-92</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-46</td>
</tr>
<tr>
<td>GAM</td>
<td>7-46</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-92</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-45</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-98</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>ELWELD</td>
<td>7-102</td>
</tr>
<tr>
<td>M2L1</td>
<td>7-105</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>M2L2</td>
<td>7-108</td>
</tr>
<tr>
<td>CALCS</td>
<td>7-112</td>
</tr>
<tr>
<td>NARBN1</td>
<td>7-114</td>
</tr>
<tr>
<td>RAINF1</td>
<td>7-116</td>
</tr>
<tr>
<td>PGETSM</td>
<td>7-120</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-100</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-44</td>
</tr>
</tbody>
</table>
Program DCTHCF Listing Temporal Order, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCTHCF</td>
<td>7-32</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-47</td>
</tr>
<tr>
<td>INIT</td>
<td>7-52</td>
</tr>
<tr>
<td>RCE</td>
<td>7-54</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-61</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-62</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-68</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-80</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-69</td>
</tr>
<tr>
<td>NORMING</td>
<td>7-82</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-84</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-76</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-45</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-86</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-90</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-92</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-93</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-94</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-95</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-96</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-97</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-92</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-89</td>
</tr>
<tr>
<td>GAM</td>
<td>7-46</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-92</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-89</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-45</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-98</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-89</td>
</tr>
<tr>
<td>ELWELD</td>
<td>7-102</td>
</tr>
<tr>
<td>M2L1</td>
<td>7-105</td>
</tr>
<tr>
<td>M2L2</td>
<td>7-108</td>
</tr>
<tr>
<td>CALCS</td>
<td>7-112</td>
</tr>
<tr>
<td>NARB1</td>
<td>7-114</td>
</tr>
<tr>
<td>RAINF1</td>
<td>7-116</td>
</tr>
<tr>
<td>PGETSM</td>
<td>7-120</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-100</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-44</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-101</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-78</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-93</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-94</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-95</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-96</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-97</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-99</td>
</tr>
</tbody>
</table>
PROGRAM DCTHCF CONTROLS THE FLOW OF LOGIC OF THE HIGH CYCLE
FATIGUE ELBOW WELDED DUCT PROBLEM

PROGRAMMER: L. NEWLIN
DATE: 4SEP91
VERSION: 3.4 (MATCHR V8.5, PIPE V8.3, INSORT V2.1)

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

C IMPLICIT NONE

INTEGER MAXBLF, MAXDAT, MAXLD, MAXLIF, MAXM, MAXMM, MAXREG
REAL PI
PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLD = 16, MAXLIF = 10000,
 & MAXM = 24000, MAXMM = 20001, MAXREG = 3,
 & PI = 3.1415926536)

COMMON IOUT
INTEGER BLFPOS(MAXBLF), CULPRT, FILNUM(MAXLD), I, II, IOUT, J,
 & K, L, LOCAT, MCOUNT, MD, MPROC, NBLife, NHyp, NLIFET, NLOAD,
 & NMAX, NUMREG, TYPE(MAXLD), VARY, ZROREG

DOUBLE PRECISION RAND

REAL ALLM(MAXMM, MAXREG), ANGLE, BIGK(0:MAXREG), BIGK1,
 & BLFPER(MAXBLF), BNRD, BZERO, CCY, CCYA, CCYB, CCZ, CCZA,
 & CCZB, CLY, CLYA, CLYB, CLZ, CLZA, CLZB, DSTR, DSTRA,
 & DSTRB, ELWELD, EMOD, FIFTY, FK(10), FTU, FTY, GAM, GAMA,
 & GAMB, IDWE, KGID, KGOD, KGODA, KGODB, KGODR, KGODRI,
 & KGOD2, KGODT, KGODT1, KGODT2, KRATIO, KT(2,2), KWID,
 & KWID2, KWIF, KWIDR, KWIDR1, KWIDR2, KWIDT, KWIDT1,
 & KWIDT2, KNYD, KWODA, KWODB, KWODT, KWODT1, KWODT2,
 & KWODT2, KWODT3, KWODT4, LAMN, LAMNA, LAMNB, LAMNC, LAMND,
 & LAMNK, LAMNUM, LAMSG, LAMS, LAMSA, LAMSB, LAMSC, LAMSD,
 & LAMUE, LAMSU, LANS, LAMST, LAMSTB, LAMWB, LIFE(MAXLIF), LIMPR, LNA(0:MAXREG), LNZ,
 & LPHIM(0:MAXREG)

REAL M(2, MAXLD), MEDM(MAXREG), MLAM(2, MAXLD),
 & MM(0:MAXREG), MWE, MSLE(2), MSTAT(2), MU(MAXREG),
 & NBND(0:MAXREG), NEWLIF, NF(MAXDAT, MAXREG), OVAL, OVALA,
 & OVALB, P(MAXLD), PERIOD, PHI, PLAM(MAXLD), PSIG, PSLAM,
 & PSTAT, PVAR, RANGE(2, MAXREG), RT(10), SBYD(0:MAXREG),
 & SIG(MAXREG), SSTR, SSTRA, SSTRA, SSTRB, STR(MAXDAT, MAXREG),
 & STRHIS(MAXLD, MAXM), SZERO, T(MAXLD), TEST, TLAM(MAXLD),
 & TRSBND(0:MAXREG), TRUNC, TLSAM, TSTAT(2), V(2, MAXLD),
 & VLAM(2, MAXLD), VSAM(2, MAXLD), VSTAT(2), WEDS, WEOF, WEOF4,
 & WEOF, WEOF4, WEOF5, WEOF7, WEOF1, WEOF6, WEOF8, WEOF9,
 & WEOF10, WEOF11, WEOF12, WEOF13, WEOF14, WEOF15, WEOF16,
 & WEFR, WEOF2, WEOF3, WEOF4, WEOF5, WEOF6, WEOF7, WEOF8,
 & WEOF9, WEOF10, WEOF11, WEOF12, WEOF13, WEOF14, WEOF15, WEOF16

CHARACTER*6 LDNAME(MAXLD)

LOGICAL FTEST

DATA (FILNUM(I), I = 1, MAXLD)/
 & 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
 & 21, 22, 23, 24, 25, 26 /
C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

OPEN (1, FILE = 'DCTHCD', STATUS = 'OLD')
OPEN (3, FILE = 'DCTHCO', STATUS = 'NEW')
OPEN (7, FILE = 'DUMP', STATUS = 'NEW')
OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')

C INITIALIZE LOAD ARRAYS

PSTAT = 0.0
TSTAT = 0.0
MSTAT(1) = 0.0
MSTAT(2) = 0.0
VSTAT(1) = 0.0
VSTAT(2) = 0.0
PSLAM = 0.0
TSLAM = 0.0
MSLAM(1) = 0.0
MSLAM(2) = 0.0
VSLAM(1) = 0.0
VSLAM(2) = 0.0

DO 10 I = 1, MAXLD
   P(I) = 0.0
   PLAM(I) = 0.0
   T(I) = 0.0
   TLAM(I) = 0.0
   M(1,I) = 0.0
   M(2,I) = 0.0
   MLAM(1,I) = 0.0
   MLAM(2,I) = 0.0
   V(1,I) = 0.0
   V(2,I) = 0.0
   VLAM(1,I) = 0.0
   VLAM(2,I) = 0.0
10 CONTINUE

READ(1,*) RAND
WRITE(8,*) ' RANDOM NUMBER SEED = ', RAND
READ(1,*) IOUT
WRITE(8,*) ' IOUT (MATCHR = 10, DCTHCF = 15, ELWELD = 25) = ', IOUT
READ(1,*) NLIFE
WRITE(8,*) ' INNER LOOP SIZE = ', NLIFE
READ(1,*) NHYPER
WRITE(8,*) ' OUTER LOOP SIZE = ', NHYPER
READ(1,*) VARY
WRITE(8,*) ' TYPE OF S/N VARIATION DESIRED = ', VARY
READ(1,*) NMED
WRITE(8,*) ' NORMAL MEDIAN CURVE (0 - NO, 1 - YES) = ', NMED
READ(1,*) MPROC
WRITE(8,*) ' MATERIALS PROCESS VARIATION DESIRED' 
   '(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(1,*) NBLIFE
DO 15 J = 1, NBLIFE
    READ(*,*) BLFPER(J)
   15 CONTINUE

C ** READ DATA FROM DCTHCD

READ(1,*), WEOFA, WEOFB, WEOR1, WEOR2, WEOF1, WEOF2,
   &    WEOF3, WEOR4, WEOF5, WEOF6, WEOF7,
   &    WEOF8, WEOF9, WEOF10, WEOF11, WEOF12,
   &    WEOF13, WEOF14, WEOF15, WEOF16,
   &    KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2,
   &    KWIDIA, KWIDIB, KWIDIR1, KWIDIR2, KWIDIT1, KWIDIT2,
   &    KGODA, KGODB, KGODR1, KGODR2, KGODT1, KGODT2,
   &    LAMNA, LAMNB, LAMNC, LAMND,
   &    LAMSA, LAMSB, LAMSC, LAMSD,
   &    LAMSTA, LAMSTB, DSTRA, DSTRB, SSRA, SSTRB,
   &    CLZA, CLZB, CLYA, CLYB,
   &    CZA, CZA, CZA, CZA, CZA, CZA,
   &    OVALA, OVALB, LAMWA, LAMWB, GAMA, GAMB

READ(1,*), NLOAD, PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1), VSTAT(2)
DO 20 I = 1, NLOAD
   READ(*,*) LDNAME(I), TYPE(I), P(I), T(I), M(I,1), M(2,I),
   &    V(I,1), V(I,2)
   IF ((TYPE(I) .LT. 1) .OR. (TYPE(I) .GT. 2)) THEN
      WRITE(8,*), 'ERROR: LOAD INCORRECTLY TYPED'
      CALL TRMNT
   ENDIF
   20 CONTINUE

READ(1,*), KGID, KT(2,1), KT(2,2), LIMPR, BNRD, WEDS, IDWE,
   &    MNWT, WITD, EMOD, LOCAT, ANGLE, PERIOD, TRUNC, NRAN
READ(1,*), (FK(I), RT(I), I = 1, 10)

C ** ECHO DATA TO DCTHCO

WRITE(3,900)
WRITE(3,901) WEOFA, WEOFB, WEOR1, WEOR2, WEOF1, WEOF2,
   &    WEOF3, WEOR4, WEOF5, WEOF6, WEOF7,
   &    WEOF8, WEOF9, WEOF10, WEOF11, WEOF12,
   &    WEOF13, WEOF14, WEOF15, WEOF16,
   &    KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2,
   &    KWIDIA, KWIDIB, KWIDIR1, KWIDIR2, KWIDIT1, KWIDIT2,
   &    KGODA, KGODB, KGODR1, KGODR2, KGODT1, KGODT2,
   &    LAMNA, LAMNB, LAMNC, LAMND,
   &    LAMSA, LAMSB, LAMSC, LAMSD,
   &    LAMSTA, LAMSTB, DSTRA, DSTRB, SSRA, SSTRB,
   &    CLZA, CLZB, CLYA, CLYB,
   &    CZA, CZA, CZA, CZA, CZA, CZA,
   &    OVALA, OVALB, LAMWA, LAMWB, GAMA, GAMB
WRITE(3,902) WEOFA, WEOFB, WEOF1, WEOF2, WEOF3, WEOF4,
WRITE(3,903) KGODA, KGODB, KGODR1, KGODR2, KGODT1, KGODT2
WRITE(3,904) LAMNA, LAMNB, LAMNC, LAMND
WRITE(3,905) LAMSA, LAMSB, LAMSC, LAMSD
WRITE(3,906) LAMSTA, LAMSTB, DSTRA, DSTRB, SSRA, SSTRB,
WRITE(3,907) CLZA, CLZB, CLYA, CLYB,
WRITE(3,908) OVALA, OVALB, LAMWA, LAMWB, GAMA, GAMB
WRITE(3,909) PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1), VSTAT(2)
DO 25 I = 1, NLOAD
   WRITE(3,910) LDNAME(I), P(I), T(I), M(I,1), M(2,I), V(I,1), V(I,2)
   IF (NRAN .GT. MAXM) THEN
      WRITE(8,*), 'ERROR: STRESS-TIME HISTORY TOO LARGE'
      CALL TRMNT
   ENDIF
   25 CONTINUE

WRITE(3,911) KGID, KT(2,1), KT(2,2), LIMPR, BNRD, WEDS, IDWE,
   &    MNWT, WITD, EMOD, LOCAT, ANGLE, PERIOD, TRUNC,
   &    NLOAD, NRAN
DO 30 I = 1, NLOAD
   INQUIRE (FILE = LDNAME(I), EXIST = FTEST)
   IF (FTEST .EQV..TRUE.) THEN
      OPEN (FILNUM(I), FILE = LDNAME(I), STATUS = 'OLD')
      DO 31 J = 1, NRAN
         READ(FILNUM(I),*) STRHIS(I, J)
      31 CONTINUE
      CLOSE (FILNUM(I))
   ELSE
      WRITE(8,*), 'ERROR: CANNOT OPEN FILE, ', LDNAME(I),
      &    ' DOES NOT EXIST'
      CALL TRMNT
   ENDIF
   30 CONTINUE
   INQUIRE (FILE = LDNUM, EXIST = FTEST)
   IF (FTEST .EQV..TRUE.) THEN
      OPEN (FILNUM(I), FILE = LDNUM, STATUS = 'OLD')
      DO 31 J = 1, NRAN
         READ(FILNUM(I),*) STRHIS(I, J)
      31 CONTINUE
      CLOSE (FILNUM(I))
   ELSE
      WRITE(8,*), 'ERROR: CANNOT OPEN FILE, ', LDNUM,
      &    ' DOES NOT EXIST'
      CALL TRMNT
   ENDIF

7 - 34
ENDIF
30 CONTINUE

C ** CALL INFGAG TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECT
OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS

CALL INFGAG (RANGEM, MU, SIG, NF, NPTS, ZZERO, ZREG, NUMREG, &
NBD, STR, FTU, FTT, VARY, MPROC, KNATIO, PVAR)

ZREG = 1
ZZERO = 0.0

IF (MPROC.EQ.1) PSIG = SQRT (PVAR)
MCOUNT = 0

C ** INITIALIZE VARIABLES

DO 35 K = 1, MAXLIF
LIFE(K) = 1.0E+36
35 CONTINUE

DO 40 J = 1, MAXBIF
BLFPOS(J) = 0
40 CONTINUE

NLIFET = NHYPER * NLIFE
ANGLE = ANGLE * PI / 180.0

C ** OUTER LOOP -- THIS LOOP SAMPLES HYPER-PARAMETER SETS

DO 150 K = 1, NHYPER

C ** CALL PFRV TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS

CALL PFRV (TEST, RAND)
IF (TEST .LE. WEOF) THEN
CALL PFRV (RAND,WEOF1,WEOF2,WEOFT1, WEOFT2, WEOF, WEOF)
WEOFLO = WEOF1
WEOFHI = WEOF2
ELSE
CALL PFRV (RAND,WEOF3,WEOF4,WEOFT3, WEOFT4, WEOF, WEOF)
WEOFLO = WEOF3
WEOFHI = WEOF4
ENDIF
IF (OUT .EQ. 15) THEN
WRITE (8,'(TEST = ', TEST, ' WEOF = ', WEOF)
ENDIF

CALL PFRV (RAND, KWIDR1, KWIDR2, KWDI1, KWD1T, KWDTR, KWIDR, KWDTR)
CALL PFRV (RAND, KGODR1, KGODR2, KGDOR1, KGDOR2, KGODR, KGODR)
CALL PFRV (RAND, LAMHA, LAMNB, LAMSA, LAMSB, LAMNK, LAMSK)
LAMH = LAMN / (1.0 + LAMNK * LAMNC)
LAMNSG = LAMNC / (1.0 + LAMNK * LAMNC)
LAMSU = LAMSD / (1.0 + LAMSK * LAMSC)
LAMSSLG = LAMSC / (1.0 + LAMSK * LAMSC)

IF (OUT .EQ. 15) THEN
WRITE (8,'(LAMNK = ', LAMNK, ' LAMNU = ', LAMNU, &
LAMNS = ', LAMNS, ' LAMSSL = ', LAMSSL, &
ENDIF

C ** CALL PAREST TO PERFORM THE PARAMETER ESTIMATION ASPECT OF THE
MATERIALS CHARACTERIZATION MODEL CALCULATIONS

CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, Z goreg, &
RAND, NBD, STR, 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)
  CONTINUE

C ** INNER LOOP -- THIS LOOP GENERATES DUCT LIVES TO FAILURE

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, WEOF, WEOFT, WEOFLO, WEOFHI, WEOF)
CALL BETAGN (RAND, KWIDR, KWIDT, KWIDA, KWIDB, KWID)
CALL BETAGN (RAND, KMODR, KMODT, KMODA, KMODB, KMOD)
CALL BETAGN (RAND, KGODR, KGODT, KGODA, KGODB, KGOD)
CALL NORMGN (RAND, LAMNU, LAMNSG, LAMN)
CALL NORMGN (RAND, LAMNU, LAMNSG, LAMN)
CALL PRYRV (RAND, OVALA, OVALE, LAMST, OVAL, LAMST)
CALL PRYRV (RAND, DSTRA, DSTRB, SSTR, DSTR, SSTR)
CALL PRYRV (RAND, CLZA, CLZB, CLYA, CLYB, CLZ, CLY)
CALL PRYRV (RAND, CCZA, CCZB, CCYA, CCYB, CCZ, CCY)
CALL PRYRV (RAND, GAMA, GAMAB, GAMA, GAMAB, GAMA)
GAM = EXP(GAM)
CALL WEIBGN ( BZERO, RAND, PHI)

IF (VARY .EQ. 0) PHI = 1.0

IF (IOUT .EQ. 15) THEN
  WRITE(8,*) 'WEOF =',WEOF,' KWID =',KWID
  WRITE(8,*) 'KMOD =',KMOD
  WRITE(8,*) 'KGOD =',KGOD
  WRITE(8,*) 'LAMN =',LAMN
  WRITE(8,*) 'LAMS =',LAMS
  WRITE(8,*) 'LAMST =',LAMST
  WRITE(8,*) 'DSTR =',DSTR
  WRITE(8,*) 'SSTR =',SSTR
  WRITE(8,*) 'CLZ =',CLZ
  WRITE(8,*) 'CLY =',CLY
  WRITE(8,*) 'CCZ =',CCZ
  WRITE(8,*) 'CCY =',CCY
  WRITE(8,*) 'OVAL =',OVAL
  WRITE(8,*) 'LAMN =',LAMN
  WRITE(8,*) 'OVAL =',OVAL
  WRITE(8,*) 'GAM =',GAM
  WRITE(8,*) 'PHI =',PHI
ENDIF

C ** SCALE TIME-VARYING LOADS

PSLAM = PSTAT * LAMST * SSTR
TSLAM = TSTAT * LAMST * SSTR
MSLAM(1) = MSTAT(1) * LAMST * SSTR
MSLAM(2) = MSTAT(2) * LAMST * SSTR
VSLAM(1) = VSTAT(1) * LAMST * SSTR
VSLAM(2) = VSTAT(2) * LAMST * SSTR

DO 235 II = 1, NLOAD
  IF (TYPE(II) .EQ. 1) THEN
    PLAM(II) = P(II) * LAMN * DSTR
    TLAM(II) = T(II) * LAMN * DSTR
    MLAM(1,II) = M(1,II) * LAMN * DSTR
    MLAM(2,II) = M(2,II) * LAMN * DSTR
    VLAM(1,II) = V(1,II) * LAMN * DSTR
    VLAM(2,II) = V(2,II) * LAMN * DSTR
  ELSE
    PLAM(II) = P(II) * LAMS * DSTR
  ENDIF
235 CONTINUE

7 - 36
TLAM(II) = T(II) * LAMS * DSTR
MLAM(1,II) = M(1,II) * LAMS * DSTR
MLAM(2,II) = M(2,II) * LAMS * DSTR
VLAM(1,II) = V(1,II) * LAMS * DSTR
VLAM(2,II) = V(2,II) * LAMS * DSTR

ENDIF

CONTINUE

IF (IOUT .EQ. 15) THEN
WRITE(8,*), 'STATIC LOADS', P = ', PSLAM, ', T = ', TSLAM,
&
WRITE(8,*), MZ = ', MSLAM(1),
&
WRITE(8,*), M3 = ', MSLAM(2), ', V2 = ', VSLAM(1),
&
V3 = ', VSLAM(2)

WRITE(8,*), 'TIME-VARYING LOADS',
DO 240 II = 1, NLOAD
WRITE(8,*), II, ' TYPE = ', TYPE(II),
WRITE(8,*), MZ = ', MSLAM(1,II),
&
WRITE(8,*), M3 = ', MSLAM(2,II), ', V2 = ',
&
V3 = ', VSLAM(2,II)
240 CONTINUE

ENDIF

C ** CALCULATE AXIAL Kt's

KT(1,1) = KGOD * KWOD
KT(1,2) = KGID * KWID

IF (IOUT .EQ. 15) THEN
WRITE(8,*), 'KT(1,1) = ', KT(1,1), ', KT(1,2) = ', KT(1,2)
ENDIF

C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS

DO 250 L = ZREG, NUMREG
LNA(L) = MM(L) * ALOG(BIGK(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 = ', BIGK(L)
WRITE(8,*), 'LNA = ', LNA(L), ', PHI = ', PHI
WRITE(8,*), 'LPHIM = ', LPHIM(L), ', SBND = ', SBND(L)
WRITE(8,*), 'KRATIO = ', KRATIO, ', Z = ', Z
ENDIF

WRITE(8,*), 'TRSBND = ', TRSBND(L)
250 CONTINUE

C ** CALL ELWELD OF PIPE V8.3 TO CALCULATE FATIGUE LIFE

NEWLIF = GAM * ELWELD(PSLAM, MSLAM, TSLAM, VSLAM, NLOAD,
&
PLAM, MLAM, TLAM, VLAM, FTY, FTU, EMOD, KT,
&
LIMPR, WTID, MNWT, IDW, WEOW, LAMW, BNDR,
&
WEED, FK, RT, CC, CCY, CL, CCL, CCY, CCZ, CLZ, OVAL,
&
LOCAL, HM, LNA, LPHIM, KRATE, LNZ, TRSBND,
&
SZR, ZREG, NUMREG, STRHIS, NRAIN,
&
PERIOD, TRUNC, ANGLE, CULPR)

IF (IOUT .EQ. 15) WRITE(8,*), 'NEWLIF = ', NEWLIF
IF (NEWLIF .GE. 100) CALL INSORT (NEWLIF, LIFE, NLIFET)

200 CONTINUE

150 CONTINUE

IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES

DO 300 J = 1, (NLIFET / 100)
WRITE(9,*), J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
300 CONTINUE

C ** PRINT EMPIRICAL BLIVES

7 - 37
FIFTY = 0.50E0
WRITE(3,925)
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)
   CONTINUE
   CALL EXPCTD (I, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG,
            NNBD, BIGKI, BZERO)
ENDIF
C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO DCTHC0
900 FORMAT(2X, 'Copyright (C) 1990, California Institute of Technology. U.S. Government', 'NASA Contract NAS7-918 is acknowledged.', 'INPUT DATA', 'PARAMETER DISTRIBUTIONS', //, 'RHO', 'THETA')
901 FORMAT(/,2X,'WELD OFFSET (%), 3X,'Be(',F4.2,',F5.2)',6X,'U(',F7.5,',F8.5)',4X,'U(',F4.1,',F5.1)',/)
902 FORMAT(/,2X,'K WELD (OD)',7X,'Be(',F4.2,',F5.2)',6X,'U(',F7.5,',F8.5)',4X,'U(',F4.1,',F5.1)',/)
903 FORMAT(/,2X,'K WELD (ID)',7X,'Be(',F4.2,',F5.2)',6X,'U(',F7.5,',F8.5)',4X,'U(',F4.1,',F5.1)',/)
904 FORMAT(/,2X,'K GEOM (OD)',7X,'Be(',F4.2,',F5.2)',6X,'U(',F7.5,',F8.5)',4X,'U(',F4.1,',F5.1)',/)
905 FORMAT(/,2X,'LAMBDA RANDOM',5X,'k: U(',F7.5,',F8.5)',//' ,20X,'COEFFICIENT OF VARIATION: ',F5.3,
uenta,'DYNAMIC STRESS ANALYSIS',13X,'U(',F8.5,',F9.5)',//, 'STRESS CARRYOVER FACTORS', 'IN-PLANE AXIAL', 19X,'U(',F8.5,',F9.5)',//, 'OUT-OF-PLANE AXIAL', 15X,'U(',F8.5,',F9.5)',//, 'IN-PLANE CIRCUMFERENTIAL', 9X,'U(',F8.5,',F9.5)',//, 'OUT-OF-PLANE CIRCUMFERENTIAL', 5X,'U(',F8.5,',F9.5)',//, 'OVALITY ANALYSIS FACTOR', 'LAMBDA SINE', 7X,'k: U(',F7.5,',F8.5)',//' ,20X,'COEFFICIENT OF VARIATION: ',F5.3,
uenta,'DAMAGE MODEL ACCURACY',15X,'U(',F8.5,',F9.5)',//,
"ln(',F8.5,)"
STOP
END

C*****************************************************************************
C* SAMPLE 'DCTHCD' INPUT FILE
C*****************************************************************************

C 675.................. RANDOM NUMBER SEED
C 0.................. OUTPUT DUMP CONTROLLER
C 100.................. INNER LOOP SIZE
C 200.................. OUTER LOOP SIZE
C 1.................. S/N VARIATION
C 0.................. MATERIALS PROC. VAR. NOT REQUIRED
C 0.................. NUMBER OF LIVES TO BE PROVIDED
C 0.0001 B.01 LIFE
C 0.001 B.1 LIFE
C 0.01 B1 LIFE
C 0.20 0.40 0.60 0.80 1.00 WELD OFFSET (A,B) (R1,R2) (T1,T2)
C 0.20 0.50 0.00 0.2 0.3 0.5 1.00 (C,D) (R3,R4) (T3,T4)
C 1.00 1.80 0.30 0.30 0.30 0.30 0.30 TEST FOR HYPER-DIST.
C 1.40 K WELD (OD) (A,B) (R1,R2) (T1,T2)
C 1.40 K WELD (ID) (A,B) (R1,R2) (T1,T2)
C 1.20 K GEOM (OD) (A,B) (R1,R2) (T1,T2)
C 1.20 K GEOM (ID) (A,B) (R1,R2) (T1,T2)
C 1.00 1.00 0.15 0.60 0.90 LAMBDABW BAND-RANDOM: K: U(A,B),
C 2.00 3.00 0.20 0.90 COEFF. OF VAR., STRAIN GAGE FACTOR
C 0.90 1.10 LAMDBA SUPERIMPOSED SINE: K: U(A,B),
C 0.80 1.20 COEFF. OF VAR., STRAIN GAGE FACTOR
C 0.90 1.10 DYNAMIC STRESS ANALYSIS ACCURACY FACTOR
C 0.40 0.60 STATIC STRESS ANALYSIS ACCURACY FACTOR
C 0.85 1.15 STRESS CARRYOVER FACTORS:
C 0.40 0.60 0.60 0.60 STRESS CARRYOVER FACTORS:
C 0.40 0.60 IN-PLANE AXIAL
C 0.40 0.60 OUT-OF-PLANE AXIAL
C 0.40 0.60 IN-PLANE CIRCUMFERENTIAL
C 0.40 0.60 OUT-OF-PLANE CIRCUMFERENTIAL
<table>
<thead>
<tr>
<th>FILE TYPE</th>
<th>P()</th>
<th>T()</th>
<th>M(1,*)</th>
<th>M(2,*)</th>
<th>V(1,*)</th>
<th>V(2,*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>'NBP'</td>
<td>1</td>
<td>237.675</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>'NBT'</td>
<td>1</td>
<td>103.41</td>
<td>0.00</td>
<td>181.21</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>'NBM2'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>626.175</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>'NBV1'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>147.49</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>'NBV3'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>178600.</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>'NBV4'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>6.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>'NBV5'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>'NBV6'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>6.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>'NBV7'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>'NBV8'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>6.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>'NBV9'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>1.0</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>'NBV10'</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>6.0</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GEOM (ID)</th>
<th>K HOOP (ID)</th>
<th>K HOOP (ID)</th>
<th>K HOOP (ID)</th>
<th>K HOOP (ID)</th>
<th>K HOOP (ID)</th>
<th>K HOOP (ID)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LIMIT PRESSURE</th>
<th>BEND RADIUS</th>
<th>WELD DIST. FROM ELBOW TANG. LINE</th>
<th>DUCT INSIDE DIAMETER</th>
<th>MINIMUM WALL THICKNESS</th>
<th>WALL THICKNESS AT BEND</th>
<th>ELASTIC MODULUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>4902</td>
<td>6.0</td>
<td>1.1</td>
<td>4.0</td>
<td>0.113</td>
<td>0.134</td>
<td>1.01E07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCATION OF INTEREST</th>
<th>ANGLE OF INTEREST (DEG)</th>
<th>STRESS-TIME HISTORY PERIOD (SEC)</th>
<th>STRESS-TIME HISTORY NOISE FILTER (PSI)</th>
<th>NUM. POINTS IN STRESS-TIME HISTORY</th>
</tr>
</thead>
<tbody>
<tr>
<td>90.0</td>
<td>500.0</td>
<td>1000.0</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>YIELD &amp; ULTIMATE STRENGTHS, NDIV, NPTS</th>
<th>20</th>
<th>0.05</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>65000.0</td>
<td>S(1)</td>
<td>N(1)</td>
<td></td>
</tr>
<tr>
<td>621000.0</td>
<td>S(2)</td>
<td>N(2)</td>
<td></td>
</tr>
<tr>
<td>625000.0</td>
<td>S(3)</td>
<td>N(3)</td>
<td></td>
</tr>
<tr>
<td>677000.0</td>
<td>S(4)</td>
<td>N(4)</td>
<td></td>
</tr>
<tr>
<td>690000.0</td>
<td>S(5)</td>
<td>N(5)</td>
<td></td>
</tr>
<tr>
<td>715000.0</td>
<td>S(6)</td>
<td>N(6)</td>
<td></td>
</tr>
<tr>
<td>4198000.0</td>
<td>S(7)</td>
<td>N(7)</td>
<td></td>
</tr>
<tr>
<td>5035000.0</td>
<td>S(8)</td>
<td>N(8)</td>
<td></td>
</tr>
<tr>
<td>921000.0</td>
<td>S(9)</td>
<td>N(9)</td>
<td></td>
</tr>
<tr>
<td>9667000.0</td>
<td>S(10)</td>
<td>N(10)</td>
<td></td>
</tr>
<tr>
<td>418000.0</td>
<td>S(11)</td>
<td>N(11)</td>
<td></td>
</tr>
<tr>
<td>732000.0</td>
<td>S(12)</td>
<td>N(12)</td>
<td></td>
</tr>
<tr>
<td>740000.0</td>
<td>S(13)</td>
<td>N(13)</td>
<td></td>
</tr>
<tr>
<td>859000.0</td>
<td>S(14)</td>
<td>N(14)</td>
<td></td>
</tr>
<tr>
<td>1181000.0</td>
<td>S(15)</td>
<td>N(15)</td>
<td></td>
</tr>
<tr>
<td>402000.0</td>
<td>S(16)</td>
<td>N(16)</td>
<td></td>
</tr>
<tr>
<td>5917000.0</td>
<td>S(17)</td>
<td>N(17)</td>
<td></td>
</tr>
<tr>
<td>6522000.0</td>
<td>S(18)</td>
<td>N(18)</td>
<td></td>
</tr>
<tr>
<td>6891000.0</td>
<td>S(19)</td>
<td>N(19)</td>
<td></td>
</tr>
<tr>
<td>4460000.0</td>
<td>S(20)</td>
<td>N(20)</td>
<td></td>
</tr>
</tbody>
</table>
NO VALUE OF SO SUPPLIED

**LIST OF VARIABLES**

**2-D ARRAY CONTAINING M VALUES TO BE sortED FOR EACH REGION**

**ANGLE**
Angle phi measured counter-clockwise from z-direction --

**BIGK()**
1-D array containing values of k, where a = k ** M for each region

**BIG1()**
Equal to BIGK(1) -- dummy parameter for calls to subroutine EXPCTD

**BLFPER()**
1-D array containing user specified blives to be provided

**BLFPOS()**
1-D array containing position in LIFE() of empirical blives

**BNRD**
Bend radius, in.

**BZERO**
Value of Betao randomly selected from Beta() interval

**CCY**
Selected out-of-plane circumferential stress carryover factor

**CCYA**
CCY lower bound

**CCYB**
CCY upper bound

**CC2**
Selected in-plane circumferential stress carryover factor

**CC2A**
CC2 lower bound

**CC2B**
CC2 upper bound

**CLY**
Selected out-of-plane axial stress carryover factor

**CLYA**
CLY lower bound

**CLYB**
CLY upper bound

**CLZ**
Selected in-plane axial stress carryover factor

**CLZA**
CLZ lower bound

**CLZB**
CLZ upper bound

**CULPRT**
Location causing failure

**DSTRA**
Dynamic stress analysis accuracy factor lower bound

**DSTRB**
Dynamic stress analysis accuracy factor upper bound

**DUM**
Dummy variable

**ELWELD**
Real function which calculates the duct life (in seconds) at a weld near an elbow

**EMOD**
Material elastic modulus

**FILNUM()**
1-D array containing unit numbers for stress-time histories files

**FK()**
1-D array containing values of FK used to find stress concentration due to weld eccentricity

**FTKTEST()**
File test -- used to test existence of file

**FTU**
Material ultimate strength

**FTY**
Material yield strength

**GAM**
Selected damage accumulation model accuracy factor, LAMBdadam

**GAM LA**
GAM lower bound

**GAMB**
GAM upper bound

**I**
Controls do loop for each life calculation

**IODE**
Weld inside diameter, in

**IOUT**
Controls dump to screen/printer

**J**
Controls do loop for each b-life

**K**
Controls do loop for each hyper-parameter set

**KGID**
K geom (ID)

**KGOD**
Selected k geom (OD)

**KGODA**
K geom (OD) lower bound

**KGODR**
Selected rho for k geom (OD)

**KGOD1**
K geom (OD) - RHO lower bound

**KGOD2**
K geom (OD) - RHO upper bound

**KGODT1**
K geom (OD) - THETA lower bound

**KGODT2**
K geom (OD) - THETA upper bound

**KRAT**
Ratio of K*K, constant over regions and components

**Kt()**
Fatigue stress concentration factors -- Kt(1,1) =

Kt axial (OD) = KGID * KWID; Kt(1,2) = Kt axial (ID)

Kt hoop (OD); Kt(2,1) = Kt hoop (ID)
SELECTED K WELD (ID)
K WELD (ID) LOWER BOUND
K WELD (ID) UPPER BOUND
SELECTED RHO FOR K WELD (ID)
K WELD (ID) - RHO LOWER BOUND
K WELD (ID) - RHO UPPER BOUND
SELECTED THETA FOR K WELD (ID)
K WELD (ID) - THETA LOWER BOUND
K WELD (ID) - THETA UPPER BOUND
SELECTED RHO FOR K WELD (OD)
K WELD (OD) - RHO LOWER BOUND
K WELD (OD) - RHO UPPER BOUND
SELECTED THETA FOR K WELD (OD)
K WELD (OD) - THETA LOWER BOUND
K WELD (OD) - THETA UPPER BOUND
L CONTROLS DO LOOP FOR EACH REGION
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
LAMBDA FOR NARROW-BAND RANDOM LOADS COEFFICIENT OF VARIATION
NARROW-BAND RANDOM LOADS STRAIN GAGE ACCURACY FACTOR
NARROW-BAND RANDOM LOADS K -- INDICATES VARIATION 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
LAMBDA FOR SUPERIMPOSED SINE LOADS -- LOWER BOUND OF K
LAMBDA FOR SUPERIMPOSED SINE LOADS -- UPPER BOUND OF K
LAMBDA FOR SUPERIMPOSED SINE LOADS COEFFICIENT OF VARIATION
SUPERIMPOSED SINE LOADS STRAIN GAGE ACCURACY FACTOR
LAMBDA FOR SUPERIMPOSED SINE LOADS K -- INDICATES VARIATION DUE TO SAMPLE SIZE
MEAN OF LAMBDA FOR SUPERIMPOSED SINE LOADS (MU, NORMAL DISTRIBUTION)
STANDARD DEVIATION OF LAMBDA FOR SUPERIMPOSED SINE LOADS (SIGMA, NORMAL DISTRIBUTION)
SELECTED LAMBDA FOR STATIC LOADS
SELECTED LAMBDA FOR STATIC LOADS
L AM D A T (ID)
L AM D A T (OD)
SELECTED ACCURACY FACTOR FOR WELD ECCENTRICITY STRESS CONCENTRATION FACTOR, Koff
Koff
L AM A (ID)
L AM A (OD)
1-D ARRAY CONTAINING LOAD NAMES FOR THE TIME-VARYING LOADS
1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM -- SORTED VALUES OF THE LEFT-HANDED TAIL LIMIT PRESSURE, PSI (INTERNAL PRESSURE)
1-D ARRAY CONTAINING LN(A) = LN(BIGK)*MM FOR EACH REGION
NORMAL DISTRIBUTION (SIGMA, PVAR) GENERATED RANDOM VARIABLE
LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE DUCT AND 2 IS THE INTERIOR SURFACE OF THE DUCT
1-D ARRAY CONTAINING LN(BIGK)*MM FOR EACH REGION
2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
M(2,*), M(3,*) ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
MAXIMUM NUMBER OF BLIVES TO BE CALCULATED
MAXIMUM NUMBER OF POINTS PER DATA SET PER REGION ALLOWED
MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA CALCULATION
MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
MAXIMUM NUMBER OF M'S TO BE SORTED FOR MEDIAN CALCULATION
MAXIMUM NUMBER OF REGIONS ALLOWED
1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
POINTER TO THE MEDIAN M VALUES -- EQUAL TO HALF OF MCOUNT
2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS SCALLED BY DSTR AND LAMS OR LAMN AS APPROPRIATE (INDICATED BY TYPE(1)) -- MLAM(1,*) ARE THE M2 LOADS; MLAM(2,*) ARE THE M3 LOADS
2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS — V(I,*) ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS

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

2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS SCALED BY DSTR AND LAMN OR LAMS AS APPROPRIATE (INDICATED BY TYPE()) — VLAM(1,*) ARE THE V2 LOADS; VLAM(2,*) ARE THE V3 LOADS

1-D ARRAY CONTAINING THE STATIC SHEAR LOADS SCALED BY DSTR AND _ OR LAMS AS APPROPRIATE (INDICATED BY TYPEO) — VSLAM(1) IS THE V2 LOAD; VSLAM(2) IS THE V3 LOAD

1-D ARRAY CONTAINING THE STATIC SHEAR LOADS SCALLED BY LAMST AND SSTR — VSTAT(1) IS THE V2 LOAD; VSTAT(2) IS THE V3 LOAD

WELD DISTANCE FROM ELBOW TANGENCY LINE, IN

SELECTED WELD OFFSET (%)

WELD OFFSET LOWER Bound — HYPER-DISTRIBUTION 1
WELD OFFSET UPPER Bound — HYPER-DISTRIBUTION 1
WELD OFFSET LOWER Bound — HYPER-DISTRIBUTION 2
WELD OFFSET UPPER Bound — HYPER-DISTRIBUTION 2
PERCENTAGE OCCURRING IN HYPER-DISTRIBUTION 1
SELECTED WELD OFFSET UPPER Bound
SELECTED WELD OFFSET LOWER Bound
SELECTED RHO FOR WELD OFFSET
WELD OFFSET — RHO LOWER Bound — HYPER-DISTRIBUTION 1
WELD OFFSET — RHO LOWER Bound — HYPER-DISTRIBUTION 1
WELD OFFSET — RHO UPPER Bound — HYPER-DISTRIBUTION 2
WELD OFFSET — RHO UPPER Bound — HYPER-DISTRIBUTION 2
SELECTED THETA FOR WELD OFFSET
WELD OFFSET — THETA LOWER Bound — HYPER-DISTRIBUTION 1
WELD OFFSET — THETA LOWER Bound — HYPER-DISTRIBUTION 1
WELD OFFSET — THETA UPPER Bound — HYPER-DISTRIBUTION 2
WELD OFFSET — THETA UPPER Bound — HYPER-DISTRIBUTION 2
SELECTED WALL THICKNESS AT BEND (ID) AT WELD, IN
LOG-NORMAL(0,PVAR) GENERATED RANDOM VARIATE

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

SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
PROGRAMMER: L. NEWLIN
DATE: 20JUN90
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)

INPUTS: NEWLIF, LIFE, NLIFET
OUTPUTS: LIFE

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

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(THE1,THE2)
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 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
SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)

COMMON IOUT

DOUBLE PRECISION RAND

REAL A, B, GAM, RHO, THETA, W, X, Y1, Y2

INTEGER IOUT


Y1 = GAM(RHO * THETA + 1.), RAND
Y2 = GAM((1. - RHO) * THETA + 1.), RAND
W = Y1 / (Y1 + Y2)

IF (IOUT .EQ. 15) WRITE(8,*), 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

X = W * (B - A) + A

IF (IOUT .EQ. 15) WRITE(8,*), 'W =', W, ' X =', X

RETURN
END

C**************************************************************************
C THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: GAM
C
C The random variates are generated using the method described in:
C Johnson, N. L. and Kotz, S., Distribution in Statistics: Continuous
C Univariate Distributions - 1, Houghton Mifflin Company, 1970,
C pp. 181-182.
C**************************************************************************

SUBROUTINE BETAGN (RAND, RHO, THETA, A, B, X)

COMMON IOUT

DOUBLE PRECISION RAND

REAL A, B, GAM, RHO, THETA, W, X, Y1, Y2

INTEGER IOUT


Y1 = GAM(RHO * THETA + 1.), RAND
Y2 = GAM((1. - RHO) * THETA + 1.), RAND
W = Y1 / (Y1 + Y2)

IF (IOUT .EQ. 15) WRITE(8,*), 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

X = W * (B - A) + A

IF (IOUT .EQ. 15) WRITE(8,*), 'W =', W, ' X =', X

RETURN
END

C**************************************************************************
C THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: GAM
C
C The random variates are generated using the method described in:
C Johnson, N. L. and Kotz, S., Distribution in Statistics: Continuous
C Univariate Distributions - 1, Houghton Mifflin Company, 1970,
C pp. 181-182.
C**************************************************************************

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.
C IF (IOUT .EQ. 15) WRITE(8,*) 'A = ', A, ', ', 'ALPHA = ', ALPHA

10 CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALOG(U1)
V2 = - ALOG(U2)
C IF (IOUT .EQ. 15) WRITE(8,*) 'U1 = ', U1, ', ', 'U2 = ', U2, ', ', 'V1 = ', V1, ', ', 'V2 = ', V2
& ARG = A * (V1 - ALOG(V1) - 1.)
IF (V2 .LT. ARG) GOTO 10
GAM = ALPHA * V1
C IF (IOUT .EQ. 15) WRITE(8,*) 'GAMMA = ', GAM
RETURN
END

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

C SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C FOR THE STRESS FORMULATION
C PROGRAMMER: L. NEWLIN
C DATE: 13JUL89 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 is acknowledged.

SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
& NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
& KRATIO, PVAR)
C INPUTS: READS DATA FROM SPECFD AND RELATD. VARY, MPROC
C OUTPUTS: RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG,
& NUMREG, NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C SUBPROGRAMS: _NIT. RCE, SW2SU2__FINDMC, INTRVL, FNDRNG, ADDR_G,
C CONCAV, MEDIAN_ EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR
C FILES: 5:RELATD-OLD; 6:RELATD-NEW
C IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPR, NNODAT,
& NF (0:MAXSET, MAXREG), NFPR(MAXREG), NFTS (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, LNSTR (MAXDAT, 0:MAXSET, MAXREG),
& MCHAT (2, MAXREG), MEDM (MAXREG), MO (MAXREG), MU (MAXREG),
& MZERO (2, MAXREG), NBND (0:MAXREG), NF (MAXDAT, MAXREG),
& PVAR, RANGEM (2, MAXREG), RAISTR (MAXDAT, 0:MAXSET),
& RRAWF (MAXDAT, 0:MAXSET), RAWNS (MAXDAT, 0:MAXSET),
& SIGMA2 (MAXREG), SIGMA2 (MAXREG), STR (MAXDAT, MAXREG),
& SUBM2 (MAXREG), SWHAT2 (MAXREG), SX2 (MAXREG),
& SKY (MAXREG), SY2 (MAXREG), SZERO

C LIST OF VARIABLES
C BIGKHT EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
CZERO: Value of Weibull parameter, Beta0, characterizing the S/N data set.

BZERO: Exogenous information in the form of a constraint on the coefficient of variation, CO.

DD(): 1-D array containing sky(L)/(L2(L)) for each region.

DELA(): 1-D array containing Bayesian multiplier used in Mu().

FTUZ: Ultimate strength (PSI) for specific material.

FTY2: 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 J0, the 95% confidence intervals on M for each region.

KRA()T: Ratio of K*/K, constant over regions and components.

LAMN: 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.

LNNF(): 3-D array containing ln(Rawnf()), also indexed for region.

LNNF(): 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. Values set with given value of C0 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) = SUBAT, 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.

MC(): 1-D array containing values of the prior normal distribution mean for each region.

MPNT(): 1-D array containing the number of points, 0, 1, or 2, in MZERO() for each region.

MPROC: Materials process variation -- controls materials process variation -- 0 -- no variation; 1 -- variation.

MU(): 1-D array containing values of the posterior normal distribution mean for each region.

MZERO(): 2-D array containing values of the prior ranges on M for each region -- MZERO(1,L) is the lower bound and MZERO(2,L) 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).

NPPTS(): 1-D array containing number of points in S/N data sets.

NSETS: Number of related material S/N data sets.

NUMREG: Number of regions of interest.

PVAR: Materials process variation.

RANGEM(): 2-D array containing values of the posterior ranges on M for each region -- RANGEM(1,L) is the lower bound and RANGEM(2,L) is the upper bound.

RATSTR(): 2-D array containing stress data (PSI) corrected for stress ratio or total strain data (%) for all S/N data sets.

RAWNF(): 2-D array containing raw cycles to failure data for all S/N data sets.

RAWSTR(): 2-D array containing raw stress data (PSI) or total strain data (%) for all S/N data sets.

REFNP(): 1-D array containing the number of points for the specific (reference) material S/N data set in each region.

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)
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
ZROREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

OPEN(5, FILE = 'RELATO', 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 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, NODAT,
& NSETS, NBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO,
& SIGMA2, KRATIO, LAMN)
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
& SWHAT2, SUHAT2, NPPR)
C CALCULATE M CONSTRAINT 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. 1) THEN
CALL GTPVAR (NSETS, NP, NUMREG, LAHN, 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 ADDRGE (RANGEM, MCHAT, NODAT, 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
WRITE(7,910)
DO 50 L = 1, NUMREG
   WRITE(7,915) L, MCHAT(2,L), MCHAT(1,L)
50 CONTINUE
IF (CZERO .GT. 0.0) THEN
   WRITE(7,960)
   DO 150 L = 1, NUMREG
      IF (MCPNT(L) .EQ. 1) THEN
         WRITE(7,965) L, MC(I,L)
      ELSEIF (MCPNT(L) .EQ. 2) THEN
         WRITE(7,970) L, MC(1,L), MC(2,L)
      ENDIF
150 CONTINUE
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 100 L = 1, NUMREG
   WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
100 CONTINUE
WRITE(7,950)
C CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO
CALL MEDIAN (NUMREG, RANGEM, MEDM)
C CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
   CALL EXPCTD (I, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG, NBND, BIGKHT, BZERO)
C CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED
   IF ((VARY .NE. 0) .OR. (VARY .EQ. 1)) THEN
      DO 200 L = 1, NUMREG
         RANGEM(1,L) = MEDM(L)
         RANGEM(2,L) = MEDM(L)
200 CONTINUE
   ELSE
      C NORMAL VARIATION IS DESIRED
      C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF
   ENDIF
C COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M
CALL NORRNG (NUMREG, MPNT, MEZERO, MCPNT, MC, MCHAT, RANGEM)

C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
& CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MEZERO, 
& 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 = i, NUMREG
    IF (MCPNT(L) .EQ. j) THEN
      WRITE(7,965) L, MC(L)
    ELSEIF (MCPNT(L) .EQ. 2) THEN
      WRITE(7,970) L, MC(L), MC(I)
    ENDIF
  360 CONTINUE
ENDIF
WRITE 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, J(L)
380 CONTINUE
ENDIF

C PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS
IF (MPROC .EQ. i) THEN
  WRITE(7,995) PVAR
ENDIF

C FORMAT STATEMENTS
900 FORMAT(2X,'Copyright (C) 1990, California Institute of ', 
& 'Technology. U.S. Government',/2X,'Sponsorship under ', 
& 'NASA Contract NAS7-918 is acknowledged.',//,/, 
& 'RESULTS OF INFORMATION AGGREGATION CALCULATIONS', 
& '//,2X,'95% CONFIDENCE INTERVALS ON C AND m', 
& 'FOR EACH REGION',/) 
905 FORMAT(7X,'REGION: ',I1,7X,'Io = (',F12.9,','',F12.9,')', 
& ',2X,'Jo = (',F12.9,','',F12.9,')')</n> 
910 FORMAT(///,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION', 
& 'REGION',8X,'E(C)',12X,'E(m)',/) 
915 FORMAT(9X,11,8X,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',/)
SUBROUTINE TRMNT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED

PROGRAMMER: L. NEWLIN
DATE: 5OCT87

SUBROUTINE TRMNT
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, & REFPN, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2)

INPUTS: ---
OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFPN, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2
IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

COMMON IOUT

INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
NPTS(0:MAXSET), REFNF(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),
& SISM2(MAXREG), STR(MAXDAT, MAXREG)

LIST OF VARIABLES

DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
& SIG() CALCULATION
I  CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT OUTPUT DUMP CONTROLLER
J  CONTROLS DO LOOP FOR EACH DATA SET
K  CONTROLS DO LOOP FOR EACH POINT IN A REGION
L  CONTROLS DO LOOP FOR EACH REGION
LNSTR() 3-D ARRAY CONTAINING LN(RAWNF()), 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
& EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
& IS THE UPPER BOUND
NFR() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
& SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
& IN EACH REGION
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
RATSTR() 2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
& STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
RAWNF() 2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
& DATA SETS
RAWSTR() 2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OF TOTAL STRAIN
& DATA (%) FOR ALL S/N DATA SETS
REFNF() 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
      DO 450 J = 0, MAXSET
         LNNF(K, J, L) = 0.0
         LNSTR(K, J, L) = 0.0
      450 CONTINUE
   425 CONTINUE
400 CONTINUE

DO 500 K = 1, MAXDAT
   DO 550 J = 0, MAXSET
      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(0, L) = 0.0
   MZERO(2, L) = 0.0
   DELTA(L) = 0.0
   MO(L) = 0.0
   SIGMA2(L) = 0.0
600 CONTINUE

RETURN
END

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

C SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATD; "CONVERTS" C THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO C SPECFD AND RELATO. RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS C SPECIFIED BY USER
C PROGRAMMER: L. NEWLIN
C DATE: 21JUN88 FORMAT/COMMENTS: 12AUG91

SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
               & LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
               & NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
               & FTU, FTUZ, DELTA, MO, SIGMA2, KRATIO, LAMN)

C INPUTS: VARY, MPROC
C OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP,
C STR, NF, SZERO, ZROREG, NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
C FTU, FTUZ, DELTA, MO, SIGMA2, KRATIO, LAMN
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,
       & NNODAT, 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),
      & LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
      & MPNT(MAXDAT, MAXREG), NBND(0:MAXDAT, 0:MAXREG), NP(MAXDAT, MAXREG),
      & RATI0, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
      & RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
LIST OF VARIABLES

COUNTER INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO, OUTPUT, AND BREAKUP
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, COV, COV
DELT A() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
DESCRP() 1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
FTU ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
FTY YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTY2 YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
I CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
ΚRATIO RATIO OF Κ*/Κ, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LAMN LAMBDA-N -- RATIO OF Var (ln N given S) / (m**2 c**2), CONSTANT OVER ALL REGIONS AND COMPONENTS
LNLF() 3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
M CONTROLS DO LOOP FOR EACH DATA DIVISION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MPROC Materials PROCESS variation -- CONTROLS MATERIALS PROCESS VARIATION -- 0 - NO VARIATION; 1 - VARIATION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON 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
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 (θ = -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, SUCH AS VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
ZROREG Zero Region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
C 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), FTU, FTY, NDIV, NPTS(0)
IF (NPTS(0) .GT. MAXDAT) THEN
  WRITE(8,*) 'ERROR: OVER NUMBER OF POINTS LIMIT IN ',
  & 'SPECIFIC MATERIAL'
  CALL TRMNAT
END IF
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)

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
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
END IF
IF (REG .GT. MAXREG) THEN
  WRITE(8,*) 'ERROR: OVER REGION LIMIT IN SPECIFIC DATA SET'
  CALL TRMNAT
END IF
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
IF (RATIO .EQ. -1.0) THEN
  STRESS RATIO IS CORRECT
  DO 120 I = (COUNT + 1), (COUNT + NUM)
    RATSTR(I,0) = RAWSTR(I,0)
 120 CONTINUE
ELSE
  STRESS RATIO TRANSFORMATION MUST BE DONE
  CALL CONVRT (0., (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR,
  & RATIO, FTU, FTY)
END IF

C ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL
DO 130 I = (COUNT + 1), (COUNT + NUM)

7-56
BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2, EXPECTD, 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(RAWNФ(I, 0))
STR(K, REG) = RATSTR(I, 0)
NF(K, REG) = RAWNF(I, 0)

IF (K .GT. MAXDAT) THEN
WRITE(8, *) 'ERROR: OVER NUMBER OF POINTS LIMIT IN ', 'SPECIFIC MATERIAL'
CALL TRMNAT
ENDIF
NP(0, REG) = K
REFNФ(REG) = K
COUNT = COUNT + NUM

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)

WRITE(3, 913)
IF (ZROREG .EQ. 0) WRITE(3, 914) SZERO
IF (IOUT .EQ. 10) THEN

130 CONTINUE

140 CONTINUE

100 CONTINUE

150 CONTINUE

160 CONTINUE
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(a,920) NBND(L)
  170 CONTINUE
WRITE(3,925) CZERO
IF (IOUT .EQ. 10) WRITE(8,925) CZERO
DO 180 L = 1, (NUMREG + NNODAT)
  WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
  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)) WRITE(8,*) 'ERROR: NORMAL VARIATION REQUIRES A PRIOR '
    & ' RANGE ON M'
    CALL TRMNAT
  180 CONTINUE
IF (VARY .EQ. 3) THEN
  C
  READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
  WRITE(3,945)
  IF (IOUT .EQ. 10) WRITE(8,945)
  DO 190 L = 1, (NUMREG + NNODAT)
    READ(1,*) DELTA(L), MO(L), SIGMA2(L)
    WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
    IF (IOUT .EQ. 10) WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
    IF ((DELTA(L) .LT. 0.0) .OR. 
      & ((DELTA(L) .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
  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
DO 200 J = 1, NSETS
  COUNT = 0
  IF (IOUT. EQ. 10) WRITE(8,*), 'B', J, ' NSETS = ', NSETS
  READ(5,*) DESCRIPT(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(6,935) NSETS
DO 200 J = 1, NSETS
**ENDIF**
WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)
WRITE(6,905)
IF (IOUT .EQ. 10) WRITE(8,905)
DO 300 M = 1, NDIV
READ(5,*) NUM, RATIO, REG
IF (ABS(RATIO) .GT. 1.0) THEN
  WRITE(8,*), 'ERROR: INVALID VALUE OF RATIO: ', RATIO
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)
CONTINUE
C CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
IF (RATIO .EQ. -1.0) THEN
  C STRESS RATIO IS CORRECT
  DO 320 I = (COUNT + 1), (COUNT + NUM)
  CONTINUE
ELSE
  C STRESS RATIO TRANSFORMATION MUST BE DONE
  CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR, RATIO, FTU, FTY)
ENDIF
C RECORD BOTH S/N DATA SETS TO RELATO
DO 330 I = (COUNT + 1), (COUNT + NUM)
   WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG, RATSTR(I,J), RAWNF(I,J)
   IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG, RATSTR(I,J), RAWNF(I,J)
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(RAWNFP(I,J))
CONTINUE
IF (K .GT. MAXDAT) THEN
  WRITE(8,*), 'ERROR: OVER LIMIT ON NUMBER OF POINTS ', ' IN SET ', J
7-59
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
END
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 ',9X,'REGION(S) WITH DATA ',
  & ',2X,'AND ',2X,'REGION(S) TO THE RIGHT WITHOUT DATA',
  & ',2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
  & '(CYCLES): ')
920 FORMAT(10X,E9.3)
925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
  & 'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
  & /,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
  & /,2X,'REGION',5X,'OF POINTS',5X,'LOWER BOUND',
  & 'UPPER BOUND',/)
930 FORMAT(6X,I1,11X,I1,12X,F7.4,9X,F7.4)
935 FORMAT(20X,'NUMBER OF DATA SETS: ',2X,I2,///,17X,
  & 'NOTE: ALL Kt ASSUMED TO BE 1.0',///,23X,
  & 'TRANSFORMED DATA')
940 FORMAT(///,2X,'DESCRIPTION:',2X,A40,
  & /,2X,'YIELD STRENGTH',18X,F7.0,
  & /,2X,'ULTIMATE STRENGTH',15X,F7.0,
  & /,2X,'NUMBER OF POINTS',16X,I2)
945 FORMAT(///,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS:',
  & /,2X,'REGION',5X,'DELTA',8X,'MO',10X,'SIGMA2',/)
950 FORMAT(5X,I1,5X,F7.2,5X,F7.4,5X,E11.5)
955 FORMAT(///,2X,'MATERIALS PROCESS VARIATION INFORMATION',
  & /,2X,'MEDK/MEDK:',5X,E11.5,///,5X,'LAMBDAN:',5X,E11.5)
RETURN
END

C*******************************************************************************
THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
STRESS RATIO, R, IS NOT -1.0

PROGRAMMER: L. NEWLIN
DATE: CODE: 6OCT87 COMMENTS: 13JUL89
VERSION: MATCHR V6, V6.1, 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 (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

C LIST OF VARIABLES

FTU    ULTIMATE STRENGTH OF MATERIAL (PSI)
FTY    YIELD STRENGTH OF MATERIAL (PSI)
I      CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
IOUT   OUTPUT DUMP CONTROLLER
J      DATA SET OF INTEREST
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
NUM1   FIRST INDEX TO BE TRANSFORMED
NUM2   LAST INDEX TO BE TRANSFORMED
R      STRESS RATIO (R = -1.0 IS DESIRED)
RSTR() STR() VALUES TRANSFORMED TO R = -1.0 PSI
STR() ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
TEST   Kt * Smax * (1 - R)/2 , TO BE COMPARED WITH FTY

C Kt IS ASSUMED TO BE ONE

DO 100 I = NUM1, NUM2
   TEST = STR(I,J) * (1.0 - R)/2.0
   IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST
   IF (TEST .GE. FTY) THEN
      RSTR(I,J) = TEST
      IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() =',RSTR(I,J)
   ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
      RSTR(I,J) = TEST/(1.0 - ((FTY - TEST)/FTU))
      IF (IOUT.EQ.10) WRITE(8,*)'2:RSTR() =',RSTR(I,J)
   ELSE
      RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
        /(2.0 * FTU)))
      IF (IOUT.EQ.10) WRITE(8,*)'3:RSTR() =',RSTR(I,J)
   END IF

7-61
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 X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87  COMMENTS: 13JUL89
C
C SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, & SY2, DD, SWHAT2, SUHAT2, NPPR)
C
C INPUTS: NUMREG, NSETS, NP, LNSTR, LNNF
C
C OUTPUTS: SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR
C
C IMPLICIT NONE
C
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), SXY(MAXREG), SY2(MAXREG)

C LIST OF VARIABLES
C
C BB() 1-D ARRAY CONTAINING SXY(L)/SY2(L) FOR EACH REGION
C DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C DIFFX() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L) AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C DIFFY() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L) AND MEANY(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
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 FOR THE BEST FIT LINE FOR EACH REGION
C SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X

7 - 62
REGRESSION FOR THE BEST FIT LINE FOR EACH REGION

SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(X = Ln S)
SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
EACH REGION (X = Ln S, Y = Ln N)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
(Y = Ln N)

C INITIALIZE ARRAYS

DO 50 L = 1, MAXREG
SX2(L) = 0.0
SXY(L) = 0.0
SY2(L) = 0.0
SWHAT2(L) = 0.0
SUHAT2(L) = 0.0
BB(L) = 0.0
DD(L) = 0.0
NPPR(L) = 0
50 CONTINUE

DO 60 J = 0, MAXSET
DO 70 K = 1, MAXDAT
DIFFY(K,J) = 0.0
DIFFX(K,J) = 0.0
70 CONTINUE
MEAN(J) = 0.0
70 CONTINUE

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

DO 100 L = 1, NUMREG

C FIRST CALCULATE SAMPLE X AND Y MEANS FOR DATA SET J IN REGION L
MEANY(J) = 0.0
MEANX(J) = 0.0
IF (IOUT .EQ. 10) WRITE(8,*)'L', L, ' J ', J,
& ' NP =', NP(J,L)
DO 250 K = 1, NP(J,L)
MEANY(J) = MEANY(J) + LNNF(K,J,L)
MEANX(J) = MEANX(J) + LNSTR(K,J,L)
IF (IOUT .EQ. 10) WRITE(8,*)'LNSTR =', LNSTR(K,J,L),
& ' LNNF =', LNNF(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)

C NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY, OF X AND Y FOR EACH REGION BY SUMMING OVER EACH DATA SET IN REGION L

DO 300 K = 1, NP(J,L)
DIFFY(K,J) = LNNF(K,J,L) - MEANY(J)
DIFFX(K,J) = LNSTR(K,J,L) - MEANX(J)
SX2(L) = SX2(L) + DIFFX(K,J) ** 2
SY2(L) = SY2(L) + DIFFY(K,J) ** 2
SXY(L) = SXY(L) + DIFFX(K,J) * DIFFY(K,J)
IF (IOUT .EQ. 10) THEN
WRITE(8,*)'K =', K, ' DIFFX(K,J) =', DIFFX(K,J),
& ' DIFFY(K,J) =', DIFFY(K,J),
& ' 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)
CONTINUE
IF (SXY(L) .LE. 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))
SKY(L) = SXY(L) / FLOAT(NPPR(L))

NOW CALCULATE THE RESIDUAL VARIANCES, SWHAT2, SUHAT2, FOR EACH
REGION FROM THE Y ON X AND X ON Y REGRESSIONS

DD(L) = SXY(L) / SX2(L)
BB(L) = SXY(L) / SY2(L)
IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'NPPR(L) = ', NPPR(L), ' SXY(L) = ', SXY(L),
' SX2(L) = ', SX2(L), ' SKY(L) = ', SKY(L), ' DD(L) = ', DD(L),
' BB(L) = ', BB(L)
ENDIF
DO 400 J = 0, NSETS
IF (IOUT .EQ. 10) WRITE(8,*), 'J = ', J, ' NP(J,L) = ', NP(J,L)
DO 500 K = 1, NP(J,L)
SWHAT2(L) = SWHAT2(L) + (DIFFY(K,J) - DD(L) * DFFX(K,J)) ** 2
SUHAT2(L) = SUHAT2(L) + (DFFX(K,J) - BB(L) * DFFY(K,J)) ** 2
IF (IOUT .EQ. 10) 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. 10) WRITE(8,*), 'NPPR(L) = ', NPPR(L),
'SWHAT2(L) = ', SWHAT2(L), ' SUHAT2(L) = ', SUHAT2(L)
100 CONTINUE
RETURN
END
C INPUTS: NUMREG, SX2, DD, SWHA121, S1UHAT2, NPPR
C OUTPUTS: IZERO, JZERO, MCHAT
C SUBPROGRAMS: TRNAT

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), SWHA121, S1UHAT2(MAXREG), SX,
& SX2(MAXREG), T, TO25(TTAB)

DATA (CHI025(I), I = 1, 75) /
& 0.00982069, 0.506356, 0.215795, 0.484419, 0.831211, 1.237347, 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, 10.25293, 10.9823, 11.6885, 12.4011, 13.1197, 13.8439, 14.5733, 15.3079, 16.0471, 16.7908, 17.53, 18.28, 19.04, 19.80, 20.56, 21.33, 22.10, 22.87, 23.63, 24.43, 25.21, 25.99, 26.76, 27.57, 28.38, 29.15, 29.95, 30.75, 31.55, 32.35, 33.16, 34.02, 34.81, 35.61, 36.41, 37.21, 38.02, 38.86, 39.66, 40.48, 41.30, 42.12, 42.95, 43.77, 44.60, 45.43, 46.26, 47.09, 47.92, 48.75, 49.59, 50.42, 51.26, 52.10, 52.94 /

DATA (CHI975(I), I = 1, 75) /
& 5.78, 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, 70.76, 71.64, 72.50, 73.36, 74.22, 75.06, 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, 89.83, 90.70, 91.57, 92.45, 93.32, 94.19, 95.05, 95.94, 96.82, 97.70, 98.57, 99.45, 100.33, 101.21, 102.09, 102.96, 103.86, 104.73, 105.61, 106.49, 107.37, 108.25, 109.14, 110.02, 110.90, 111.79, 112.67, 113.56, 114.44, 115.33, 116.21, 117.10, 117.98 /

DATA (CHI975(I), I = 1, 75) /
& 5.02389, 7.37776, 9.34840, 11.1433, 12.8325, 14.4494, 16.0128, 17.5348, 19.0228, 20.4831, 21.9200, 23.3877, 24.7556, 26.1900, 27.4884, 28.8454, 30.1910, 31.5264, 32.8523, 34.1696, 35.4789, 36.8070, 38.0757, 39.3451, 40.6457, 41.9232, 43.1944, 44.4607, 45.7222, 46.9792, 48.23, 49.48, 50.72, 51.96, 53.20, 54.44, 55.67, 56.90, 58.12, 59.3417, 60.56, 61.77, 62.99, 64.20, 65.42, 66.62, 67.82, 69.02, 70.22, 71.42, 72.61, 73.81, 75.00, 76.20, 77.38, 78.57, 79.75, 80.93, 82.12, 83.31, 84.48, 85.65, 86.83, 88.00, 89.18, 90.35, 91.52, 92.69, 93.86, 95.03, 96.19, 97.35, 98.52, 99.69, 100.84 /

DATA (CHI975(I), I = 1, 150) /
& 102.00, 103.16, 104.31, 105.47, 106.62, 107.78, 108.96, 110.09, 111.24, 112.39, 113.54, 114.69, 115.84, 117.00, 118.16, 119.28, 120.43, 121.57, 122.72, 123.88, 125.00, 126.14, 127.28, 128.42, 129.56, 130.70, 131.84, 132.98, 134.11, 135.25,
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

C VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE F(EJ_R_INGMANNER=

DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
& 2.201, 2.179, 2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
& 2.160, 2.145, 2.120, 2.110, 2.101,
& 2.093, 2.086, 2.074, 2.069, 2.064,
& 2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /

LIST OF VARIABLES

ARG
INTERMEDIATE CALCULATION VARIABLE

CHI025()
TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION

CHI975()
TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION

CHITAB
MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975

DD()
1-D ARRAY CONTAINING SX/L/SX2(L) FOR EACH REGION

I
CONTROLS LOOP FOR CHI025() AND CHI975()

IOUT
OUTPUT DUMP CONTROLLER

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

ZEROJ()
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
MCHAT(2,L) = SUHAT, THE ESTIMATE FOR 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)

NUM
EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS

NUMREG
NUMBER OF REGIONS OF INTEREST

SUHAT
EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS

SUHAT2()
1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
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
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

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
IZEROL(1,L) = 0.0
IZEROL(2,L) = 0.0
JZERO(1,L) = 0.0
JZERO(2,L) = 0.0
MCHAT(1,L) = 0.0
MCHAT(2,L) = 0.0
50 CONTINUE
C CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

DO 75 L = 1, NUMREG
   IF (NPPR(L) .GT. CHITAB) THEN
      WRITE(8,*) 'ERROR: EXCEEDED LIMIT ON DEGREES OF FREEDOM',
      & 'IN CHI-SQUARE TABLE, IN REGION ', L
      CALL TRMNAT
   ENDIF
75 CONTINUE

C ASSIGN VALUES TO NUM, T, SWHAT, SUHAT AND THEN CALCULATE
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
   CAL...
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)
C
C LIST OF VARIABLES
C ARG1 INTERMEDIATE CALCULATION VARIABLE
C ARG2 INTERMEDIATE CALCULATION VARIABLE
C CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C COEFFICIENT OF VARIATION, CO
C CZERO2 EQUAL TO CZERO ** 2
C IOUT OUTPUT DUMP CONTROLLER
C L CONTROLS DO LOOP FOR EACH REGION
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. -- MC(1,L) IS
C THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
C MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C MC() FOR EACH REGION
C NUMREG NUMBER OF REGIONS OF INTEREST
C SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C (X = Ln S)
C SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR
C EACH REGION (X = Ln S, Y = Ln N)
C SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C (Y = Ln N)
C
C INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
MCPNT(L) = 0
MC(1,L) = 0.0
MC(2,L) = 0.0
50 CONTINUE
C BEGIN CALCULATIONS
CZERO2 = CZERO ** 2
IF (IOUT .EQ. 10)
& WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2
DO 100 L = 1, NUMREG
ARG1 = SX2(L) - CZERO2
ARG2 = 0.0
IF (CZERO .EQ. 0.0) THEN
C THEN NO M CONSTRAINT IS REQUIRED
MCPNT(L) = 0
ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN
C THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M

7 - 68
MCPNT(L) = 1
MC(1,L) = - SY2(L) / (2.0 * SXY(L))

ELSE

THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
COMMON CALCULATIONS

ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)
IF (ARG2 .LT. 0.0) THEN
ARG2 IS NEGATIVE -- IMPLIES M IS COMPLEX
WRITE(8,*), 'ERROR: CO TOO LOW'
CALL TRNAT
ELSE
ARG2 = ARG2 ** 0.5
ENDIF

IF (SX2(L) .LT. CZERO2) THEN
AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M
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 = I, NUMREG
WRITE(8,*), 'L = ', L, ' MCPNT = ', MCPNT(L)
WRITE(8,*), 'ARG1 = ', ARG1, ' ARG1 = ', ARG2
WRITE(8,*), 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
200 CONTINUE
ENDIF

RETURN
END

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

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
MATCHR V4.1; V4.2; V4.3; V4.4; V4.5

SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
C INPUTS: NSETS, NP, NUMREG, LAMN, MCHAT
C OUTPUTS: PVAR
C IMPLICIT NONE

7-69
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),
  MAXREG    CONSTANT OVER REGIONS AND COMPONENTS
  MAXSET    MAXIMUM NUMBER OF REGIONS ALLOWED
  MCHAT()   2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
  NP()      FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
             MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
             MCHAT(2,L) = SUNAT, THE ESTIMATE FOR C
  NSETS     2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
  NUM()     SET IN EACH REGION
  NUMREG    NUMBER OF RELATED MATERIAL S/N DATA SETS ALLOWED
  PSIG2()   1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
  PVAR      VARIATION IN EACH REGION
  TOTAL     SUM OF RELATED MATERIAL S/N DATA SETS
  SUM()     EQUAL TO N(j-1) FOR EACH REGION WHERE N(j) IS THE SUM OF THE
             NUMBER OF POINTS IN EACH DATA SET
  TOTALNUM() NUMBER OF REGIONS OF INTEREST
  TOTALPSIG() NUMBER OF REGIONS OF INTEREST
  PSIG2()   1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
  TOTALSUM() VARIATION IN EACH REGION
  SUM()     CURVE WARRANTED BY THE AVAILABLE INFORMATION
  TOTALSUM() WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
             AVERAGE

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)
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 NAXREG
PARAMETER (MAXREG = 3)

COMMON IOUT
 INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
& MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES

IOUT  OUTPUT DUMP CONTROLLER
JZERO()  2-D ARRAY CONTAINING J0, THE 95% CONFIDENCE INTERVALS ON M
FOR EACH REGION
L  CONTROLS DO LOOP FOR EACH REGION
LOWER  LOWER BOUND OF INTERSECTION
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MC()  2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
-- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
BOUND
MCHAT()  2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
FOR EACH REGION -- MCHAT(1,L) = SUHAT, THE ESTIMATE FOR M AND MCHAT(2,L) = SHAT, 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),
    & '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 .LT. LOWER) THEN
    WRITE(8, *) 'ERROR: NO INTERSECTION BETWEEN J0 AND MC"
    CALL TRNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  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 .LT. LOWER) THEN
    WRITE(8, *) 'ERROR: NO INTERSECTION BETWEEN J0 AND MC"
    CALL TRNAT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF

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

ENDIF

100 CONTINUE
THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR

\[
\text{RANGEM}(1,L) = \text{MZERO}(1,L) \\
\text{RANGEM}(2,L) = 0.0
\]

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

\[
\begin{align*}
\text{LOWER} &= \text{AMAX1}(JZERO(1,L), \text{MZERO}(1,L)) \\
\text{UPPER} &= \text{AMIN1}(JZERO(2,L), \text{MZERO}(2,L))
\end{align*}
\]

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

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

\[
\begin{align*}
\text{LOWER} &= \text{AMAX1}(JZERO(1,L), \text{MZERO}(1,L), \text{MC}(1,L)) \\
\text{UPPER} &= \text{AMIN1}(JZERO(2,L), \text{MZERO}(2,L), \text{MC}(2,L))
\end{align*}
\]

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

ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN

THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO CO CONSTRAINT INTERSECT THESE TWO RANGES WITH J0

\[
\begin{align*}
\text{LOWER} &= \text{AMAX1}(JZERO(1,L), \text{MZERO}(1,L), \text{MC}(1,L)) \\
\text{UPPER} &= \text{AMIN1}(JZERO(2,L), \text{MZERO}(2,L), \text{MC}(2,L))
\end{align*}
\]

IF (UPPER .LT. LOWER) THEN
  WRITE(8,*) 'ERROR: NO INTERSECTION BETWEEN J0, M0, ',  \\
  'AND MC'
ENDIF

7 - 73
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,*), RANGEM(1,L) = RANGEM(1,L),
& WRITE(8,*), RANGEM(2,L) = RANGEM(2,L)
ENDIF
ELSE
WRITE(8,*), 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', 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(1,L))
 & .OR. (MCHAT(I,L) .GT. RANGEM(2,L)))
 & WRITE(8,*), 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
 & 'ON m IN REGION ', L
300 CONTINUE

RETURN
END

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

C SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C WITHOUT DATA
C PROGRAMMER: L. NEWLINS
C DATE: CODE: 2FEB88 FORMAT/COMMENTS: 12AUG91
 & V8.4, V8.5
C SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C
C INPUTS: RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C OUTPUTS: RANGEM, MCHAT, NUMREG
C
C IMPLICIT NONE
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT

7 - 74
INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG
REAL MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
LL EQUAL TO NUMREG FOR A SET OF CALCULATIONS
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
MCHAT(1,L) = dd(l), THE ESTIMATE FOR M AND
MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
IS UPPER BOUND
NNODAT NUMBER OF NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND

IF (IOUT .EQ. 10) WRITE(8,*), 'NUMREG = ', NUMREG
DO 100 L = 1, NNODAT
  NUMREG = NUMREG + 1
  LL = NUMREG
  IF (IOUT .EQ. 10) WRITE(8,*), 'L = ', L, ' NUMREG = ', NUMREG,
  & ' LL = ', LL, ' MPNT(LL) = ', MPNT(LL)
  IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
    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)
      WRITE(8,*), 'RANGEM(2,LL) = ', RANGEM(2,LL),
      & ' MZERO(2,LL) = ', MZERO(2,LL)
    ENDIF
  ELSE
    SPECIFY E(M) OF POSTERIOR FOR SAKE OF
    CALCULATIONS IN SUBROUTINE EXPCTD
    IF (RANGEM(2,LL) .EQ. 0.0) THEN
      MCHAT(1,LL) = RANGEM(1,LL)
    ELSE
      MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
    ENDIF
    IF (IOUT .EQ. 10) WRITE(8,*), 'MCHAT = ', MCHAT(1,LL)
    ELSE
      WRITE(8,*), 'ERROR: OVERALL PRIOR RANGE INCORRECTLY ',
      & 'SPECIFIED IN REGION WITHOUT DATA'
      CALL TRMNAT
    ENDIF
  ENDIF
100 CONTINUE
RETURN
END

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

C SUBROUTINE CONCAV ADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY

7 - 75
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) 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

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. 10) THEN
WRITE(8,*) 'RANGEM(1, L-1) = ', RANGEM(1, L-1),
& 'RANGEM(2, L-1) = ', RANGEM(2, L-1)
ENDIF
ENDIF
100 CONTINUE
RETURN
SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C INPUTS: NUMREG, RANGEM
C OUTPUT: MEDM
C
C IMPLICIT NONE
C
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
LOWER LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND

C INITIALIZE ARRAY MEDM
DO 50 L = 1, MAXREG
MEDM(L) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS FOR EACH REGION
DO 100 L = 1, NUMREG
IF (RANGEM(2,L) .EQ. 0.0) THEN
MEDM(L) = RANGEM(1,L)
ELSEIF (L .EQ. 1) THEN
WE ARE IN REGION ONE — NOT AFFECTED BY OTHER REGIONS
— MEDIAN WILL JUST BE AVERAGE OF RANGEM VALUES
MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
ELSE
MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT

7-77
LOWER M = \text{Amax1}(\text{RANGEM}(1,L), \text{MEDM}(L-1))
\text{MEDM}(L) = (\text{LOWER M} + \text{RANGEM}(2,L)) / 2.0
\text{ENDIF}

IF (IOUT .EQ. 10) THEN
\text{WRITE}(8,*), 'L =', L, ' NUMREG =', NUMREG
\text{WRITE}(8,*), 'RANGEM(1,L) =', RANGEM(1,L),
\text{WRITE}(8,*), ' RANGEM(2,L) =', RANGEM(2,L),
\text{WRITE}(8,*), 'LOWER M =', LOWER M, ' MEDM(L) =', MEDM(L)
\text{ENDIF}

100 \text{CONTINUE}
\text{RETURN}
\text{END}

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

\text{SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N}
\text{CURVE PARAMETERS}
\text{PROGRAMMER: L. NEWLIN}
\text{DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89}
\text{VERSION: MATCHR V8.3, V8.4, V8.5 MATGRM V4.3, V4.4, V4.5}
\text{Copyright (C) 1990, California Institute of Technology.}
\text{U.S. Government Sponsorship under NASA Contract NAS7-918}
\text{is acknowledged.}

\text{SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,}
\text{ZROREG, NBND, BIGK1, BZHAT)}
\text{INPUTS: NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND}
\text{OUTPUTS: BIGK1, BZHAT}
\text{SUBPROGRAMS: TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO}
\text{IMPLICIT NONE}
\text{INTEGER MAXDAT, MAXREG}
\text{PARAMETER (MAXDAT = 50, MAXREG = 3)}
\text{COMMON IOUT}
\text{INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG}
\text{REAL BIGK(0:MAXREG), BIGK1, BZHAT, FACTR, KHAT, MEANZ,}
\text{MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG),}
\text{NF(MAXDAT, MAXREG), SBND(0:MAXREG), STR(MAXDAT, MAXREG),}
\text{SZERO, TRBIGK(0:MAXREG), ZZ(MAXDAT)}

\text{LIST OF VARIABLES}
\text{BIGK()} 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR}
\text{EACH REGION}
\text{BIGK1 EQUAL TO BIGK(1)}
\text{BZHAT E(BETA0)}
\text{FACTR A SCALE FACTOR = PHI * KRATIO * Z}
\text{IOUT OUTPUT DUMP CONTROLLER}
\text{KHAT E(K)}
\text{L CONTROLS DO LOOP FOR EACH REGION}
\text{MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED}
\text{MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED}
\text{MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)}
\text{MEDM()} 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
\text{MM()} 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 containingrawnfn() (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
ZROREG Zero region -- values chosen to facilitate region do loop
BEGINning VALUE -- 0 = NO ZERO REGION EXISTS, 1 = NO ZERO REGION
ZZ() 1-D array containing transformed S-N data, z = F STR, NP, 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, K'HAT, BZHAT)
C CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION
   CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)
   BIGK = BIGK(L)
C CALCULATE BOUNDARIES OF STRESS REGIONS
   CALL FINDSB (NUMREG, ZROREG, PBND, 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, ZBND, TRBIGK,
      & FACTR, NUMREG)
   ENDIF
C WRITE RESULTS TO FILE
   IF (NCOMPS .EQ. 1) THEN
      WRITE(7,900) NUMREG, BZHAT, KHAT
   ENDIF
7 - 79
IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZHAT, KHAT

DO 200 L = ZROREG, NUMREG
   WRITE(7,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
   IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L), NBND(L), SBND(L)
200   CONTINUE
   WRITE(7,920)
ELSE
   WRITE(7,930) MM(1), BIGK(1), KHAT
ENDIF

FORMAT STATEMENTS
900 FORMAT ///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',///,2X,
   & 'NUMBER OF REGIONS:',I4,5X,'E(BETAO) =',F8.4,5X,'E(k) =',F8.4,///,2X,
   & 'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,/'
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',
   & ///,11X,'m',14X,'K',13X,'E(k)',///,7X,F8.5,5X,E12.5,6X,F7.4,/

RETURN
END

C SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 2JUN88 COMM: 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
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NUMREG, NPPR(MAXREG)
C REAL ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
   & MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
   & SUHAT2(MAXREG), SUHAT2(MAXREG), SX2(MAXREG)
C LIST OF VARIABLES
C ARG INTERMEDIATE CALCULATION VARIABLE
C DD() 1-D ARRAY CONTAINING SX(L)/SX2(L) FOR EACH REGION
C DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C  IOUT    SIG()  CALCULATION
C  OUTPUT DUMP CONTROLLER
C  CONTROLS DO LOOP FOR EACH REGION
C  MAXREG  MAXIMUM NUMBER OF REGION ALLOWED
C  MCHAT()  2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR
C            EACH REGION, BASED ON MATERIALS DATA ONLY — MCHAT(1,L) =
C            DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT,
C            THE ESTIMATE FOR C
C  MO()     1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C            MEAN FOR EACH REGION
C  MU()     1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C            DISTRIBUTION MEAN FOR EACH REGION
C  NUMREG   NUMBER OF REGIONS OF INTEREST
C  SIG()    1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C            DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C  SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C            VARIANCE FOR EACH REGION
C  SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C            REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C            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
C  INITIALIZE ARRAYS
C
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. IOUTDUMP) THEN
    WRITE(8,*), 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ', MCHAT(1,L)
    & WRITE(8,*), 'SUHAT2 = ', SUHAT2(L), ' MCHAT2 = '
    & WRITE(8,*), 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
    & WRITE(8,*), 'SUMX2 = ', SUMX2,
    & WRITE(8,*), 'DELTA = ', DELTA(L), ' ARG = ', ARG
    & WRITE(8,*), 'MO = ', MO(L), ' MU = ', MU(L),
    & WRITE(8,*), 'SIG2 = ', SIG(L), ' SIG = ', SIG(L)
  ENDIF
100 CONTINUE
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
SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C
C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT
C
C IMPLICIT NONE
C
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),
&s RANGEM(2, MAXREG), UPPER
C
LIST OF VARIABLES
C
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) = SDHAT, 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
C
DO 50 L = 1, MAXREG
 RANGEM(1,L) = 0.0
 RANGEM(2,L) = 0.0
50 CONTINUE
C
PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
DO 100 L = 1, NUMREG

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
  WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
ENDIF

IF (MPNT(L) .EQ. 1) THEN
  THERE IS A POINT PRIOR ON M -- THIS OVERrides ALL OTHER
  INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
  RANGEM(1,L) = MZERO(1,L)
  RANGEM(2,L) = 0.0
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
    WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)
    RANGEM(2,L) = RANGEM(2,L)
  ENDIF
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN
  THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT USE MO
  RANGEM(1,L) = MZERO(1,L)
  RANGEM(2,L) = MZERO(2,L)
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L)
    WRITE(8,*) 'MZERO(2,L) = ', MZERO(2,L)
    WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)
    WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L)
  ENDIF
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN
  THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO
  CONSTRAINT ADJUST THE LOWER BOUND OF MO BY MC
  LOWER = AMAX1(MZERO(1,L), MC(1,L))
  UPPER = AMIN1(MZERO(2,L), MC(2,L))
  IF (UPPER .LT. LOWER) THEN
    WRITE(8,*) 'ERROR: NO INTERSECTION BETWEEN MO AND MC'
    CALL TRMNT
  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
  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 TRMNT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
  WRITE(8,*), 'MZERO(2,L) = ', MZERO(2,L)
ENDIF
ELSE
  WRITE(8,*), 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
  CALL TRNSAT
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', L
  ENDIF
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 VERSION: MATCHR V7, V7.1, VS, VS.1, V8.2, V8.3, V8.4, V8.5
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
MCHEAT( )  2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
MCHET(1,L) = - DD(L), THE ESTIMATE FOR M AND
MCHET(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 MEZERO( ) 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 -- MEZERO(1,L) IS THE LOWER BOUND AND MEZERO(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 PRIOR NORMAL DISTRIBUTION VARIANCE 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,L) = MZERO(1,L)
  RANGEM(2,L) = MZERO(2,L)
  MU(LL) = MZERO(LL)
  SIG(LL) = SQRT(SIGMA2(LL))
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*)' RANGEM(1,L) =', RANGEM(1,L),
    &   ' RANGEM(2,L) =', RANGEM(2,L),
    &   ' MU(LL) =', MU(LL),
    &   ' SIG(LL) =', SIG(LL),
    &   ' SIGMA2(LL) =', SIGMA2(LL)
  ENDIF
ENDIF

SPECIFY E(M) OF POSTERIOR FOR SAKE OF CALCULATIONS IN SUBROUTINE EXPCTD

IF (RANGEM(2,L).EQ.0.0) THEN
  MCHET(1,L) = RANGEM(1,L)
  SIG(LL) = 0.0
ELSE
  MCHET(1,L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
ENDIF
IF (IOUT .EQ. 10) WRITE(8,*)' MCHET =', MCHET(1,L),
&   ' MU = ', MU(LL), ' SIG = ', SIG(LL)
ELSE
  WRITE(8,*)' ERROR: OVERALL PRIOR RANGE INCORRECTLY SPECIFIED IN REGION WITHOUT DATA'
ENDIF
100 CONTINUE
C SUBROUTINE PAREST CONTROLS THE CALCULATIONS FOR THE PARAMETER
C ESTIMATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89
C VERSION: MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
C
Copyright (c) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG,
& ZROREG, RAND, NBND, STR, BIGK, BZERO, MM,
& SBND)
C INPUTS: VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
C OUTPUTS: BIGK, BZERO, MM, SBND
C SUBPROGRAMS: FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB
C
C 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
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION MEAN FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
REGIONS OF INTEREST
NF() 2-D ARRAY CONTAINING RANNF() (CYCLES TO FAILURE) FOR THE
SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
NPTS() 1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
SPECIFIC MATERIAL S/N DATA SET
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND
RAND RANDOM NUMBER SEED
SBND() 1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
   CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
   REGION CONTAINED IN NBND()
SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
   STANDARD DEVIATION FOR EACH REGION
STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
   DATA SET BROKEN INTO REGIONS (PSI OR %)
SZ2 SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
   1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION;
   3 - TRUNCATED NORMAL VARIATION
ZROREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
   BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
ZZ() 1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
   Z = F(STR,NF,NBND,MM)

C OBTAIN THE VALUES OF M FOR EACH REGION
   IF (VARY .LE. 2) THEN
   C UNIFORM OR NO VARIATION IN M IS DESIRED
   CALL FINDM (RAND, NUMREG, RANGEM, MM)
   ELSE
   C NORMAL VARIATION IN M IS DESIRED
   CALL FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
   ENDIF
C TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)
   CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NF, ZZ)
C CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)
   CALL SMNVAR (NP, ZZ, MEANZ, SZ2)
C CALCULATE THE VALUES FOR K AND BETAO FROM THE SAMPLE MEAN
   C AND VARIANCE
   CALL KBETA (MEANZ, SZ2, K, BZERO)
C CALCULATE THE VALUE OF K FOR EACH REGION WHERE A .. K ** M
   CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)
C CALCULATE STRESS TIE-POINTS
   CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C WRITE RESULTS TO FILE
   C WRITE(7,900) NUMREG, BZERO
   C DO 200 L = ZROREG, NUMREG
   C WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
   C 200 CONTINUE
C WRITE(7,920)
C FORMAT STATEMENTS
900 FORMAT(///,2X,'SELECTED VALUES OF S/N CURVE PARAMETERS',
   & '/2X,'NUMBER OF REGIONS': ',I4.5X,'BETAO = ',F8.4,
   & '/2X,'REGION':7X,'m',15X,'K',9X,'LIFE BOUND',5X,
   & 'STRESS BOUND'//)
910 FORMAT(5X,11,5X,9.5,5X,E12.5,5X,'F9.3,6X,E11.5)
920 FORMAT(///)
RETURN
C** SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION BY **C** SAMPLING OFF THE APPROPRIATE M RANGE **C** PROGRAMMER: L. NEWLIN **C** DATE: CODE: 7JUN88 COMMENTS: 13JUL89 **C** VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5 **C** MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5 **C**

SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C INPUTS: RAND, NUMREG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: RANDOM, TRMNAT
C
C IMPLICIT NONE
C
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOA, L, NUMREG
REAL MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
DOUBLE PRECISION RAND

C LIST OF VARIABLES

IOUT
OUTPUT DUMP CONTROLLER
MAXREG
MAXIMUM NUMBER OF REGIONS ALLOWED
MM()
1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG
NUMBER OF REGIONS OF INTEREST
PICK()
1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RANGEM()
2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
   FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
   RANGEM(2,L) IS THE UPPER BOUND
X
UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED OFF THE RANGE ON M

C INITIALIZE MM()

DO 50 L = 0, MAXREG
      MM(MAXREG) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS

DO 100 L = 1, NUMREG

      PICK(1) = 0.0
      PICK(2) = 0.0

      IF (RANGEM(2,L) .EQ. 0.0) THEN
         M IS SPECIFIED AS A POINT VALUE
         MM(L) = RANGEM(1,L)
         IF (IOUT .EQ. 10) WRITE(*,*) '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

100 CONTINUE

7 - 88
WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
&
WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,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 TRA$NAT
ELSE
SAMPLE ON ADJUSTED RANGE
CALL RANDOM (X, RAND)
MM(L) = (PICK(2) - PICK(1)) * X + PICK(1)
ENDIF
IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'L = ', L, ', MM(L-1) = ', MM(L-1),
&
WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L)
&
WRITE(8,*), 'PICK(1) = ', PICK(1), ', PICK(2) = ', PICK(2)
&
WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,L), ' X = ', X,
&
ENDIF
ENDIF
100 CONTINUE
RETURN
END

C**********************************************************************
C SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
Miles, R. F., The RANDOM Computer Program: A Linear Congruential
Random Number Generator, JPL Publication 85-98, JPL Document
C
PROGRAMMER: L. GRONDALSKI, L. NEWLIN
DATE: 1DEC87
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
MATCHRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
V4.3, V4.4, V4.5
C**********************************************************************
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

7-89
C RANT INTERNAL CALCULATION
C RANX INTERNAL CALCULATION

C USING LCG RANDOM $ GENERATOR

RANA = 671093.0
RANC = 7090885.0
RANM = 33554432.0

10 RANX = RANA * RAND + RANC
RANDIV = RANX / RANM
RANT = DINT(RANDIV)
RANSUB = RANT * RANM
RAND = RANX - RANSUB
FRAC = SNGL(RAND / RANM)

IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
IF (IOUT .EQ. 2) WRITE(8,*)'RANX = ', RANX, ' RANDIV = ', RANDIV,
& ' RANT = ', RANT, ' RANSUB = ', RANSUB, ' RAND = ', RAND,
& ' FRAC = ', FRAC
RETURN
END

C NOTES: IOUT=2 DUMPS TO SCREEN

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

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

SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
C INPUTS: RAND, NUMREG, MU, SIG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: NORMGN, TRMNAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG),
& SIG(MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
MU() 1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
PICK() 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RAND RANDOM NUMBER SEED
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M

7-90
FOR EACH REGION -- RANGETM(1,L) IS THE LOWER BOUND AND
RANGETM(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
    M IS SPECIFIED AS A POINT VALUE
    MM(L) = RANGETM(1,L)
  ELSEIF (L .EQ. 1) THEN
    SAMPLE ON EXISTING RANGE
    CALL NORMGN (RAND, MU(L), SIG(L), X)
    IF ((X LT. RANGETM(1,L)) OR. (X GT. RANGETM(2,L))) GOTO 10
    MM(L) = X
    IF (IOUT .EQ. 10) THEN
      WRITE(8,*), 'RANGETM(1,L) = ', RANGETM(1,L),
      & ' RANGETM(2,L) = ', RANGETM(2,L), ' MM(L) = ', MM(L)
    ENDIF
    ELSE
      ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
      AND THEN SAMPLE
      PICK(1) = ANAX(MM(L-1), RANGETM(1,L))
      PICK(2) = RANGETM(2,L)
      IF (PICK(1) .GT. PICK(2)) THEN
        NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
        STOP PROGRAM
        ELSE
          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
    ENDIF
100 CONTINUE

RETURN
END

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

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

C SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA

7 - 91
The random variates are generated using the "Direct Method"

A. M. Abramowitz, and I. A. Stegun, 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)

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

UNIFORM(0,1) RANDOM VARIATE
OUTPUT DUMP CONTROLLER
MEAN OF NORMAL DISTRIBUTION
RANDOM NUMBER SEED
STANDARD DEVIATION OF NORMAL DISTRIBUTION
NORMAL RANDOM VARIATE
UNIFORM RANDOM NUMBER U(0,1)
NORMAL RANDOM NUMBER ON N(0,1)

IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA

CALL RANDOM (FRAC, RAND)
U1 = FRAC

CALL RANDOM (FRAC, RAND)
U2 = FRAC
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'U1 =', U1, ' U2 =', U2

Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)

X = SIGMA * Z1 + MU
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'Z1 =', Z1, ' Z2 =', Z2, ' X =', X

RETURN
END

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

C SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM
THE S/N DATA INTO THE VARIABLE Z = Ln(X)

SUBROUTINE TRNSFM

PROGRAMMER: L. NEWLIN
DATE: CODE: 7JUN88 COMMENTS: 13JUL89

7 - 92
SUBROUTINE TRANSM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

IMPLICIT NONE

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

COMMON IOUT
firstname
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
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
MM() 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,
2 = \text{F}(\text{STR}, \text{NF}, \text{NBND}, \text{MM})

C INITIALIZE VARIABLES

NP = 0

DO 50 I = 1, MAXDAT
ZZ(I) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS

DO 100 L = 1, NUMREG
MML = MM(L)
IF (IOUT .EQ. 10) WRITE(8, *)'L =', L, ' MM =', MML, ' 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-1))
& \times ((1.0 / MM(LL-1)) - (1.0 / MM(LL)))
IF (IOUT .EQ. 10) WRITE(8, *)'LL =', LL, ' NBND(LL-1) =',
& NBND(LL-1), ' MM(LL-1) =', MM(LL-1), ' MM(LL) =',

7 - 93
C SUBROUTINE SMNVAR CALCULATES THE Sample Mean and V a r i a n c e 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
    INTEGER MAXDAT
PARAMETER (MAXDAT = 50)
    COMMON IOUT
    INTEGER I, IOUT, NP
    REAL MEANZ, SZ2, ZZ(MAXDAT)

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 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

7 - 94
SZ2 = SZ2 + (ZZ(I) - MEANZ)**2
IF (IOUT .EQ. I0) WRITE(8,*)'I = ', I, ' SZ2 = ', SZ2
200 CONTINUE
SZ2 = SZ2 / FLOAT(NP - 1)
IF (IOUT .EQ. I0) WRITE(8,*)' SZ2 = ', SZ2
RETURN
END

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

C SUBROUTINE KBETA CALCULATES K AND BETAO FROM THE SAMPLE MEAN AND
C VARIANCE OF Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE : 6OCT87 COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C V8.4, V8.5
C
SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)
C
C INPUTS: MEANZ, SZ2
C OUTPUTS: K, BZERO
C IMPLICIT NONE
REAL PI
PARAMETER (PI = 3.1415926536)
COMMON IOUT
INTEGER IOUT
REAL BZERO, K, MEANZ, SZ, SZ2

LIST OF VARIABLES

BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE
SPECIFIC MATERIAL S/N DATA SET
IOUT OUTPUT DUMP CONTROLLER
K VALUE OF K -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL
DATA BASE
MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
PI SELF EXPLANATORY CONSTANT
SZ SZ2 ** 0.5
SZ2 SAMPLE VARIANCE OF THE TRANSFORMED DATA,
Z = F(STR, NF, NBND, MM)

C PERFORM CALCULATIONS

SZ = SZ2 ** 0.5
BZERO = PI / (SZ = (6.0 ** 0.5))
K = MEANZ

C DATA DUMP STATEMENTS

IF (IOUT .EQ. I0) THEN
WRITE(8,*) 'SZ2 = ', SZ2, ' SZ = ', SZ
WRITE(8,*) 'MEANZ = ', MEANZ, ' K = ', K, ' BZERO = ', BZERO
ENDIF
RETURN

7 - 95
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)

C LIST OF VARIABLES
BZERO -- VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING SPECIFIC MATERIAL DATA BASE
GAMMA -- EULER'S CONSTANT
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)
IF (IOUT .EQ. 10) WRITE(8,*) 'REGION =', L, ' K =', BIGK(1)
   & ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)

C CALCULATE K FOR REMAINING REGIONS
DO 100 L = 2, NUMREG
   BIGK(L) = BIGK(L-1) * NBND(L-1) /
   * ((1.0 / MM(L)) - (1.0 / MM(L-1)))
   WRITE(7,*) 'REGION =', 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******************************************************************************

C SUBROUTINE FINDSB CALCULATES THE REGION 'TIE-POINTS' -- THE STRESS
C VALUES WHICH CORRESPOND TO THE "LIFE BOUNDARIES" ACCORDING TO THE
C RANDOMLY SELECTED MS, AND THE Ks CALCULATED FROM THE BETA AND K
C CHARACTERIZING SPECIFIC MATERIAL
C
PROGRAMMER: L. NEWLIN
DATE: 22DEC88
VERSION: MATCHR V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C INPUTS: NUMREG, ZROREG, NBND, BIGK, MM
C OUTPUTS: SBND
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG, ZROREG
REAL BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG), &
& SBND (0:MAXREG)

LIST OF VARIABLES
BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
FOR EACH REGION
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
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 CHOSE 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
   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
    SBN(L) = BIGN(L) * NBN(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.

C SUBROUTINE WEIBGN (BETA, RAND, WEIB)
C
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

C 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
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: AUG91

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)

INPUTS: SZERO, BIGK, MM, NBND, TRSBND, FACTR

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), 
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
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, ' BIGK = ', TRBIGK(0)
WRITE (8,*) 'FACTOR = ', FACTR, ' BIGK1 = ', TRBIGK(1)
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)

INPUTS: S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO

OUTPUTS: GTLIFE

IMPLICIT NONE

INTEGER IOUT, L, MAXREG, NUMREG, ZROREG
PARAMETER (MAXREG= 3)
COMMON IOUT
REAL GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
& MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

LIST OF VARIABLES

GETLIF VALUE TO BE ASSIGNED TO GTLIFE -- CYCLES TO FAILURE FOR THE REQUIRED STRESS LEVEL
IOUT OUTPUT DUMP CONTROLLER
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LNA() 1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
LNZ NORMAL(0,PVAR) GENERATED RANDOM VARIATE
LPHIM() 1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE PHI IS A WEIBULL(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 TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER FLOWS
ZROREG ZERO REGION VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO REGION

GETLIF = 0.0

C CALCULATE CYCLES TO FAILURE

IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
GETLIF = 1.0
ELSE
DO 100 L = ZROREG, NUMREG
IF (S .GT. SBND(L)) THEN
    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

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

SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
M FOR EACH REGION
PROGRAMMER: L. NEWLIN
DATE: 10FEB88
Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.

SUBROUTINE SORTM (ALLM, NUMREG, NUM)

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
   5 INC = NUM
   10 IF (INC .GT. 1) THEN
       INC = INC / 2
   20 INORDR = .TRUE.

7 - 101
DO 300 I = 1, (NUM - INC)
   IF (ALLM(I, L) .GT. ALLM(I + INC, L)) THEN
      TEMP = ALLM(I, L)
      ALLM(I, L) = ALLM(I + INC, L)
      ALLM(I + INC, L) = TEMP
      INORDR = .FALSE.
   ENDIF
   CONTINUE

IF (.NOT. INORDR) GOTO 20
GOTO i0
ENDIF

400 CONTINUE
RETURN
END

C***********************************************************************
C FUNCTION ELWELD CONTROLS THE CALLS REQUIRED TO CALCULATE A LIFE FOR THE
C WELD AT AN ELBOW CASE (MODE 2)
C PROGRAMMER: L. NEWLIN
C DATE: 3MAY90
C VERSION: PIPE V8.1, V8.2, V8.3

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

**FUNCTION ELWELD (PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V,
& FTY, FTU, EM, K, PSUBI, TIB, TOB, DI, WOFF,
& LAMW, RB, WD, FK, RT, CYY, CCZ, CLY, CLZ, OVAL,
& LOCAT, MM, LNA, LPHIM, KRATTO, LNZ, SBND,
& SZERO, ZRORREG, NUNREG, STRHIS, NRAAN, PERIOD,
& TRUNC, ANGLE, CULPRT)***

**INPUTS:**
PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, FTY, FTU,
EM, K, PSUBI, TIB, TOB, DI, WOFF, LAMW, RB, WD, FK, RT,
CYY, CCZ, CLY, CLZ, OVAL, LOCAT, MM, LNA, LPHIM, KRATTO,
LNZ, SBND, SZERO, ZRORREG, NUNREG, STRHIS, NRAAN, PERIOD,
TRUNC, ANGLE

**OUTPUTS:**
ELWELD, CULPRT

**SUBPROGRAMS:** M2L1, M2L2, NARB1, TARNAT

**C**
**IMPLICIT** NONE
**COMMON** IOUT
**INTEGER** CULPRT, IOUT, LOCAT, MAXLD, MAXM, MAXREG, NLOAD, NRAN,
& NUNREG, ZRORREG
**REAL** PI
**PARAMETER** (MAXLD = 16, MAXM = 24000, MAXREG = 3,
& PI = 3.1415926536)

**REAL** ANGLE, CYY, CCZ, CLY, CLZ, DI, ELWELD, EM, FTY, FTU,
& FK(10), FTY, FTU, K(2, 2), KRATTO, LAMW, LIFE, LIFE1,
& LIFE2, LNA(0:MAXREG), INS, LPHIM(0:MAXREG), M(2, MAXL),
& MM(0:MAXREG), MSTAT(2), OVAL, P(MAXLD), PERIOD, PSTAT,
& PSUBI, RB, RT(10), SBND(0:MAXREG), STATIC(4),
& STRAMP(4, MAXLD), STRHIS(MAXLD, MAXM), SZERO, T(MAXLD),
& TIB, TOB, TRUNC, TSTAT, V(2, MAXLD), VSTAT(2), WD, WOFF

**C**
**LIST OF VARIABLES**
ANGLE

C

CCY

CCZ

CLZ

CULPRT

CLE

EM

FATLIF

FK()

FTU

IOUT

K()

KMAXR

KMAXM

KMAXLD

K()

LOCAT

LOC

LPH()

LPI

LNA(

LNZ

LPHI()

LOCAT

MAXLD

MAXM

MAXREG

M()

MSTAT()

NLOAD

NUM

OVAL

OVAL

P(

PSUBRI

PERIOD

PI

PI

PLSTAT

R1

RT()

SBND()

STATIC()

STRAMP()

STRHIS()

SZERO

T1

TIB

TOB

TRUNC

V()

VSTAT()

WD

woff

ZREG

ZREG

-- ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --

GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS

OUT-OF-PLANE CIRCUMFERENTIAL STRESS CARRYOVER FACTOR

IN-PLANE CIRCUMFERENTIAL STRESS CARRYOVER FACTOR

OUT-OF-PLANE AXIAL STRESS CARRYOVER FACTOR

LOCATION CAUSING FAILURE

INTERIOR DIAMETER

ELASTIC MODULUS

VALUE OF LIFE CALCULATED

1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS CONCENTRATION DUE TO WELD ECCENTRICITY

YIELD STRENGTH

ULTIMATE STRENGTH

OUTPUT DUMP CONTROLLER

FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR PIPE EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR PIPE EXTERIOR FOR HOOP DIRECTION; K(1,2) IS FOR PIPE INTERIOR FOR AXIAL DIRECTION; K(2,2) IS FOR PIPE INTERIOR FOR HOOP DIRECTION

RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS

ACCURACY FACTOR OF Fk - r/t CURVE

VALUE OF LIFE CAUSING FAILURE

VALUE OF LIFE AT LOCATION 1

VALUE OF LIFE AT LOCATION 2

1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION

NORMAL(0,PVAR) GENERATED RANDOM VARIATE

1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE

LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE PIPE AND 2 IS THE INTERIOR SURFACE OF THE PIPE

2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*) MAY BE THE M2 LOADS; M(2,*) ARE THE M3 LOADS

MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED

MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY

MAXIMUM NUMBER OF REGIONS ALLOWED

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

NUMBER OF TIME-VARYING LOADS

NUMBER OF POINTS IN STRESS-TIME HISTORY

NUMBER OF REGIONS OF INTEREST

OVALITY ANALYSIS ACCURACY FACTOR

1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS

INTERNAL PRESSURE

LENGTH OF TIME IN SECONDS FOR RANDOM STRESS-TIME HISTORY

CONSTANT EQUAL TO 3.14......

STATIC AXIAL LOAD

BEND RADIUS MEASURED FROM CENTER OF PIPE

1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS CONCENTRATION DUE TO WELD ECCENTRICITY

1-D ARRAY CONTAINING STRESS VALUES (PSI; R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CORRECTED FOR PHI, KRATIO, AND LNZ

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

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

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

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

STRESS TANGENT POINT (PSI)

1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS

WALL THICKNESS AT INNER BEND

WALL THICKNESS AT PIPE OUTER BEND

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

STATIC TORQUE LOAD

2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS

2-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V1 LOAD; V(2) IS THE V3 LOAD

WELD DISTANCE FROM ELBOW TANGENCY

WELD OFFSET

Zero Region -- VALUES CHosen TO FACILITATE REGION DO LOOP

7 - 103
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

IF ((LOCAT .EQ. 1) .OR. (LOCAT .EQ. 0)) THEN

EXTERIOR SURFACE OF THE PIPE
CALL M2LI (PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, &
      PSUBI, TIB, TOB, DI, EM, WOFF, LAMW, RB, K, WD, &
      CCY, CCZ, CLY, CLZ, OVAL, FK, RT, ANGLE, STATIC, &
      STRAMP)
CALL NARBNI (STRHIS, NRAN, PERIOD, TRUNC, STATIC, STRAMP, &
      NLOAD, FTY, FTU, 1.0, MM, LNA, LPHIM, KRATIO, &
      LNZ, SBND, ZZERO, ZR0REG, NUMREG, FATLIF)
LIFE1 = FATLIF
IF (IOUT .EQ. 25) WRITE(8,*) 'FATLIF = ', FATLIF
ENDIF

IF ((LOCAT .EQ. 2) .OR. (LOCAT .EQ. 0)) THEN

INTERIOR SURFACE OF THE PIPE
CALL M2L2 (PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, &
      PSUBI, TIB, TOB, DI, EM, WOFF, LAMW, RB, K, WD, &
      CCY, CCZ, CLY, CLZ, OVAL, FK, RT, ANGLE, STATIC, &
      STRAMP)
CALL NARBNI (STRHIS, NRAN, PERIOD, TRUNC, STATIC, STRAMP, &
      NLOAD, FTY, FTU, 1.0, MM, LNA, LPHIM, KRATIO, &
      LNZ, SBND, ZZERO, ZR0REG, NUMREG, FATLIF)
LIFE2 = FATLIF
IF (IOUT .EQ. 25) WRITE(8,*) 'FATLIF = ', FATLIF
ENDIF

IF ((LOCAT .LT. 0) .OR. (LOCAT .GT. 2)) THEN
  WRITE(8,*), 'ERROR: LOCATION INCORRECTLY SPECIFIED'
  CALL TRMNAT
ENDIF

IF (LOCAT .EQ. 0) THEN
LIFE = MIN (LIFE1, LIFE2)
IF (LIFE .EQ. LIFE1) THEN
  CULPR = 1
ELSE IF (LIFE .EQ. LIFE2) THEN
  CULPR = 2
ELSE
  WRITE(8,*) 'ERROR: CANNOT FIND CULPR LOCATION'
  CALL TRMNAT
ENDIF
ELWELD = LIFE
IF (IOUT .EQ. 25) WRITE(8,*), 'LIFE = ', LIFE, ' CULPR = ', CULPR
ELSE
  ELWELD = FATLIF
ENDIF

7 - 104
C SUBROUTINE M2L1 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
FOR MODE 2, LOCATION 1 (WELD NEAR AN ELBOW, EXTERIOR WALL OF THE PIPE)
C PROGRAMMER: L. NEWLIN
C DATE: 8JAN91
C VERSION: PIPE V8.3

SUBROUTINE M2L1(PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V,
& PSUBI, TIB, TOB, DI, EM, WOFF, LAMW, RB, K, WD,
& CCY, CCZ, CLY, CLZ, OVAL, FK, RT, ANGLE, STATIC,
& STRAMP)

INPUTS: PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, PSUBI,
& TIB, TOB, DI, EM, WOFF, LAMW, RB, K, WD, CCY, CCZ, CLY,
& CLZ, OVAL, FK, RT, ANGLE

OUTPUTS: STATIC, STRAMP

SUBPROGRAMS: CALCS

IMPLICIT NONE

COMMON I OUT
INTEGER IOUT, J, MAXLD, NLOAD
REAL NU, PI

PARAMETER (MAXLD = 16, NU = 0.30, PI = 3.1415926536)
REAL ANGLE, AREA, ARGNB, ARGNB2, ARGTM3, B, CPHI, CCY, CCZ,
& CLY, CLZ, DI, EM, FK(10), GCI, GCZ, GLY, GLZ, GNB1,
& GNO, GTMI, GTMO, IFK, K(2, 2), KOFF, KT1, KT2, L, L2,
& LAMW, M(2, MAXLD), MIIB, MIOB, MSTAT(2), NU2, OVAL,
& P(MAXLD), PSI, PSTAT, PSUBI, QO, QT, R, RB, RI, RM, RO,
& ROIB, ROOB, ROVER, ROT, RT(10), SIPHI, STATIC(4),
& STRAMP(4, MAXLD), T(MAXLD), TIB, TM, TOB, TSTAT,
& V(2, MAXLD), VSTAT(2), WD, WOFF, X1, X2, X3, X4

LIST OF VARIABLES

ANGLE ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --
GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
AREA CROSS SECTION AREA OF PIPE WALL
ARGNB, ARGNB2, ARGTM3 INTERMEDIATE CALCULATION VARIABLES USED IN OVALITY EFFECT
CALCULATIONS
B TORUS EFFECT
CPHI EQUAL TO COS(1*PHI)
CCY, CCZ, CLY, CLZ STRESS CARRY OVER FACTORS
DI INTERIOR DIAMETER
EM ELASTIC MODULUS
FK() 1-D ARRAY CONTAINING VALUES OF FK USED TO FIND STRESS
CONCENTRATION DUE TO WELD ECCENTRICITY
GCI, GCZ, GLY, GLZ OVALITY EFFECT COEFFICIENTS
GNB1, GNBO, GTMI, GTMO COEFFICIENTS USED IN OVALITY EFFECT CALCULATIONS
IFK INTERPOLATED VALUE OF FK CORRESPONDING TO PARTICULAR VALUE
& OF r/t
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DUE LOOP FOR EACH POINT IN RT() AND FK() DURING
INTERPOLATION
K() FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR PIPE
& EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR PIPE EXTERIOR
& FOR HOOP DIRECTION; K(1,2) IS FOR PIPE INTERIOR FOR AXIAL

RETURN
END

7 - 105
C

DIRECTION: K(2,2) IS FOR PIPE INTERIOR FOR HOOP DIRECTION
KOFF
STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
KT1
STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
KT2
STRESS CONCENTRATION FACTOR FOR HOOP STRESS
L
OVALITY EFFECT VARIABLE
L2
EQUAL TO L ** 2
LAMDA
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
MIIB
MOMENT OF INERTIA BASED ON MAXIMUM WALL THICKNESS (INNER BEND)
MIOB
MOMENT OF INERTIA BASED ON MINIMUM WALL THICKNESS (OUTER BEND)
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
NU2
EQUAL TO NU ** 2
OVAL
OVALITY ANALYSIS ACCURACY FACTOR
P()  
1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
PI
SELF EXPLANATORY CONSTANT
PSI
OVALITY EFFECT VARIABLE
PSTAT
STATIC AXIAL LOAD
PSUBI
INTERNAL PRESSURE
Q0
OVALITY EFFECT STRESS DECAY FACTOR
QT
TORUS EFFECT STRESS DECAY FACTOR
R
RADIUS OF INTEREST -- RO FOR EXTERIOR ANALYSIS; RI FOR INTERIOR ANALYSIS
RB
BEND RADIUS MEASURED FROM CENTER OF PIPE
RI
INTERIOR RADIUS
RM
MEAN RADIUS
RO
OUTER RADIUS
ROIB
OUTER RADIUS AT THE MAXIMUM WALL THICKNESS (INNER BEND)
ROOB
OUTER RADIUS AT THE MINIMUM WALL THICKNESS (OUTER BEND)
ROVERI
EQUAL TO THE MAXIMUM R / I
RVT
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
S1PHI
EQUAL TO SIN(1*PHI)
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
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
TIB
WALL THICKNESS AT INNER BEND
TM
MEAN WALL THICKNESS
TOB
WALL THICKNESS AT PIPE OUTER BEND
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
WD
WELD DISTANCE FROM ELBOW TANGENCY
WOFF
WELD OFFSET
X1, X2, X3, X4
OVALITY EFFECT VARIABLES

C
CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF THE WELD

NU2 = NU ** 2
RI = DI / 2.0
ROIB = RI + TIB
RO = ROIB
TM = (TOB + TIB) / 2.0
RM = RI + TM
R = ROIB
ROT = MAX (ROIB / TIB, ROOB / TOB)

DO 100 J = 2, 10

7 - 106
INTERPOLATE TO FIND FACTOR Fk CORRESPONDING TO VALUE OF \( r/t \)

\[
\text{IF} \quad \left( \text{ROT} \leq \text{RT}(\text{J}) \right) \quad \text{AND} \quad \left( \text{ROT} \geq \text{RT}(\text{J-1}) \right) \quad \text{THEN}
\]

\[
\text{IFK}(\text{J}) = \left( \text{FR}(\text{J}) - \text{FR}(\text{J-1}) \right) \times \left( \text{ROT} - \text{RT}(\text{J-1}) \right)
\]

\[
/ \left( \text{RT}(\text{J}) - \text{RT}(\text{J-1}) \right) + \text{FR}(\text{J-1})
\]

ENDIF

100 CONTINUE

KOFF = LAMW \times (1.0 + 3.0 \times \text{IFK} \times \text{WOFF})

CALCULATE THE CROSS-SECTIONAL AREA AND MOMENT OF INERTIA

\[
\text{AREA} = \pi \times (\text{ROOB}^2 - \text{RI}^2)
\]

\[
\text{MIIB} = \pi \times (\text{ROIB}^4 - \text{RI}^4) / 4.0
\]

\[
\text{MIOB} = \pi \times (\text{ROOB}^4 - \text{RI}^4) / 4.0
\]

ROVERI = \text{MAX} \left( \text{ROIB} / \text{MIIB}, \text{ROOB} / \text{MIOB} \right)

OBTAIN STRESS CONCENTRATION FACTORS AND RADII APPROPRIATE TO LOCATION

\[
\text{KT1} = K(1,1)
\]

\[
\text{KT2} = K(2,1)
\]

CALCULATE STRESS INCREASE DUE TO TORUS EFFECT

\[
\text{SI} = \sin(\text{ANGLE})
\]

\[
\text{CI} = \cos(\text{ANGLE})
\]

\[
\text{QT} = 1.0 - (\text{WD} / \text{RM})
\]

\[
\text{B} = 1.0 + \text{QT} \times \left( (2.0 \times \text{RB} + \text{RM} \times \text{SI}) / (2.0 \times (\text{RB} + \text{RM} \times \text{SI})) - 1.0 \right)
\]

CALCULATE STRESS INCREASE DUE TO OVALITY EFFECT

\[
\text{L} = \text{TM} \times \text{RB} / ((\text{RM} \times 2) \times \sqrt{1.0 - \text{NU}^2})
\]

\[
\text{PSI} = \text{PSUB} \times \text{RB} \times 2 / (\text{EM} \times \text{RM} \times \text{TM})
\]

\[
\text{IF} (\text{L} < 0.16) \quad \text{THEN}
\]

\[
\text{IF} (\text{L} < 0.16) \quad \text{WRITE}(8,*) \quad \text{‘WARNING: LAMBDA < .16 DURING OVALITY CALCULATIONS’}
\]

\[
L2 = L \times 2
\]

\[
X1 = 5.0 + 6.0 \times L2 + 24.0 \times \text{PSI}
\]

\[
X2 = 17.0 + 600.0 \times L2 + 480.0 \times \text{PSI}
\]

\[
X3 = X1 \times X2 - 6.25
\]

\[
X4 = (1.0 - \text{NU}^2) \times (X3 - 4.5 \times X2)
\]

\[
\text{ARGTM3} = 1.5 \times X2 - 18.75
\]

\[
\text{ARGNB} = L / X4
\]

\[
\text{ARGNB2} = 9.0 \times X2
\]

\[
\text{GTMI} = \text{SI} \times \text{ARGTM3} \times \sin(3.0 \times \text{ANGLE})
\]

\[
+ 11.25 \times \sin(5.0 \times \text{ANGLE}) / X4
\]

\[
\text{GTM} = \text{CI} \times \text{ARGTM3} \times \cos(3.0 \times \text{ANGLE})
\]

\[
+ 11.25 \times \cos(5.0 \times \text{ANGLE}) / X4
\]
GNBI = ARGNB * (ARGINB2 * COS (2.0 * ANGLE) + 225.0 * COS (4.0 * ANGLE))
& GNBO = ARGNB * (ARGINB2 * SIN (2.0 * ANGLE) + 225.0 * SIN (4.0 * ANGLE))

QQ = 1.0 - WD / (4.0 * RM)

GLZ = OVAL * (S1PHI + QQ * (CLZ * (GTMI + NU * GNBI) - S1PHI))
GCZ = OVAL * QQ * CCZ * (NU * GTMI + GNBI)
GLY = OVAL * (C1PHI + QQ * (CLY * (GTMO + NU * GNBO) - C1PHI))
GCY = OVAL * QQ * CCY * (NU * GTMO + GNBO)

IF (IOUT .EQ. 25) THEN
  WRITE(8,*,'(PSUBI = ', PSUBI, ', EM = ', EM)
  WRITE(8,*,'(L = ', L, ', PSI = ', PSI)
  WRITE(8,*,'(X1 = ', X1, ', X2 = ', X2)
  WRITE(8,*,'(X3 = ', X3, ', X4 = ', X4)
  WRITE(8,*,'(ARGTM3 = ', ARGTM3)
  WRITE(8,*,'(ARGINB = ', ARGNB)
  WRITE(8,*,'(ARGINB2 = ', ARGNB2)
  WRITE(8,*,'(GTM1 = ', GTMI)
  WRITE(8,*,'(GTM1 = ', GTMO)
  WRITE(8,*,'(GNBI = ', GNBI)
  WRITE(8,*,'(GNBI = ', GNBO)
  WRITE(8,*,'(OVAL = ', OVAL)
  WRITE(8,*,'(GLZ = ', GLZ)
  WRITE(8,*,'(GLY = ', GLY)
  ENDIF

CALL CALCS to CALCULATE the Stresses

CALL CALCS (PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, 
& PSUBI, R1, AREA, RO, R, KT1, KT2, ROVERI, KOFF, B 
& GLZ, GLY, GCY, ANGLE, STATIC, STRAMP)

IF (IOUT .EQ. 25) THEN
  WRITE(8,*,'(R1 = ', R1, ', AREA = ', AREA)
  WRITE(8,*,'(RO = ', RO, ', R = ', R)
  WRITE(8,*,'(KT1 = ', KT1, ', KT2 = ', KT2)
  WRITE(8,*,'(ROVERI = ', ROVERI, ', KOFF = ', KOFF)
  WRITE(8,*,'(B = ', B, ', GLZ = ', GLZ)
  WRITE(8,*,'(GLY = ', GLY)
  ENDIF

RETURN
END

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

C SUBROUTINE M2L2 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
C FOR MODE 2, LOCATION 2 (WELD NEAR AN ELBOW, INTERIOR WALL OF THE PIPE)
C PROGRAMMER: L. NEWLIN
C DATE: 8JAN91
C VERSION: PIPE V8.3

SUBROUTINE M2L2 (PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, 
& PSUBI, TIB, TOB, DI, EM, WOFF, LAMW, RB, K, WD, 
& GLZ, GLY, GCY, ANGLE, STATIC, STRAMP)

C INPUTS: PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, PSUBI, 
& TIB, TOB, DI, WOFF, EM, LAMW, RB, K, WD, CCY, CCZ, CLY, 
& CLZ, OVAL, FK, RT, ANGLE

C OUTPUTS: STATIC, STRAMP

C SUBPROGRAMS: CALCS

C IMPLICIT NONE

COMMON IOUT
INTEGER IOUT, J, MAXLD, NLOAD

7 - 108
REAL NU, PI

PARAMETER (MAXLD = 16, NU = 0.30, PI = 3.1415926536)

REAL ANGLE, AREA, ARGBN, ARGBN2, ARGTH3, B, CI1PHI, CCY, CCZ,
& CLY, CLZ, DI, EM, FK(10), GCY, GCLZ, GLY, GLZ, GNBI,
& GNB0, GTMI, GTM0, IFK, K(2, 2), KOFF, K1, K2, L, L2,
& LAMW, M(2, MAXLD), MI, MSTAT(2), NU, OVAL, P(MAXLD),
& PSI, PSTAT, PSUBI, Q0, QT, R, Rb, RI, RM, RO, ROIB,
& ROO8, ROJ, RT(10), S1PHI, STATIC(4), STRAMP(4, MAXLD),
& T(MAXLD), TIB, TM, TOB, TSTAT, V(2, MAXLD), VSTAT(2),
& WD, WOFF, X1, X2, X3, X4

LIST OF VARIABLES

ANGLE ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --
GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
AREA CROSS SECTION AREA OF PIPE WALL
ARGBN, ARGBN2, ARGTH3 INTERMEDIATE CALCULATION VARIABLES USED IN OVALITY EFFECT
CALCULATIONS
B TORUS EFFECT
CI1PHI EQUAL TO COS(1*PHI)
CCY, CCZ, CLY, CLZ STRESS CARRY OVER FACTORS
DI INTERIOR DIAMETER
EM ELASTIC MODULUS
FK() 1-D ARRAY CONTAINING VALUES OF FK USED TO FIND STRESS
CONCENTRATION DUE TO WELD ECCENTRICITY
GCY, GCLZ, GLY, GLZ OVALITY EFFECT COEFFICIENTS
GNBI, GNB0, GTMI, GTMO COEFFICIENTS USED IN OVALITY EFFECT CALCULATIONS
IFK INTERPOLATED VALUE OF FK CORRESPONDING TO PARTICULAR VALUE
OF R/T
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DUE LOOP FOR EACH POINT IN RT() AND FK() DURING
INTERPOLATION
K() FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR PIPE
EXTerior FOR AXIAL DIRECTION; K(2,1) IS FOR PIPE EXTERIOR
FOR HOOP DIRECTION; K(1,2) IS FOR PIPE INTERIOR FOR AXIAL
DIRECTION; K(2,2) IS FOR PIPE INTERIOR FOR HOOP DIRECTION
KOFF STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
K1, K2 STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
KSTRESS CONCENTRATION FACTOR FOR HOOP STRESS
L OVALITY EFFECT VARIABLE
L2 EQUAL TO L ** 2
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 MOMENT LOADS -- M(1) IS THE
M2 LOAD; M(2) IS THE M3 LOAD
MLOAD NUMBER OF TIME-VARYING LOADS
NU POISSON'S RATIO
NUZ EQUAL TO NU ** 2
OVAL OVALITY ANALYSIS ACCURACY FACTOR
P() 1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
PE SELF EXPLANATORY CONSTANT
PSI OVALITY EFFECT VARIABLE
PSTAT STATIC AXIAL LOAD
PSUBI INTERNAL PRESSURE
Q0 OVALITY EFFECT STRESS DECAY FACTOR
QT TORUS EFFECT STRESS DECAY FACTOR
RADIUS OF INTEREST -- RO FOR EXTERIOR ANALYSIS; RI FOR INTERIOR
ANALYSIS
RB BEND RADIUS MEASURED FROM CENTER OF PIPE
RI INTERIOR RADIUS
RM MEAN RADIUS
RO OUTER RADIUS
ROIB OUTER RADIUS AT THE MAXIMUM WALL THICKNESS (INNER BEND)
ROOB  OUTER RADIUS AT THE MINIMUM WALL THICKNESS (OUTER BEND)

R(T)  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

S1PHI  EQUAL TO SIN(1*PHI)

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

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

TIB  MEAN WALL THICKNESS AT INNER BEND

TOb  WALL THICKNESS AT PIPE OUTER BEND

TSSTAT  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

WDEF  WELD DISTANCE FROM ELBOW TANGENCY

WOFF  WELD OFFSET

X1, X2, X3, X4  OVALITY EFFECT VARIABLES

C  CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF THE WELD

NU2 = NU ** 2
RI = DI / 2.0
ROIB = RI + TIB
ROOB = RI + TOb
RO = ROIB
TM = (TOb + TIB) / 2.0
RM = RI + TM
R = RI
ROT = MAX (ROIB / TIB, ROOB / TOb)

DO 100 J = 2, 10

C  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)) / (RT(J) - RT(J-1)) * (ROT - RT(J-1))
ENDIF

100 CONTINUE

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

IF (IOUT .EQ. 25) THEN

WRITE(8,*) 'TIB = ', TIB, ' TOB = ', TOb
WRITE(8,*) 'DI = ', DI, ' RI = ', RI
WRITE(8,*) 'RM = ', RM, ' TM = ', TM
WRITE(8,*) 'RO = ', RO, ' R = ', R
WRITE(8,*) 'ROIB = ', ROIB, ' ROOB = ', ROOB
WRITE(8,*) 'NU2 = ', NU2, ' ROT = ', ROT
WRITE(8,*) 'IFK = ', IFK, ' WOFF = ', WOFF
WRITE(8,*) 'AREA = ', AREA, ' MI = ', MI
ENDIF

C  CALCULATE THE CROSS-SECTIONAL AREA AND MOMENT OF INERTIA

AREA = PI * (ROOB ** 2 - RI ** 2)
MI = PI * (ROOB ** 4 - RI ** 4) / 4.0

C  OBTAIN STRESS CONCENTRATION FACTORS AND RADII APPROPRIATE TO LOCATION

KT1 = K(1,2)
KT2 = K(2,2)

IF (IOUT .EQ. 25) THEN

WRITE(8,*) 'AREA = ', AREA, ' MI = ', MI

7 - 110
CALCULATE STRESS INCREASE DUE TO TORUS EFFECT

\[ \text{SIPHI} = \sin(\text{ANGLE}) \]
\[ \text{C1PHI} = \cos(\text{ANGLE}) \]
\[ Q = 1.0 - (\text{WD} / \text{RM}) \]
\[ B = 1.0 + Q \times ((2.0 \times \text{RB} + \text{RM} \times \text{SIPHI}) / (2.0 \times (\text{RB} + \text{RM} \times \text{SIPHI})) - 1.0) \]

IF (IOUT .EQ. 25) THEN
WRITE(8,*)
'RB = ', RB, ' WD = ', WD
WRITE(8,*)
'QT = ', QT, ' B = ', B
ENDIF

CALL CALCS to CALCULATE the Stresses

CALL CALCS (PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V,
& PSUBI, RI, AREA, RO, R, KT1, KT2, R/MI, KOFF, B,
& GLZ, GLY, GCZ, GCY, ANGLE, STATIC, STRAMP)

IF (IOUT .EQ. 25) THEN

7-111
SUBROUTINE CALCS (PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, & PSUBI, R, AREA, RO, R, KT1, KT2, ROVERI, KOFF, & B, GLZ, GLY, GCY, GCZ, ANGLE, STATIC, STRAMP)

C INPUTS: PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, PSUBI, R, AREA, RO, R, KT1, KT2, ROVERI, KOFF, B, GLZ, GLY, GCY, GCZ, ANGLE

C OUTPUTS: STATIC, STRAMP

C IMPLICIT NONE

COMMON IOUT

INTEGER I, IOUT, MAXLD, NLOAD
REAL PI
PARAMETER (MAXLD = 16, PI = 3.1415926536)
REAL ANGLE, AREA, ARGA, ARG, ARG2, B, GCY, GCZ, GLY, & GLZ, KOFF, KT1, KT2, MAX(2, MAXLD), MSTAT(2), P(MAXLD), & PSTAT, PSUBI, R, RI, RO, RO2, ROVERI, R2, STATIC(4), & STRAMP(4, MAXLD), T(MAXLD), TSTAT, V(2, MAXLD), VSTAT(2)

LIST OF VARIABLES

ANGLE ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION -- GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS

AREA CROSS SECTION AREA OF PIPE WALL

ARGA INTERMEDIATE CALCULATION VARIABLE

ARG INTERMEDIATE CALCULATION VARIABLE

ARG2 INTERMEDIATE CALCULATION VARIABLE

KOFF STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD

KT1 STRESS CONCENTRATION FACTOR FOR AXIAL STRESS

KT2 STRESS CONCENTRATION FACTOR FOR HOOP STRESS

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

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

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

PI SELF EXPLANATORY CONSTANT

PSTAT STATIC AXIAL LOAD
PERFORM INTERMEDIATE CALCULATIONS

R2 = R ** 2
RO2 = RO ** 2
ARG1 = RI ** 2 / (RO2 - RI ** 2)
ARG2 = (RO2 + R2) / R2
ARGA = PSUBI * ARG1
ARGB = ARGB * ARGA

IF (IOUT .EQ. 25) THEN
  WRITE(8,*) 'R = ', R, ' R, ' R2 = ', R2,
  ' RO = ', RO, ' RO, ' RO2 = ', RO2
  WRITE(8,*) 'RI = ', RI, ' AREA = ', AREA
  WRITE(8,*) 'ARG1 = ', ARG1, ' ARG1, ' ARG2 = ', ARG2
  WRITE(8,*) 'ARGA = ', ARGA, ' ARGB = ', ARGB
ENDIF

CALCULATE STATIC STRESS COMPONENTS

STATIC(1) = KT1 * (PSTAT / AREA + (GLZ * MSTAT(2) + GLY * MSTAT(1)) * ROVERI + ARGA) * KOFF
STATIC(2) = KT2 * (GC1 * M(2,I) + GGY * M(I,I)) * ROVERI
STATIC(3) = ARGA * (RO2 - R2) / R2
STATIC(4) = (TSTAT * ROVERI / 2.0) - 2.0 * (VSTAT(1) * COS (ANGLE) + V(2,I) * SIN (ANGLE)) / AREA

CALCULATE TIME-VARYING STRESS COMPONENTS

DO 100 I = 1, NLOAD
  STRAMP(1,I) = KT1 * (P(I) / AREA + (GLZ * M(2,I) + GLY * M(I,I)) * ROVERI) * ROFF
  STRAMP(2,I) = KT2 * (GC1 * M(2,I) + GGY * M(I,I)) * ROVERI
  STRAMP(3,I) = 0.0
  STRAMP(4,I) = (T(I) * ROVERI / 2.0) - 2.0 * (V(1,I) * COS (ANGLE) + V(2,I) * SIN (ANGLE)) / AREA
100 CONTINUE

IF (IOUT .EQ. 25) THEN
  WRITE(8,*) 'I AXIAL HOOP RADIAL SHEAR'
  WRITE(8,*) STATIC(1), STATIC(2), STATIC(3), STATIC(4)
DO 300 I = 1, NLOAD
  WRITE(8,*) STRAMP(1,I), STRAMP(2,I), STRAMP(3,I), STRAMP(4,I)
300 CONTINUE
ENDIF
C SUBROUTINE NARBN1 CALCULATES THE FATIGUE LIFE WHEN A RANDOMLY DISTRIBUTED LOAD IS PRESENT USING A SIMULATED NARROW BAND STRESS-TIME HISTORY
C PROGRAMMER: L. NEWLIN
C DATE: 3MAY80
C VERSION: 1.5 (PIPE V8.1, V8.2, V8.3)

SUBROUTINE NARBN1 (STRHIS, M, PERIOD, TRUNC, STATIC, STRAMP,
& NLOAD, PTY, FTU, KT, MM, LNA, LPHIM, KRATIO,
& LNZ, SBND, SZERO, ZROREG, NUMREG, FATLIF)

INPUTS: STRHIS, M, PERIOD, TRUNC, STATIC, STRAMP, NLOAD, PTY,
& FTU, KT, MM, LNA, LPHIM, KRATIO, LNZ, SBND, SZERO,
& ZROREG, NUMREG

OUTPUTS: FATLIF

SUBPROGRAMS: RAINFI

IMPLICIT NONE
COMMON IOUT
INTEGER I, IO_T, J, M, MAXLD, MAXM, MAXREG, NLOAD, NUMREG,
& ZROREG
PARAMETER (MAXLD = 16, MAXM = 24000, MAXREG = 3)
REAL FATLIF, PTY, FTU, KRATIO, KT, LNA(0:MAXREG), LNZ,
& LPHIM(0:MAXREG), MM(0:MAXREG), PERIOD, RAINFI,
& S(4, MAXM), SBND(0:MAXREG), SEFF(MAXM), STATIC(4),
& STRAMP(4, MAXLD), STRHIS(MAXLD, MAXM), SZERO, TRUNC

LIST OF VARIABLES

FATLIF VALUE OF FATIGUE LIFE CALCULATED
PTY YIELD STRENGTH
FTU ULTIMATE STRENGTH
IO_T CONTROLS DO LOOP FOR RANDOM AND SUPERIMPOSED SINE LOADS
J OUT CONTROLS DO LOOP FOR EACH POINT IN THE STRESS-TIME HISTORY
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
KT STRESS CONCENTRATION FACTOR
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(ETAO, ETAO) GENERATED RANDOM VARIATE
M NUMBER OF POINTS IN STRESS-TIME HISTORY
MAXLD MAXIMUM NUMBER OF TIME-VARYING LOADS
MAXM MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING VALUES FOR M FOR EACH REGION
NLOAD NUMBER OF TIME-VARYING LOADS
NUMREG NUMBER OF REGIONS OF INTEREST
PERIOD TIME IN SECONDS FOR ONE PERIOD OF STRESS-TIME HISTORY
RAINFI FUNCTION WHICH CALCULATES THE TIME TO FAILURE FOR A GIVEN
& UNI-AXIAL STRESS-TIME HISTORY
S() 2-D ARRAY CONTAINING THE TOTAL COMPONENT STRESS-TIME HISTORIES
SEFF() 1-D ARRAY CONTAINING THE EFFECTIVE (OR UNI-AXIAL) STRESS-TIME
& HISTORY RESULTING FROM THE COMBINATION OF STATIC, RANDOM, AND
& SINUSOIDAL LOADS FOR ALL FOUR COMPONENTS
SBND() 1-D ARRAY CONTAINING STRESS VALUES (PSI R = -1.0)
& CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
& REGION CORRECTED FOR PHI, KRATIO, AND LNZ
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
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

STRHIS()  2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING
STRESS-TIME HISTORIES
SZERO  STRESS TENSILE POINT (PSI)
TRUNC  VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
ZROREG  ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

DO 50 J = 1, M
   S(1,J) = STATIC(1)
   S(2,J) = STATIC(2)
   S(3,J) = STATIC(3)
   S(4,J) = STATIC(4)
50 CONTINUE
DO 100 I = 1, NLOAD
   DO 150 J = 1, M
      S(1,J) = S(1,J) + STRHIS(I,J) * STRAMP(1,I)
      S(2,J) = S(2,J) + STRHIS(I,J) * STRAMP(2,I)
      S(3,J) = S(3,J) + STRHIS(I,J) * STRAMP(3,I)
      S(4,J) = S(4,J) + STRHIS(I,J) * STRAMP(4,I)
150 CONTINUE
100 CONTINUE
DO 300 J = 1, M
   SEFF(J) = (S(1,J) / ABS(S(1,J))) * SORT((
     {S(1,J) - S(2,J)} ** 2 + {S(1,J) - S(3,J)} ** 2 +
     {S(2,J) - S(3,J)} ** 2 + {6.0 * S(4,J) ** 2}
   ) / 2.0)
300 CONTINUE
IF (IOUT .EQ. 25) THEN
   DO 125 J = 1, M
      WRITE(8,*) J, 'S: ', S(1,J), S(2,J), S(3,J), S(4,J)
   END DO
   WRITE(8,*) 'SEFF = ', SEFF(J)
125 CONTINUE
ENDIF

FATLIF = RAINFI (SEFF, M, TRUNC, PERIOD, KT, FTU, FTY, MM, LNA,
   & LPHIM, KRATIO, LNZ, SBND, SZERO, ZROREG, NUMREG)
IF (IOUT .EQ. 25) WRITE(8,*) 'PERIOD = ', PERIOD,
   & 'FATLIF = ', FATLIF

RETURN

C******************************************************************************
C******************************************************************************
FUNCTION RAINFI CALCULATES THE TIME (in seconds) TO FAILURE FOR
THE GIVEN UNI-AXIAL (OR EFFECTIVE) STRESS-TIME HISTORY
PROGRAMMER: L. NEWLIN
DATE: 23MAY89
VERSION: 3.1+ (MATCHR V8.3, V8.4, V8.5)
Copyright (C) 1990, California Institute of Technology,
U.S. Government Sponsorship under NASA Contract NAS7-918

7 - 115
C is acknowledged.

FUNCTION RAINFI (SEFF, M, TRUNC, PERIOD, KT, FTU, FTY, MM, LNA, LPHIM, KRATIO, LNZ, SBND, SZERO, ZROREG, NUMREG)

INPUTS: SEFF, M, TRUNC, PERIOD, KT, FTU, FTY, MM, LNA, LPHIM, KRATIO, LNZ, SBND, SZERO, ZROREG, NUMREG

OUTPUTS: RAINFI

IMPLICIT NONE

COMMON IOUT
COMMON / COUNT / TOHIGH

INTEGER MAXREG, MAXM

PARAMETER (MAXREG = 3, MAXM = 24000)

INTEGER BIG1, I, INDEX(MAXM), IOUT, J, JMAX, K, M, N, NEWTOT, NUMREG, TOHIGH, ZROREG

REAL ARGM, CHKFT, E(MAXM), FTU, FTY, GTLIFE, INVLIF(MAXM), KRATIO, KT, LIFE(MAXM), LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MM(0:MAXREG), PERIOD, PGETSM, RAINFI, S(MAXM), SALTF(MAXM), SBND(0:MAXREG), SEFF(MAXM), SEFFM(2, MAXM), SEFMAX, SM, SMEANF(MAXM), SP(MAXM), STR(MAXM), SUMDAM, SZERO, TEST1(MAXM), TEST2(MAXM), TRUNC

LIST OF VARIABLES

RAINFI CYCLES TO FAILURE FOR THE GIVEN STRESS LEVELS

input variables:

SEFF(M) EFFECTIVE STRESSES BEFORE FILTERING/RAINFLOW
M TOTAL NUMBER OF STRESS DATA POINTS PER PERIOD
TRUNC VALUE USED TO FILTER OUT NOISE
PERIOD TIME IN SECONDS FOR ONE PERIOD
KT FATIGUE CONCENTRATION FACTOR
FTU ULTIMATE TENSILE STRENGTH (PSI)
FTY YIELD TENSILE STRENGTH (PSI)

intermediate variables:

SEFMAX LARGEST EFFECTIVE STRESS
JMAX INDEX (LOCATION) OF SEFMAX IN SEFF()
I,J,K COUNTERS FOR VARIOUS DO LOOPS
SP(M+1) RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
S(MAXM) FILTERED EFFECTIVE STRESSES
NEWTOT TOTAL NUMBER OF EFFECTIVE STRESS 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 STRESSES AFTER RESEQUENCING/FILTERING/RAINFLOW
SEFFM(1,I) = sigma max,eff,i
SEFFM(2,I) = sigma min,eff,i
SALTF(I) = sigma alternating,eff,i
SMEANF(I) = sigma mean,eff,i
BIG1 VALUE OF I FOR SEFMAX
SM = EQUIVALENT MEAN STRESS
ARGM INTERMEDIATE CALCULATION VARIABLE EQUAL TO KT/(1 - SM/FTU)
STR(N) STR(I) = EQUIVALENT (COMBINED) STRESS;
LIFE(N) LIFE(I) = CALCULATED LIFE FOR STRESS LEVEL STR(I)
INVLIF(N) INVLIF(I) = 1/LIFE(I); DAMAGE FRACTION
SUMDAM SUM OF ALL THE DAMAGE FRACTIONS

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
C MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
C MM()  1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C NUMREG  NUMBER OF REGIONS OF INTEREST
C SBND()  1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
C CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C REGION CONTAINED IN NBND() CORRECTED BY PHI, KRATIO,
C AND LNZ
C SZERO  STRESS TENSILE POINT (PSI)
C ZROREG  ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C BEGINNING VALUE -- 0 -- ZERO REGION EXISTS, 1 -- NO ZERO
C REGION
C dump input data
C if (iout.eq.20) then
C write(8,'(8,*)','rainfl inputs',m,period)
C write(8,'(8,*)','kt',kt,'ftu',ftu,fty)
C write(8,'(8,*)','numreg',numreg,'zroreg',zroreg)
C write(8,'(8,*)','lna(i),mm(i),iphim(i),sbnd(i),i=zroreg,numreg)
C endif
C INITIALIZE ARRAYS
C DO 50 I = 1, MAXM
C SP(I) = 0.0
C S(I) = 0.0
C E(I) = 0.0
C SEFFM(1,I) = 0.0
C SEFFM(2,I) = 0.0
C SALTF(I) = 0.0
C SMEANF(I) = 0.0
C STR(I) = 0.0
C LIFE(I) = 0.0
C INVLIF(I) = 0.0
C INDEX(I) = 0
C TEST1(I) = 0.0
C TEST2(I) = 0.0
C 50 CONTINUE
C SM = 0.0
C TOHIGH = 0
C*************** BEGIN RESEQUENCE ***************
C RESEQUENCE effective stresses (needed for rainflow analysis);
C largest effective stress is placed at beginning and end of SP(M+1)
C find SEFMAX, the largest sigma,eff, and JMAX, its location within SEFF(M)
C SEFMAX = -I.0E+20
C DO 200 I = 1, M
C IF (SEFF(I) .GT. SEFMAX ) THEN
C SEFMAX = SEFF(I)
C JMAX = I
C ENDIF
C 200 CONTINUE
C assign all points from JMAX out, to the beginning of SP()
C DO 210 I = 1, M-JMAX+1
C J = JMAX-1 + I
C SP(I) = SEFF(J)
C 210 CONTINUE
C assign points before JMAX to the end of SP()
C J = 0
C DO 220 I = M-JMAX+2, M
C J = J + 1
C SP(I) = SEFF(J)
C 220 CONTINUE
C SP(M+1) = SEFF(JMAX)
C if (iout.eq.20) then

7 - 117
write(8,*)'sefmax=',sefmax,'jmax',jmax
write(8,*)'sp(m+1):',(sp(i),i=1,m+1)
endif

C******************************************************** END RESEQUENCE ********************************************************

C******************************************************** BEGIN FILTER ********************************************************

C FILTER the resequenced effective stresses, 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

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), ' test2 = ', test2(i)
  endif

K = 1
INDEX(1) = 1
DO 310 I = 2, M
  IF ((TEST1(I) .NE. 0) .AND. (TEST2(I) .LE. 0)) THEN
    K = K + 1
    INDEX(K) = I
  ENDIF
310 CONTINUE

NEWTOT = K + 1
INDEX(NEWTOT) = M + 1
DO 320 I = 1, NEWTOT
  S(I) = SP(K)
320 CONTINUE

if (iout.eq.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 stress data,
C S(NEWTOT); places each cycle’s max and min values into SEFFM(2,N)
C counters: I counts # of cycles found, J counts how many S()’s counted,
C K accumulates unmatched points

I = 0
J = 0
K = 0
400 CONTINUE

J = J+1
K = K+1

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
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
    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
400 CONTINUE

E(K-2) = E(K)  
K = K-2  
c return for more counting  
GOTO 410  

499 continue  
C N equals the final number of cycles found  
N = I  
if (iout.eq.20) then  
write(8,'*')'N : ',n  
write(8,'*')'seffm(2,n):'  
do 12 i=1,n  
write(8,'*') seffm(1,i), seffm(2,i)  
12 continue  
endif  

if (N .EQ. 0) THEN  
C truncation filter value too large -- no cycles left  
SUMDAM = 1.0E-36  
GOTO 710  
ENDIF  

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

C calculate alternating and mean effective stresses  
C  
DO 500 I=1,N  
SALTF(I) = ( SEFFM(1,I) - SEFFM(2,I) ) / 2.0  
SMEANF(I) = ( SEFFM(1,I) + SEFFM(2,I) ) / 2.0  
500 continue  

if (iout.eq.20) write(8,'*')'saltf(n) : ',(saltf(i),i=1,n)  
write(8,'*')'smeanf(n) :',(smeanf(1),i=1,n)  

C***** Determine Equivalent Mean Stress, SM(N), (two methods) *****  
C  
BIG1 = N  
SM = PGETSM(SALTF(BIG1), SMEANF(BIG1), FTY, KT)  
if (iout.eq.20) write(8,'*')'sm : ', sm  

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

C calculate equivalent stresses, STR(N)  
C  
ARGM = KT / (1.0 - SM / FTU)  
DO 530 I=1,N  
STR(I) = SALTF(I) * ARGM  
IF (STR(I) .GE. FTU) TOHIGH = TOHIGH + 1  
530 continue  
if (iout.eq.20) write(8,'*')'str(n) :',(str(i),i=1,n)  

C calculate lives and damage fractions: LIFE(N) and INVLIF(N)  
C  
DO 600 I=1,N  
LIFE(I) = GTLIFE (STR(I), MH, LNA, LPHIM, KRATIO, LNZ, SBND,  
& ZZOREG, NUMREG, SZERO)  
600 continue  
DO 650 I=1,N  
INVLIF(I) = 1.0 / LIFE(I)  
650 continue  
if (iout.eq.20) then  
do 14 i=1,n  
write(8,'*')'life(n) : ',life(i),''  
invlif(n) : ',invlif(i)  
14 continue  

7 - 119
endif

C Miner's Rule -- sum the damage fractions

C
SUMDAM = 0.0
DO 700 I=1,N
SUMDAM = SUMDAM + INVLI(I)
700 CONTINUE
710 CONTINUE
if (iout.eq.20) write(8,*) 'sumdam: ',sumdam

C calculate fatigue life (time to failure) in seconds
C
RAINF1 = PERIOD / SUMDAM
if (iout.eq.15) then
chkft = period/sumdam
write(8,*) ' rainfl life',chkft
write(8,*)
endif
RETURN
END

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

FUNCTION PGETSM (SALT, SMEAN, FTY, KT)
C SM is the Equivalent Mean Stress
C
IMPLICIT NONE
REAL FTY, KT, PGETSM, SALT, SMEAN, ST, SX
ST = KT*(SALT+SMEAN)
IF (ST.GT.FTY) THEN
SX = KT*SALT
IF (SX.GT.FTY) THEN
PGETSM = 0.0
ELSE
PGETSM = FTY-SX
ENDIF
ELSE
PGETSM = SMEAN
ENDIF
RETURN
END
7.1.2 HEXHCF Program

7.1.2.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for HEXHCF using Uniform variation on the materials shape parameter \( m \) is given in Figure 7-3, while the tree structure for the truncated Normal case is given in Figure 7-4. 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.

7.1.2.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7-3. The section numbers where the subprograms are described by means of flowcharts are given next to the names.

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDREG</td>
<td>4.1.3.9</td>
<td>Adds the ( m ) ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>4.1.3.15</td>
<td>Adds the ( m ) ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.</td>
</tr>
<tr>
<td>BETAGN</td>
<td>4.4.5</td>
<td>Generates Beta(( a, b, p, \theta )) random variates.</td>
</tr>
<tr>
<td>CONCAV</td>
<td>4.1.3.10</td>
<td>Adjusts the upper bound of the posterior ranges on ( m ) to be consistent with concavity constraints.</td>
</tr>
<tr>
<td>CONVRT</td>
<td>4.1.3.3</td>
<td>Transforms stress data to equivalent zero-mean stresses with stress ratio of ( -1.0 ).</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>4.1.3.12</td>
<td>Calculates the median S/N curve parameters from the results of the information aggregation calculations.</td>
</tr>
<tr>
<td>FINDK</td>
<td>4.1.5.6</td>
<td>Calculates the value of the location parameter ( K ) (where ( A = K^m )) for each life region by using Equations 2-37 and 2-41.</td>
</tr>
<tr>
<td>FINDM</td>
<td>4.1.5.1</td>
<td>Obtains the value of ( m ) for each life region by adjusting the range (to ensure concavity) and then sampling from the Uniform distribution over the appropriate ( m ) range.</td>
</tr>
<tr>
<td>FINDMC</td>
<td>4.1.3.5</td>
<td>Calculates the ( m ) range for each life region implied by the constraint on the coefficient of variation of fatigue strength ( C ) by using Equations 2-28 through 2-32.</td>
</tr>
<tr>
<td>FINDMN</td>
<td>4.1.5.2</td>
<td>Obtains the value of ( m ) for each life region by sampling from the appropriate truncated Normal distribution on ( m ).</td>
</tr>
<tr>
<td>FINDSB</td>
<td>4.1.5.7</td>
<td>Calculates the life region &quot;tie-points&quot; or stress values which correspond to the &quot;life boundaries&quot; conditional on the randomly...</td>
</tr>
</tbody>
</table>
Figure 7-3 Tree Structure for Program HEXHCF for the Uniform Variation in Materials Shape Parameter $m$
Figure 7-4  Tree Structure for Program HEXHCF for the Truncated Normal Variation in Materials Shape Parameter $m$
Table 7-3 List of Subprograms for Program HEXHCF (Cont’d)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FNDRNG⁷</td>
<td>4.1.3.8</td>
<td>Combines the 95% confidence interval, ( J_0 ), with the implicit and explicit constraints on ( m ) to obtain posterior credibility ranges on ( m ) for each life region.</td>
</tr>
<tr>
<td>GAM</td>
<td>4.4.4</td>
<td>Generates Gamma(( \alpha ), 1) random variates.</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.</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>4.1.3.7</td>
<td>Calculates ( \sigma^2 ), Equation 2-49, the extent of departures from the multiple heat median S/N curve warranted by the information available.</td>
</tr>
<tr>
<td>HEXHCF</td>
<td>5.1.3.1</td>
<td>The main routine that controls the logical flow of the high cycle fatigue plain welded duct program with thermal loads.</td>
</tr>
<tr>
<td>INFAGG⁸</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>5.1.3.6</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 and 2-26.</td>
</tr>
<tr>
<td>KBETA</td>
<td>4.1.5.5</td>
<td>Calculates ( k ) and ( \beta_o ) from the sample mean and variance of ( Z ), where ( Z ) is a function of stress, life, the life region boundaries, and the ( m )'s by using Equation 2-42.</td>
</tr>
<tr>
<td>KOMO⁹</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 ). Disabled for this application.</td>
</tr>
<tr>
<td>M4L1</td>
<td>5.1.3.3</td>
<td>Performs the driver transformation, Equations 2-68 through 2-73, 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, Equations 2-68 through 2-73, for location 2, the interior surface of the duct.</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.</td>
</tr>
<tr>
<td>MUSIG¹⁰</td>
<td>4.1.3.13</td>
<td>Calculates the posterior Normal distribution parameters: mean ( m_b ) and standard deviation ( \sigma_b ) for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NARBN2</td>
<td>5.1.3.4</td>
<td>Calculates the composite stress-time history by using Equations 2-82 and 2-84 and then calls RAINF2 to calculate the fatigue life.</td>
</tr>
</tbody>
</table>
### Table 7-3  List of Subprograms for Program HEXHCF (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<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 by using Equations 2-88 and 2-89.</td>
</tr>
<tr>
<td>NORMGN$^{11}$</td>
<td>4.4.3</td>
<td>Generates Normal($\mu$, $\sigma^2$) random variates.</td>
</tr>
<tr>
<td>NORGNG$^{7}$</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$^{12}$</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$^{13}$</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$^{14}$</td>
<td>5.1.3.5</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 HEXHCD and RELATD; calls CONVRT to transform the stress data to a stress ratio of $-1.0$; and echoes the data to HEXHCO 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 stress, life, the life region boundaries, and the $m$'s by using Equation 2-42.</td>
</tr>
<tr>
<td>SORTM$^{15}$</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>4.1.3.4</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 $X = \ln(Stress)$ by using Equations 2-20 and 2-21; to be used in the credibility range calculations.</td>
</tr>
<tr>
<td>THWELD</td>
<td>5.1.3.2</td>
<td>Controls the logical flow for the driver transformation and fatigue life calculations.</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>4.1.11</td>
<td>Performs premature program termination, when required.</td>
</tr>
<tr>
<td>TRNSFM$^{16}$</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>

---

1. No data regions to the right are discussed on Page 2-17.  
2. The Beta distribution is discussed on Page 2-25.  
3. Concavity constraints are discussed on Pages 2-13 through 2-14.  
4. The stress transformation is discussed on Page 2-7.
5 The median S/N curve parameter estimation calculations are described on Pages 2-15 through 2-18.
6 Selection of the \( \{m_j\} \) parameters is discussed on Page 2-15.
7 Combining information to obtain the posterior credibility ranges on \( m \) is discussed on Page 2-13.
8 The information aggregation calculations are discussed on Pages 2-6 through 2-14.
9 Extension of the S/N curve to the left is discussed on Page 2-17.
10 Calculation of the truncated Normal distribution parameters is discussed on Page 2-14.
11 The Normal distribution is discussed on Page 2-23.
12 The parameter estimation calculations are discussed on Pages 2-15 through 2-18.
13 The Uniform distribution is discussed on Page 2-23.
14 Rainflow cycle counting is discussed on Page 2-51 and in Appendix 2A.
15 The need for saving \( m \)'s is discussed on Page 2-15.
16 The S/N data transformation is discussed on Page 2-16.
7.1.2.3 Description of Variables

A list of variables used in the plain welded duct HCF code, HEXHCF, is given in Table 7-4. 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, MAXLD, MAXLIF, MAXM, MAXMM, MAXREG, and MAXSEG.

### Table 7-4 List of Variables for Program HEXHCF

(Footnotes are at the end of the table)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>AERD</td>
<td>RE</td>
<td>( \lambda_{DAERO} ) in Equation 2-81, the randomly selected load scale factor for the AERodynamic Dynamic load components.</td>
</tr>
<tr>
<td>AERDA</td>
<td>RE</td>
<td>Dynamic aerodynamic load scale factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>AERDB</td>
<td>RE</td>
<td>Dynamic aerodynamic load scale factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>AERS</td>
<td>RE</td>
<td>( \lambda_{STAERO} ) in Equation 2-81, the randomly selected load scale factor for the AERodynamic Static load components.</td>
</tr>
<tr>
<td>AERSA</td>
<td>RE</td>
<td>Static aerodynamic load scale factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>AERSB</td>
<td>RE</td>
<td>Static aerodynamic load scale factor Uniform distribution upper bound.</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 to be used in the truncated Normal median S/N curve calculation. ¹</td>
</tr>
<tr>
<td>ALPHA</td>
<td>RE</td>
<td>( \alpha ) ( (1 , ^{°}\text{R}) ) in Equation 2-70, the coefficient of thermal expansion. (see COEXP in HEXHCF)</td>
</tr>
<tr>
<td>ANGLE</td>
<td>RE</td>
<td>( \phi ) ( \text{(rad)} ) in Equation 2-68, the angle measured counterclockwise from Z-direction to the critical circumferential location.</td>
</tr>
<tr>
<td>AREA</td>
<td>RE</td>
<td>( A ) ( \text{(in.}^2) ) in Equation 2-68, the cross-sectional area of the duct wall.</td>
</tr>
<tr>
<td>ASTR</td>
<td>RE</td>
<td>( \lambda_{AERO_{str}} ) in Equation 2-81, the randomly selected aerodynamic stress analysis accuracy factor.</td>
</tr>
</tbody>
</table>

¹ Footnotes are at the end of the table.
Table 7-4  List of Variables for Program HEXHCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Astra</td>
<td>Re</td>
<td>Aerodynamic stress analysis accuracy factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>ASTRB</td>
<td>Re</td>
<td>Aerodynamic stress analysis accuracy factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>BIGK(0:MAXREG)</td>
<td>Re</td>
<td>1-D array containing values of the materials model location parameter $K$, Equation 2-12, where $A = K^m$.</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>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 $\beta_0$, Equation 2-11, which characterizes the intrinsic variation of the S/N data set.</td>
</tr>
<tr>
<td>COEXP</td>
<td>Re</td>
<td>$\alpha$ ($/^{*}$R) in Equation 2-70, the COefficient of thermal EXPansion. (see ALPHA in THWELD)</td>
</tr>
<tr>
<td>DI</td>
<td>Re</td>
<td>$D_i$ (in.), the randomly selected duct inner diameter at the weld used to calculate $R_i$ in Equation 2-68.</td>
</tr>
<tr>
<td>DIA</td>
<td>Re</td>
<td>$D_i$ lower bound of Beta distribution.</td>
</tr>
<tr>
<td>DIB</td>
<td>Re</td>
<td>$D_i$ upper bound of Beta distribution.</td>
</tr>
<tr>
<td>DIR</td>
<td>Re</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for $D_i$.</td>
</tr>
<tr>
<td>DIR1</td>
<td>Re</td>
<td>$\rho$ Uniform distribution lower bound of Beta distribution of $D_i$.</td>
</tr>
<tr>
<td>DIR2</td>
<td>Re</td>
<td>$\rho$ Uniform distribution upper bound of Beta distribution of $D_i$.</td>
</tr>
<tr>
<td>DIT</td>
<td>Re</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ 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>DIT1</td>
<td>RE</td>
<td>$\theta$ Uniform distribution lower bound of Beta distribution of $D_i$.</td>
</tr>
<tr>
<td>DIT2</td>
<td>RE</td>
<td>$\theta$ Uniform distribution upper bound of Beta distribution of $D_i$.</td>
</tr>
<tr>
<td>DLTAT</td>
<td>RE</td>
<td>DeLTA T. $\Delta T$ (°R) in Equation 2-70, the temperature difference across the wall of the duct.</td>
</tr>
<tr>
<td>DPCMUB</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>DSTR</td>
<td>RE</td>
<td>$\lambda_{Dyn}$ in Equation 2-81, the randomly selected dynamic stress analysis accuracy factor.</td>
</tr>
<tr>
<td>DSTRA</td>
<td>RE</td>
<td>Dynamic stress analysis accuracy factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>DSTRB</td>
<td>RE</td>
<td>Dynamic stress analysis accuracy factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>DTIMU</td>
<td>RE</td>
<td>Value of (TIMUB – TIMUA).</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>DUM</td>
<td>RE</td>
<td>Dummy variable</td>
</tr>
<tr>
<td>E(MAXSEG)</td>
<td>RE</td>
<td>1-D array containing the strain $\varepsilon$ values for the stress/strain versus strain curve.</td>
</tr>
<tr>
<td>EM</td>
<td>RE</td>
<td>$E$ (psi) in Equation 2-70, Young’s modulus of elasticity for the material.</td>
</tr>
<tr>
<td>FATLIF</td>
<td>RE</td>
<td>Value of FATigue LIFe calculated (sec).</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>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 2-73, used to find stress concentration due to weld eccentricity, $K_{OFF}$.</td>
</tr>
<tr>
<td>FTEST</td>
<td>LOG</td>
<td>File TEST. Used to test for the existence of a reference time history file before attempting to open it.</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>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>GAM</td>
<td>RE</td>
<td>$\lambda_{\text{dam}}$ in Equation 2-91, the randomly selected damage accumulation model accuracy factor. See Section 2.2.1.4 for a discussion of the damage calculations.</td>
</tr>
<tr>
<td>GAMA</td>
<td>RE</td>
<td>Damage accumulation model accuracy factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>GAMB</td>
<td>RE</td>
<td>Damage accumulation model accuracy factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls inner DO loop.</td>
</tr>
<tr>
<td>II</td>
<td>INT</td>
<td>Controls DO loop for narrow-band random, superimposed sinusoidal, and dynamic aerodynamic loads.</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>K(2, 2)</td>
<td>RE</td>
<td>2-D array containing the fatigue stress concentration factors required for the stress analysis. $K(1,<em>)$ is $K_{T1}$ in Equation 2-68 and $K(2,</em>)$ is $K_{T2}$ in Equation 2-69. $K(1,1)$ is the outer diameter axial stress concentration factor, the value of $KGOD \times KWOD$; $K(1,2)$ is the inner diameter axial stress concentration factor, the value of $KGID \times KWID$; $K(2,1)$ is the outer diameter hoop stress concentration factor; and $K(2,2)$ is the inner diameter hoop stress concentration factor. (see $KT(2,2)$ in HEXHCF)</td>
</tr>
<tr>
<td>KGID</td>
<td>RE</td>
<td>Axial stress concentration factor due to geometry for the duct inner diameter used to calculate $K_{T1}$ in Equation 2-68.</td>
</tr>
<tr>
<td>KGOD</td>
<td>RE</td>
<td>Axial stress concentration factor due to geometry for the duct outer diameter used to calculate $K_{T1}$ in Equation 2-68.</td>
</tr>
<tr>
<td>KOFF</td>
<td>RE</td>
<td>$K_{\text{OFF}}$ in Equation 2-73, the stress concentration factor due to eccentricity of the weld.</td>
</tr>
<tr>
<td>KRATIO</td>
<td>RE</td>
<td>Ratio of $\text{MED} K^* / \text{MED} K$ in Equation 2-48. $KRATIO$ is constant over life regions for the materials model.</td>
</tr>
<tr>
<td>KT(2, 2)</td>
<td>RE</td>
<td>2-D array containing the fatigue stress concentration factors required for the stress analysis. $KT(1,<em>)$ is $K_{T1}$ in Equation 2-68 and $KT(2,</em>)$ is $K_{T2}$</td>
</tr>
</tbody>
</table>
Table 7-4  List of Variables for Program HEXHCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>in Equation 2-69. KT(1,1) is the outer diameter axial stress concentration factor, the value of KGOD * KWOD; KT(1,2) is the inner diameter axial stress concentration factor, the value of KGID * KWID; KT(2,1) is the outer diameter hoop stress concentration factor; and KT(2,2) is the inner diameter hoop stress concentration factor. (see K(2,2) in THWELD)</td>
</tr>
<tr>
<td>KWID</td>
<td>RE</td>
<td>Randomly selected axial stress concentration factor due to the weld for the duct inner diameter used to calculate $K_{T_1}$ in Equation 2-68.</td>
</tr>
<tr>
<td>KWIDA</td>
<td>RE</td>
<td>Inner diameter weld axial stress concentration factor lower bound of Beta distribution.</td>
</tr>
<tr>
<td>KWIDB</td>
<td>RE</td>
<td>Inner diameter weld axial stress concentration factor upper bound of Beta distribution.</td>
</tr>
<tr>
<td>KWIDR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location $\rho$ for the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWIDR1</td>
<td>RE</td>
<td>$\rho$ Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWIDR2</td>
<td>RE</td>
<td>$\rho$ Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWDIT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ for the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWDIT1</td>
<td>RE</td>
<td>$\theta$ Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWDIT2</td>
<td>RE</td>
<td>$\theta$ Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWOD</td>
<td>RE</td>
<td>Randomly selected axial stress concentration factor due to the weld for the duct outer diameter used to calculate $K_{T_1}$ in Equation 2-68.</td>
</tr>
<tr>
<td>KWODA</td>
<td>RE</td>
<td>Outer diameter weld axial stress concentration factor lower bound of Beta distribution.</td>
</tr>
<tr>
<td>KWODB</td>
<td>RE</td>
<td>Outer diameter weld axial stress concentration factor upper bound of Beta distribution.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>KWODR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter ( \rho ) for the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODR1</td>
<td>RE</td>
<td>( \rho ) Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODR2</td>
<td>RE</td>
<td>( \rho ) Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter ( \theta ) of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODT1</td>
<td>RE</td>
<td>( \theta ) Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.</td>
</tr>
<tr>
<td>KWODT2</td>
<td>RE</td>
<td>( \theta ) Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.</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>LAMN</td>
<td>RE</td>
<td>( \lambda_{\text{RANDOM}} ) in Equation 2-81, the randomly selected load scale factor for the narrow-band random loads. See Section 2.1.3.2 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>
</tbody>
</table>
Table 7-4  List of Variables for Program HEXHCF (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<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_{\text{SINUSOIDAL}} ) in Equation 2-81, the randomly selected load scale factor for the superimposed sinusoidal loads. See Section 2.1.3.2 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>LAMSSSG</td>
<td>RE</td>
<td>The resulting standard deviation ( \sigma ) of the Normal distribution for the superimposed sinusoidal load scale factor, where ( \sigma = C/(1 + kC) ).</td>
</tr>
<tr>
<td>LAMW</td>
<td>RE</td>
<td>LAMbda Weld offset, the randomly selected ( \lambda_{\text{OFF}} ) in Equation 2-73, the accuracy factor for the weld offset eccentricity stress concentration factor, ( K_{\text{OFF}} ).</td>
</tr>
<tr>
<td>LAMWA</td>
<td>RE</td>
<td>( \lambda_{\text{OFF}} ) Uniform distribution lower bound.</td>
</tr>
<tr>
<td>LAMWB</td>
<td>RE</td>
<td>( \lambda_{\text{OFF}} ) Uniform distribution upper bound.</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 HEXHCF. 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(\text{BIGK}) \times \text{MM} ) for each life region of the S/N curve.</td>
</tr>
</tbody>
</table>
### Table 7-4 List of Variables for Program HEXHCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LNZ</td>
<td>RE</td>
<td>(\ln(2)) in Equation 2-48, the Normal(0, PVAR) random variate for the materials process variation aspect of the materials model.</td>
</tr>
<tr>
<td>LOCAT</td>
<td>INT</td>
<td>Critical location of interest on the duct wall where 1 is the exterior surface of the duct, and 2 is the interior surface of the duct.</td>
</tr>
<tr>
<td>LPHIM(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of (\ln(\text{PHI}) \cdot MM) for each life region of the S/N curve.</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-lb) in Equation 2-68, the moment load components about the y axis; and (M(2,</em>)) is (M_z) (in-lbs) in Equation 2-68, 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 set per region allowed for S/N curve. The maximum number of data points per set allowed is 50.</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 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 24,000.</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>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>MCOUNT</td>
<td>INT</td>
<td>Counts number of m’s to be used to calculate 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>MEDM(MAXMM)</td>
<td>RE</td>
<td>1-D array containing the empirical median ( m ) for each life region of the S/N curve.(^4)</td>
</tr>
<tr>
<td>MI</td>
<td>RE</td>
<td>( I ) (in.(^4)) in Equation 2-68, the cross-sectional Moment of Inertia.</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>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.-lbf) in Equation 2-68, the moment load components about the ( y ) axis; and MLAM(2,</em>) is ( M_z ) (in.-lbf) in Equation 2-68, the moment load components about the ( z ) axis.</td>
</tr>
<tr>
<td>MM(0:MAXREG)</td>
<td>RE</td>
<td>( m_j ) in Equation 2-12, 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.(^5)</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. MSLAM(1) is ( M_y ) (in.-lbf) in Equation 2-68, the moment load component about the ( y ) axis; and MSLAM(2) is ( M_z ) (in.-lbf) in Equation 2-68, the moment load component about the ( z ) axis.</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 2-68, the moment load component about the ( y ) axis; and MSTAT(2) is ( M_z ) (in.-lbf) in Equation 2-68, the moment load component about the ( z ) axis.</td>
</tr>
<tr>
<td>MU(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution mean(^6) 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.(^3)</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NBND(0:MAXREG)</td>
<td>RE</td>
<td>$N_{i,i+1}$ in Equation 2-35, 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>NEUB</td>
<td>RE</td>
<td>$\lambda_{neu}$ in Equation 2-89, the randomly selected Neuber's rule model accuracy factor.</td>
</tr>
<tr>
<td>NEUBA</td>
<td>RE</td>
<td>Neuber's rule model accuracy factor Uniform distribution lower bound.</td>
</tr>
<tr>
<td>NEUBBB</td>
<td>RE</td>
<td>Neuber's rule model accuracy factor Uniform distribution upper bound.</td>
</tr>
<tr>
<td>NEWLIF</td>
<td>RE</td>
<td>Fatigue life value (sec) returned from call to function THWELD.</td>
</tr>
<tr>
<td>NF(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values from the array RAWNF() for the specific material S/N data set partitioned into life regions.</td>
</tr>
<tr>
<td>NHYPER</td>
<td>INT</td>
<td>The outer loop size.</td>
</tr>
<tr>
<td>NLIFE</td>
<td>INT</td>
<td>The inner loop size.</td>
</tr>
<tr>
<td>NLIFET</td>
<td>INT</td>
<td>Total number of lives calculated by program HEXHCF. Value of NHYPER * NLIFE.</td>
</tr>
<tr>
<td>NLOAD</td>
<td>INT</td>
<td>NLOAD in Equation 2-81, the number of dynamic or time-varying loads.</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 wishes the median calculation to be performed.</td>
</tr>
<tr>
<td>NORM</td>
<td>RE</td>
<td>The variable NORM functions in two capacities. In the outer loop of HEXHCF, NORM is a Uniformly distributed random variate used to select the Normal distribution parameters $\mu$ and $\sigma$ for the flow condition drivers $T_i$, $T_o$, and $p_i$. In the inner loop, NORM is a Normally distributed random variate used to select the actual values of the flow conditions to be used in the driver transformation.</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>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 2-70, the materials Poisson's ratio.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NUMREG</td>
<td>INT</td>
<td>R in Equation 2-11, the number of life regions of interest in the S/N curve.</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 2-68, the dynamic or time-varying axial load components.</td>
</tr>
<tr>
<td>PC</td>
<td>RE</td>
<td>$p_i$ (psi) in Equation 2-68, 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>$\mu$ Uniform distribution lower bound of Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCMUB</td>
<td>RE</td>
<td>$\mu$ Uniform distribution upper bound of Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCO</td>
<td>RE</td>
<td>$p_o$ (psi) in Equation 2-68, 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>$\sigma$ Uniform distribution lower bound of Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PCSIGB</td>
<td>RE</td>
<td>$\sigma$ Uniform distribution upper bound of Normal distribution of the internal pressure $p_i$.</td>
</tr>
<tr>
<td>PERIOD</td>
<td>RE</td>
<td>$T$ (sec) in Equation 2-91, the length of time in seconds of the reference time history.</td>
</tr>
<tr>
<td>PHI</td>
<td>RE</td>
<td>$\phi$ in Equation 2-11, the materials intrinsic variation, or scatter, given by a Weibull($\beta_o$, $\eta_o$($\beta_o$)) random variate.</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 2-68, 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>PSIG</td>
<td>RE</td>
<td>$\sigma$ in Equation 2-48, the value of SQRT(PVAR).</td>
</tr>
<tr>
<td>PSLAM</td>
<td>RE</td>
<td>P (lbs) in Equation 2-68, the static axial load component scaled by ASTR and AERS.</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 2-68, the static axial load component.</td>
</tr>
<tr>
<td>PVAR</td>
<td>RE</td>
<td>$\sigma^2$ in Equation 2-48, characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.</td>
</tr>
<tr>
<td>RAINF2</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.</td>
</tr>
<tr>
<td>RI</td>
<td>RE</td>
<td>$R_i$ (in.) in Equation 2-68, the duct inner radius.</td>
</tr>
<tr>
<td>RO</td>
<td>RE</td>
<td>$R_o$ (in.) in Equation 2-68, the duct outer radius.</td>
</tr>
<tr>
<td>ROT</td>
<td>RE</td>
<td>$R$ Over $T$, the value of the ratio $R/\epsilon_t$.</td>
</tr>
<tr>
<td>RT(IO)</td>
<td>RE</td>
<td>1-D array containing values of $R/\epsilon_t$ used in conjunction with $F_k$, Equation 2-73, to find stress concentration due to weld eccentricity, $K_{OFF}$.</td>
</tr>
<tr>
<td>S(4, MAXM)</td>
<td>RE</td>
<td>2-D array containing the total component stress-time histories $\sigma_k(t)$ (psi), Equation 2-82, resulting from the combination of static, narrow-band random, sinusoidal, and aerodynamic loads. $S(1,\ast)$ is the axial stress-time history $\sigma_1(t)$; $S(2,\ast)$ is the hoop stress-time history $\sigma_2(t)$; $S(3,\ast)$ is the radial stress-time history $\sigma_3(t)$; and $S(4,\ast)$ is the shear stress-time history $\sigma_4(t)$.</td>
</tr>
<tr>
<td>SBND(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the stress values (psi) with stress ratio $\epsilon = -1.0$, 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>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>
</tbody>
</table>

7 - 138
<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEFF(MAXM)</td>
<td>RE</td>
<td>1-D array containing the EFFective or uni-axial stress-time history $\sigma(t)$ (psi), Equation 2-84, resulting from the combination of static, narrow-band random, sinusoidal, and aerodynamic loads for all four stress components.</td>
</tr>
<tr>
<td>SIG(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution standard deviation $\beta$ of the materials model shape parameter $m$, for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>SKT1</td>
<td>RE</td>
<td>$K_{T1}$ in Equation 2-68, the stress concentration factor for the axial stress.</td>
</tr>
<tr>
<td>SKT2</td>
<td>RE</td>
<td>$K_{T2}$ in Equation 2-69, the stress concentration factor for the hoop stress.</td>
</tr>
<tr>
<td>STATIC(4)</td>
<td>RE</td>
<td>1-D array containing values of the static stresses $\sigma_{STk}$ (psi), Equation 2-82. STATIC(1) is the axial stress $\sigma_{ST1}$; STATIC(2) is the hoop stress $\sigma_{ST2}$; STATIC(3) is the radial stress $\sigma_{ST3}$; and STATIC(4) is the shear stress $\sigma_{ST4}$.</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>STRAMP(4, MAXLD)</td>
<td>RE</td>
<td>2-D array containing values of the amplitudes of the dynamic or time-varying stresses $\sigma_{Dk}$ (psi), Equation 2-82. STRAMP(1,1) is $\sigma_{D11}$, the amplitude of the $i$th axial stress; STRAMP(2,1) is $\sigma_{D21}$, the amplitude of the $i$th hoop stress; STRAMP(3,1) is $\sigma_{D31}$, the amplitude of the $i$th radial stress; and STRAMP(4,1) is $\sigma_{D41}$, the amplitude of the $i$th shear stress.</td>
</tr>
<tr>
<td>STRHIS(MAXLD, MAXM)</td>
<td>RE</td>
<td>2-D array containing $\sigma_i(t)$, Equation 2-82, the reference time histories for the dynamic or time-varying load components.</td>
</tr>
<tr>
<td>SZERO</td>
<td>RE</td>
<td>Stress tensile test point, $S_o$ (psi).</td>
</tr>
<tr>
<td>T(MAXLD)</td>
<td>RE</td>
<td>1-D array containing $M_x$ (in.-lbs) in Equation 2-72, the dynamic or time-varying torsional load components.</td>
</tr>
<tr>
<td>TEST</td>
<td>RE</td>
<td>Uniform(0, 1) random variate used to determine Beta distribution for $w_{OFF}$.</td>
</tr>
</tbody>
</table>
Table 7-4  List of Variables for Program HEXHCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<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 2-68.</td>
</tr>
<tr>
<td>THICA</td>
<td>RE</td>
<td>$t$ lower bound of Beta distribution.</td>
</tr>
<tr>
<td>THICB</td>
<td>RE</td>
<td>$t$ upper bound of Beta distribution.</td>
</tr>
<tr>
<td>THICR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for the wall thickness $t$.</td>
</tr>
<tr>
<td>THICR1</td>
<td>RE</td>
<td>$\rho$ Uniform distribution lower bound of Beta distribution of $t$.</td>
</tr>
<tr>
<td>THICR2</td>
<td>RE</td>
<td>$\rho$ Uniform distribution upper bound of 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>$\theta$ Uniform distribution lower bound of Beta distribution of $t$.</td>
</tr>
<tr>
<td>THICT2</td>
<td>RE</td>
<td>$\theta$ Uniform distribution upper bound of Beta distribution of $t$.</td>
</tr>
<tr>
<td>THWELD</td>
<td>RE</td>
<td>Real function that controls the logical flow for the driver transformation and fatigue life calculations of a duct at a weld subjected to thermal loads, and then returns the fatigue life (sec).</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), in Equation 2-70, the temperature difference across the wall of the duct.</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>$\mu$ Uniform distribution lower bound of Normal distribution of the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TIMUB</td>
<td>RE</td>
<td>$\mu$ Uniform distribution upper bound of 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>$\sigma$ Uniform distribution lower bound of Normal distribution of the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>TISIGB</td>
<td>RE</td>
<td>$\sigma$ Uniform distribution upper bound of Normal distribution of the inner wall surface temperature $T_i$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TLAM(MAXLD)</td>
<td>RE</td>
<td>1-D array containing $M_x$ (in.-lbf) in Equation 2-72, the dynamic or time-varying torsional load components scaled by DSTR or ASTR and LAMN, LAMS or AERD, as appropriate, according to variable TYPE().</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), in Equation 2-70, the temperature difference across the wall of the duct.</td>
</tr>
<tr>
<td>TOMU</td>
<td>RE</td>
<td>Randomly selected Normal distribution parameter $\mu$ for the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TOMUA</td>
<td>RE</td>
<td>$\mu$ Uniform distribution lower bound of Normal distribution of the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TOMUB</td>
<td>RE</td>
<td>$\mu$ Uniform distribution upper bound of 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>$\sigma$ Uniform distribution lower bound of Normal distribution of the outer wall surface temperature $T_o$.</td>
</tr>
<tr>
<td>TOSIGB</td>
<td>RE</td>
<td>$\sigma$ Uniform distribution upper bound of Normal distribution of the outer wall surface temperature $T_o$.</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 &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 $S_o$.</td>
</tr>
<tr>
<td>TRUNC</td>
<td>RE</td>
<td>Value used to filter out noise in the composite stress-time history during rainflow cycle counting. See Section 2.2.1.4 for a discussion of rainflow cycle counting.</td>
</tr>
<tr>
<td>TSLAM</td>
<td>RE</td>
<td>$M_x$ (in.-lbf) in Equation 2-72, the static torsional load component scaled by ASTR and AERS.</td>
</tr>
<tr>
<td>TSTAT</td>
<td>RE</td>
<td>$M_x$ (in.-lbf) in Equation 2-72, the static torsional load component.</td>
</tr>
</tbody>
</table>
Table 7-4  List of Variables for Program HEXHCF (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<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 dynamic aerodynamic load factor.</td>
</tr>
<tr>
<td>V(2, MAXLD)</td>
<td>RE</td>
<td>2-D array containing the dynamic or time-varying shear load components. V(1,<em>) is $V_y$ (lbs) in Equation 2-72, the shear load components along the y axis; and V(2,</em>) is $V_z$ (lbs) in Equation 2-72, the shear load components along the z axis.</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; 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>VLAM(2, MAXLD)</td>
<td>RE</td>
<td>2-D array containing the dynamic or time-varying shear load components scaled by DSTR or ASTR and LAMN, LAMS, or AERD, as appropriate, according to variable TYPE(). VLAM(1,<em>) is $V_y$ (lbs) in Equation 2-72, the shear load components along the y axis; and VLAM(2,</em>) is $V_z$ (lbs) in Equation 2-72, the shear load components along the z axis.</td>
</tr>
<tr>
<td>VSLAM(2)</td>
<td>RE</td>
<td>1-D array containing the static shear load components scaled by ASTR and AERS. VSLAM(1) is $V_y$ (lbs) in Equation 2-72, the shear load component along the y axis; and VSLAM(2) is $V_z$ (lbs) in Equation 2-72, the shear load component along the z axis.</td>
</tr>
<tr>
<td>VSTAT(2)</td>
<td>RE</td>
<td>1-D array containing the static shear load components. VSTAT(1) is $V_y$ (lbs) in Equation 2-72, the shear load component along the y axis; and VSTAT(2) is $V_z$ (lbs) in Equation 2-72, the shear load component along the z axis.</td>
</tr>
<tr>
<td>WOFF</td>
<td>RE</td>
<td>$W_{OFF}$ in Equation 2-73, the randomly selected Weld OFFSET (%).</td>
</tr>
<tr>
<td>WOFFA</td>
<td>RE</td>
<td>$W_{OFF}$ lower bound of Beta distribution 1.</td>
</tr>
<tr>
<td>WOFFB</td>
<td>RE</td>
<td>$W_{OFF}$ upper bound of Beta distribution 1.</td>
</tr>
<tr>
<td>WOFFC</td>
<td>RE</td>
<td>$W_{OFF}$ lower bound of Beta distribution 2.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>WOFFD</td>
<td>RE</td>
<td>$W^{OFF}$ upper bound of Beta distribution 2.</td>
</tr>
<tr>
<td>WOFFE</td>
<td>RE</td>
<td>Decimal equivalent percentage weight occurring in Beta distribution 1 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>$\rho$ Uniform distribution lower bound of Beta distribution 1 of $W^{OFF}$.</td>
</tr>
<tr>
<td>WOFFR2</td>
<td>RE</td>
<td>$\rho$ Uniform distribution upper bound of Beta distribution 1 of $W^{OFF}$.</td>
</tr>
<tr>
<td>WOFFR3</td>
<td>RE</td>
<td>$\rho$ Uniform distribution lower bound of Beta distribution 2 of $W^{OFF}$.</td>
</tr>
<tr>
<td>WOFFR4</td>
<td>RE</td>
<td>$\rho$ Uniform distribution upper bound of Beta distribution 2 of $W^{OFF}$.</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>$\theta$ Uniform distribution lower bound of Beta distribution 1 of $W^{OFF}$.</td>
</tr>
<tr>
<td>WOFFT2</td>
<td>RE</td>
<td>$\theta$ Uniform distribution upper bound of Beta distribution 1 of $W^{OFF}$.</td>
</tr>
<tr>
<td>WOFFT3</td>
<td>RE</td>
<td>$\theta$ Uniform distribution lower bound of Beta distribution 2 of $W^{OFF}$.</td>
</tr>
<tr>
<td>WOFFT4</td>
<td>RE</td>
<td>$\theta$ Uniform distribution upper bound of Beta distribution 2 of $W^{OFF}$.</td>
</tr>
<tr>
<td>Z</td>
<td>RE</td>
<td>$Z$ in Equation 2-48, the randomly selected process variation shift factor given by a Lognormal(0, $P^{VAR}$) 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.</td>
</tr>
</tbody>
</table>
1 The need for saving m's is discussed on Page 2-15.
2 Neuber's rule and the stress/strain curve are discussed on Pages 2-53 through 2-54.
3 See variable BLFPER() for a description of B-life.
4 The median S/N curve for the truncated Normal case is discussed on Page 2-15.
5 See Section 2.1.2.3 for a discussion on process variation in materials.
6 $m_\pi$ of the posterior density of $m$ is discussed on Page 2-14.
7 The posterior credibility ranges $\pi(m)$ are discussed on Page 2-13.
8 $\sigma_\pi$ of the posterior density of $m$ is discussed on Page 2-14.
9 Extension of the S/N curve to the left using the tensile point is discussed on Page 2-17.
   Disabled for this application.
### 7.1.2.4 Program HEXHCF Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program HEXHCF Listing Temporal Order, Uniform Distribution</td>
<td>7-147</td>
</tr>
<tr>
<td>Program HEXHCF Listing Temporal Order, Truncated Normal Distribution</td>
<td>7-149</td>
</tr>
<tr>
<td>HEXHCF</td>
<td>7-151</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-165</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-166</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-167</td>
</tr>
<tr>
<td>GAM</td>
<td>7-167</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-168</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-173</td>
</tr>
<tr>
<td>INIT</td>
<td>7-173</td>
</tr>
<tr>
<td>RCE</td>
<td>7-175</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-181</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-182</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-185</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-188</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-190</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-191</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-195</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-196</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-197</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-199</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-201</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-202</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-205</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-206</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-208</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-211</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-212</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-213</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-214</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-215</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-216</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-217</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-219</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-219</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-220</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-222</td>
</tr>
<tr>
<td>THWELD</td>
<td>7-223</td>
</tr>
<tr>
<td>M4L1</td>
<td>7-225</td>
</tr>
<tr>
<td>M4L2</td>
<td>7-228</td>
</tr>
<tr>
<td>NARBN2</td>
<td>7-232</td>
</tr>
<tr>
<td>RAINF2</td>
<td>7-234</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
</tr>
<tr>
<td>NEUBER</td>
<td>7-239</td>
</tr>
</tbody>
</table>

HEXHCF Version 4.2
Program HEXHCF Listing Temporal Order, Uniform Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEXHCF</td>
<td>7-151</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-168</td>
</tr>
<tr>
<td>INIT</td>
<td>7-173</td>
</tr>
<tr>
<td>RCE</td>
<td>7-175</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-181</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-182</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-188</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-185</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-190</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-191</td>
</tr>
<tr>
<td>ADDRREG</td>
<td>7-195</td>
</tr>
<tr>
<td>CONCAB</td>
<td>7-196</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-197</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-199</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-213</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-214</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-215</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-216</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-217</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-219</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-220</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-221</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-222</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-223</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-224</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-225</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-226</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-227</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-228</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-229</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-230</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-231</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-232</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-233</td>
</tr>
<tr>
<td>GAM</td>
<td>7-234</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>7-237</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-238</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-239</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-240</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-241</td>
</tr>
<tr>
<td>THWELD</td>
<td>7-242</td>
</tr>
<tr>
<td>M4L1</td>
<td>7-243</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
</tr>
<tr>
<td>M4L2</td>
<td>7-228</td>
</tr>
<tr>
<td>NARBN2</td>
<td>7-232</td>
</tr>
<tr>
<td>RAINF2</td>
<td>7-234</td>
</tr>
<tr>
<td>NEUBER</td>
<td>7-239</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-220</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-165</td>
</tr>
</tbody>
</table>
# Program HEXHCF Listing Temporal Order, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEXHCF</td>
<td>7-151</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-168</td>
</tr>
<tr>
<td>INIT</td>
<td>7-173</td>
</tr>
<tr>
<td>RCE</td>
<td>7-175</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-181</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-182</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-188</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-201</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-190</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-202</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-205</td>
</tr>
<tr>
<td>CONCAB</td>
<td>7-196</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-206</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-211</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-212</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-213</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-214</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-215</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-216</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-217</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-212</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-167</td>
</tr>
<tr>
<td>GAM</td>
<td>7-167</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-212</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-166</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-219</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-210</td>
</tr>
<tr>
<td>THWELD</td>
<td>7-223</td>
</tr>
<tr>
<td>M4L1</td>
<td>7-225</td>
</tr>
<tr>
<td>M4L2</td>
<td>7-228</td>
</tr>
<tr>
<td>NARB2</td>
<td>7-232</td>
</tr>
<tr>
<td>RAINF2</td>
<td>7-234</td>
</tr>
<tr>
<td>NEUBER</td>
<td>7-239</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-220</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-165</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-222</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-199</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-213</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-214</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-215</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-216</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-217</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-219</td>
</tr>
</tbody>
</table>
PROGRAM HEXHCF CONTROLS THE FLOW OF LOGIC OF THE HIGH CYCLE
FATIGUE PLAIN WELDED DUCT PROBLEM UNDER THERMAL LOADS

PROGRAMMER: L. NEWLIN
DATE: 6SEP91
VERSION: 4.2 -- (MATCHR V8.5, THDUCT V4.1, INSORT V2.1)

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

PROGRAM HEXHCF

SUBPROGRAMS: INFAGG, PAREST, NORMGN, PRYRV, BETAGN, WEIBGN, THWELD,
TRMNAT, INSORT, SORTM, EXPCTD

7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
11-26: user named-OLD

NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

IMPLICIT none

INTEGER MAXBLF, MAXDAT, MAXLD, MAXLIF, MAXM, MAXMM, MAXREG,
& MAXSEG

REAL PI

PARAMETER (MAXBLF = 10, MAXDAT = 50, MAXLD = 16,
& MAXLIF = 10000, MAXM = 24000, MAXMM = 20001, MAXREG = 3, MAXSEG = 10,
& PI = 3.141592654)

COMMON IOUT

INTEGER BLFPOS(MAXBLF), FILNUM(MAXLD), I, II, IOUT, J, K, L,
& LOCAT, MOUNT, MID, MPROC, NLIFE, NHYPER, NLLIFET, NLOAD, NMED,
& NPTS(MAXREG), NRAN, NUMREG, NUMSEG,
& TYPE(MAXLD), VARY, ZRREG

DOUBLE PRECISION RAND

REAL AERD, AERDA, AERDB, AERS, AERSA, AERSB,
& ALLM(MAXMM, MAXREG), ANGLE, ASTR, ASTR, ASTRB,
& BIGK(0:MAXREG), BIGK1, BLFPER(MAXBLF), BZERO, COEXP,
& DI, DIA, DIB, DIR, DIR1, DIR2, DIT, DIT1, DIT2, DLTAT,
& DPCF, DPCSIG, DSTR, DSTRA, DSTRB, DTMU, DTISIG, DTOMU,
& DTOSIG, DUM, E(MAXSEG), EM, FIFTY, FK(10), FTU, FTY,
& GAM, GAMA, GAMB, KGID, KGOD, KOTAT, KT(2, 2), KWD,
& KWD1, KWD1B, KWD1R, KWD1RI, KWD2, KWD21, KWD2I,
& KWD3, KWD3B, KWD3R, KWD3RI, KWD4, KWD4B, KWD4R,
& KWOD, KWOD1, KWOD1B, KWOD1R, KWOD1RI, KWOD2,
& KWOD2B, KWOD2R, KWOD2RI, KWOD3, KWOD3B, KWOD3R,
& KWOD3RI, KWOD4, KWOD4B, KWOD4R, KWOD4RI,
& KWE, KAMN, KAMN1, KAMN2, KAMN3, KAME, KAMRA, KAMRB,
& LAMN, LAMNA, LAMNB, LAMB, LAMMB, LAME, LAMS,
& LAMS, LAMSA, LAMSC, LAMSL, LAMSD,
& LAMSK, LAMSM, LAMSSG, LAMM, LAMMB, LMWB, LIFE(MAXLIF),
& LNA(0:MAXREG), LNZ, LPHIN(0:MAXREG)

REAL M(2, MAXLD), MEDM(MAXREG), MSLAM(2, MAXL), MM(0:MAXREG),
& MSLAM(2), MSTAT(2), MU(MAXL), NEND(0:MAXREG), NEUB,
& NEUM, NNEWB, NEWLIF, NF(MAXDAT, MAXREG), NORM, NNU,
& P(MAXLD), PC, PCMUA, PCMUB, PCO, PCSIG, PCsiga,
& PCsigb, PERIOD, PHI, PLAM(MAXL), PSLAM, PSTAT,
& PVAR, RANGE(2, MAXREG), RT(10), SBNB(0:MAXREG),
& SE(MAXSEG), SIG(MAXREG), STR(MAXDAT, MAXREG),
& STRHIS(MAXL, MAXM), SZERO, T(MAXL), TEST, THIC, THICA,
& THICB, THICR, THICRI, THICT, THICT1, THICT2,
& THWDEL, TIN, TIMU, TIMU, TIMUB, TISIG, TISIGA, TISIGB,
& TLAM(2), TOUT, TOUT, TOMA, TOMIB, TOSIG, TOSIGA,
& TOSIGB, TSS(0:MAXREG), TRUNC, TSLAM, TSTAT,
& V(2, MAXL), VSLAM(2, MAXL), VSTAT(2), WOFF,
& WOFFA, WOFFB, WOPF, WOPF, WOFF, WOFFH, WOFFH, WOFFLO,
& WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFR5, WOFFR6,
& WOFFR7, WOFFR8, WOFFR9, WOFFR10, WOFFR11, WOFFT, WOFFT4,
& Z

CHARACTER*6 LDNAME(MAXLD)
LOGICAL FTTEST

DATA {FILNUM(I), I = 1, MAXLD} /
& 1; 12, 13, 14, 15, 16, 17, 18, 19, 20,
& 21; 22, 23, 24, 25, 26' /

C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

OPEN (1, FILE = 'HEXHC', STATUS = 'OLD')
OPEN (2, FILE = 'HEXHCO', STATUS = 'NEW')
OPEN (8, FILE = 'DUMP', STATUS = 'NEW')
OPEN (9, FILE = 'LOWLIF', STATUS = 'NEW')

C INITIALIZE LOAD ARRAYS

PSTAT = 0.0
PSLAM = 0.0
TSTAT = 0.0
TSLAM = 0.0
MSTAT(1) = 0.0
MSTAT(2) = 0.0
MSLAM(1) = 0.0
MSLAM(2) = 0.0
VSTAT(1) = 0.0
VSTAT(2) = 0.0
VSLAM(1) = 0.0
VSLAM(2) = 0.0

DO 5 I = 1, MAXLD
P(I) = 0.0
PLAM(I) = 0.0
T(I) = 0.0
TLAM(I) = 0.0
M(1,I) = 0.0
M(2,I) = 0.0
MLAM(1,I) = 0.0
MLAM(2,I) = 0.0
V(1,I) = 0.0
V(2,I) = 0.0
VLAM(1,I) = 0.0
VLAM(2,I) = 0.0
CONTINUE

READ(1,*) RAND
WRITE(8,*)' RANDOM NUMBER SEED =', RAND
READ(1,*) IOUT
WRITE(8,*)' MATCHR = 10, HEXHC = 15, THWELD = 25 =', IOUT
READ(1,*) NLIFE
WRITE(8,*)' INNER LOOP SIZE =', NLIFE
READ(1,*) NHYPER
WRITE(8,*)' OUTER LOOP SIZE =', NHYPER
READ(1,*) VARY
WRITE(8,*)' TYPE OF S/N VARIATION DESIRED =', VARY
READ(1,*) NMED
WRITE(8,*)' NORMAL MEDIAN CURVE (0 - NO, 1 - YES) =', NMED
READ(1,*) MPROC
WRITE(8,*)' MATERIALS PROCESS VARIATION DESIRED' 
(0 - NO, 1 - YES) =', MPROC

IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
WRIT(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

7-152
WRITE(8,*) 'ERROR: INVALID TYPE OF MATERIALS PROCESS',
& 'VARIATION DESIRED'
ENDIF
READ(1,*) NBLIFE
IF (NBLIFE .GT. 0) READ(1,*) (BLFPER(J), J = 1, NBLIFE)

C ** READ DATA FROM HEXHC0
READ(1,*) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,
& WOFFC, WOFFD, WOFFR4, WOFFT3, WOFFT4,
& WOFFE, WOFFF
READ(1,*) KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2,
& KWIDA, KWIDB, KWIDR1, KWIDR2, KWIDT1, KWIDT2,
& DIA, DIR, DIR1, DIR2, DIT, DIT1, DIT2,
& THICA, THICB, THICR1, THICR2, THICT1, THICT2,
& LAMNA, LAMB, LAMNC, LAMND,
& LAMSA, LAMSB, LAMSC, LAMSD,
& TUMTA, TUMTB, TUMTC, TUMTD,
& TUMUA, TUMUB, TUMUC, TUMUD,
& PCMU, PCMU, PCSIGA, PCSIGB,
& PCMU, PCMU, PCMU, PCMU,
& PCMU, PCMU, PCMU, PCMU,
& PCMU, PCMU, PCMU, PCMU,
& DSA, DSA, DSA, DSA,
& LAMWA, LAMWB, NEUBA, NEUHB,
& GAMA, GAMBC
READ(I,*) NLOAD, PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1),
& VSTAT(2)
DO 15 I = 1, NLOAD
READ(I,*) LDNAME(I), P(I), T(I), M(I,I), M(2,I), V(I,I),
& V(2,I)
IF ((TYPE(I) .LT. I) .OR. (TYPE(I) .GT. 3)) THEN
WRITE(8,*) 'ERROR: LOAD INCORRECTLY TYPED'
CALL TRMNAT
ENDIF
15 CONTINUE
READ(I,*) KGOD, KGID, KT(2,1), KT(2,2), PCO, LOCAT, ANGLE,
& PERIOD, TRUNC, NRAN

C ** ECHO DATA TO HEXHCO
WRITE(3,900)
WRITE(3,901) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,
& WOFFC, WOFFD, WOFFR4, WOFFT3, WOFFT4,
& WOFFE, WOFFF
WRITE(3,902) KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2,
WRITE(3,903) KWIDA, KWIDB, KWIDR1, KWIDR2, KWIDT1, KWIDT2,
WRITE(3,904) DIA, DIR, DIR1, DIR2, DIT, DIT1, DIT2,
WRITE(3,905) THICA, THICB, THICR1, THICR2, THICT1, THICT2,
WRITE(3,906) LAMNA, LAMB, LAMNC, LAMND,
WRITE(3,907) LAMSA, LAMSB, LAMSC, LAMSD,
WRITE(3,908) TUMTA, TUMTB, TUMTC, TUMTD,
& TUMUA, TUMUB, TUMUC, TUMUD,
& PCMU, PCMU, PCSIGA, PCSIGB,
& PCMU, PCMU, PCMU, PCMU,
& PCMU, PCMU, PCMU, PCMU,
& PCMU, PCMU, PCMU, PCMU,
& PCMU, PCMU, PCMU, PCMU,
& DSA, DSA, DSA, DSA,
& LAMWA, LAMWB, NEUBA, NEUHB,
& EXP(GAMA), EXP(GAMBC)
WRITE(3,909) AERDA, AERDB, AERSA, AERSB,
& AERDA, AERDB, AERSA, AERSB,
& LAMWA, LAMWB, NEUBA, NEUHB,
& EXP(GAMA), EXP(GAMBC)
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,I), M(2,I), V(I,I),
& V(2,I)
20 CONTINUE
WRITE(3,925) KGOD, KGID, KT(2,1), KT(2,2), PCO, LOCAT, ANGLE,
& PERIOD, TRUNC, NLOAD, NRAN

C CONVERT ANGLE TO RADIANS FOR CALCULATIONS
ANGLE = ANGLE/180.00000 * PI
WRITE(3,926) ANGLE
IF (NRAN .GT. MAXM) 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 INITIALIZE THE STRESS-STRAIN ARRAYS

DO 30 J = 1, MAXSEG
  SE(J) = 0.00
  E(J) = 0.00
30 CONTINUE

READ(1,*), EM, COEXP, NU
READ(1,*), (FK(I), RT(I), IM = 1, 10)
READ(1,*), NUMSEG

WRITE(3,927) EM, COEXP, NU
WRITE(3,930) NUMSEG

C READ IN THE STRESS-STRAIN VALUES

DO 35 J = 1, NUMSEG
  READ(1,*), SE(J), E(J)
35 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, PTU, PTY, VARY, MPROC, KRATIO, PVAR)

ZROREG = 1
SZERO = 0.0

IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)
MCOUNT = 0

C ** INITIALIZE VARIABLES

DO 40 K = 1, MAXLIF
  LIFE(K) = 1.0E+36
40 CONTINUE

DO 45 J = 1, MAXBLF
  BLFPOS(J) = 0
45 CONTINUE

NLIFET = NHYPER * NLIFE

DTIMU = TIMUB - TIMUA
DTSIG = TISIGB - TISIGA
DTMU = TOMUB - TOMUA
DPSIG = TOSIGB - TOSIGA
DPCMU = PCMUB - PCMUA
DPSCIG = 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 ** 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 RANDOM (TEST, RAND)
  IF (TEST .LE. WOFFE) THEN
    CALL PRYRV (RAND, WOFFR1, WOFFR2, WOFFT1, WOFFT2, WOFF, WOFF)
    WOFFLO = WOFFA
    WOFFHI = WOFFB
  ELSE
    CALL PRYRV (RAND, WOFFR3, WOFFR4, WOFFT3, WOFFT4, WOFF, WOFF)
    WOFFLO = WOFFC
    WOFFHI = WOFFD
  ENDIF
ENDIF

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)
CALL NORMGN (RAND, 0.0, PSIG, LNZ)
IF (MPROC .EQ. 1) THEN
  Z = EXP (LNZ)
ELSE
  KRATIO = 1.0
  Z = 1.0
ENDIF
C ** INNER LOOP -- THIS LOOP GENERATES FATIGUE LIVES
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 LIFE
CALL BETAGN (RAND, WOFFR, WOFFT, WOFFLO, WOFFHI, WOFF)
CALL BETAGN (RAND, WIDR, WIDT, WID, WIDB, WID)
CALL BETAGN (RAND, WDR, WD, DIA, DIB, D)
CALL BETAGN (RANDOM, THICR, THIC, THICA, THICB, THIC)
CALL NORMGN (RANDOM, LA, LAMSG, LAM)
CALL NORMGN (RANDOM, TIMU, TISIG, TIN)
CALL NORMGN (RANDOM, TOSIG, TOU)
DLAT = TIN - TOUT
CALL NORMGN (RANDOM, PCMU, PCSIG, PC)
CALL PRIYRV (RANDOM, AERDA, AERDB, AERSA, AERSB, AERD, AERS)
CALL PRIYRV (RANDOM, DSTR, ASTR, DSTRA, ASTRB, DSTR, ASTR)
CALL PRIYRV (RANDOM, LAMN, LAMNB, NEUBA, NEUBB, LAMN, NEUB)
CALL PRIYRV (RANDOM, GAMMA, GAMB, GAMA, GAMB, GAM, DUM)
GAM = EXP(GAM)
CALL WEIBGN (BZERO, RANDOM, PHI)
IF (VARY .EQ. 0) PHI = 1.0
IF (IOUT .EQ. 15) THEN
WRITE(8,*) 'LAMN = ', LAMN, ' LAMS = ', LAMS
WRITE(8,*) 'THIC = ', THIC, ' AERD = ', AERD, ' AERS = ', AERS
WRITE(8,*) 'DSTR = ', DSTR, ' ASTR = ', ASTR
WRITE(8,*) 'PHI = ', PHI, ' PC = ', PC
END IF
C ** SCALE AERO STATIC LOADS
PSLAM = AER * ASTR * PSTAT
TSLAM = AER * ASTR * TSTAT
MSLAM(1) = AER * ASTR * MSTAT1
MSLAM(2) = AER * ASTR * MSTAT2
VSLAM(1) = AER * ASTR * VSTAT1
VSLAM(2) = AER * ASTR * VSTAT2
C ** SCALE TIME-VARYING LOADS
DO 230 II = 1, NLOAD
IF (TYPE(II) .EQ. 1) THEN
PLAM(II) = LAMN * DSTR * P(II)
TLAM(II) = LAMN * DSTR * T(II)
MLAM(1,II) = LAMN * DSTR * M(1,II)
MLAM(2,II) = LAMN * DSTR * M(2,II)
VLAM(1,II) = LAMN * DSTR * V(1,II)
VLAM(2,II) = LAMN * DSTR * V(2,II)
ELSE IF (TYPE(II) .EQ. 2) THEN
PLAM(II) = LAMN * DSTR * P(II)
TLAM(II) = LAMN * DSTR * T(II)
END IF
MLAM(1,II) = LAMS * DSTR * M(1,II)  
MLAM(2,II) = LAMS * DSTR * M(2,II)  
VLAM(1,II) = LAMS * DSTR * V(1,II)  
VLAM(2,II) = LAMS * DSTR * V(2,II)  
ELSE  
PLAM(II) = AERD * ASTR * P(II)  
TLAM(II) = AERD * ASTR * T(II)  
MLAM(1,II) = AERD * ASTR * M(1,II)  
MLAM(2,II) = AERD * ASTR * M(2,II)  
VLAM(1,II) = AERD * ASTR * V(1,II)  
VLAM(2,II) = AERD * ASTR * V(2,II)  
ENDIF

230 CONTINUE

IF (IOUT .EQ. 15) THEN
    WRITE(8,*) 'AERO STATIC LOADS'
    WRITE(8,*) 'P = ', PSLAM, ' T = ', TSLAM,
    & '12 = ', MSLAM(1), ' M3 = ', MSLAM(2),
    & 'V2 = ', VSLAM(1), ' V3 = ', VSLAM(2)
    WRITE(8,*) 'TIME-VARYING LOADS'
    DO 240 II = 1, NLOAD
        WRITE(8,*) 'P = ', PLAM(II), ' T = ', TLAM(II),
    & 'M2 = ', MLAM(1,II), ' M3 = ', MLAM(2,II),
    & 'V2 = ', VLAM(1,II), ' V3 = ', VLAM(2,II)
    240 CONTINUE
ENDIF

C ** CALCULATE AXIAL Kt's

K(1,1) = KGID * KWID
K(1,2) = KGID * KWID

IF (IOUT .EQ. 15) THEN
    WRITE(8,*) 'K(1,1) = ', K(1,1), ' K(1,2) = ', K(1,2)
ENDIF

C ** CALCULATE REGION DEPENDENT S/N CURVE PARAMETERS

DO 250 L = ZROREG, NUMREG
    LNA(L) = MM(L) * ALOG(BIGK(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 = ', 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)
    ENDIF

250 CONTINUE

C ** CALL THWELD OF THDUCT V4.1 TO CALCULATE FATIGUE LIFE

NEWLIF = GAM * THWELD (COEXP, ANGLE, DLTAT, E, EM, FTU,
    & FTY, DI, KT, KRATIO, LAMS, FK, KT, LNA, LNZ,
    & LOCAT, LPHIM, MLAM, MM, MSLAM, NEUB, NLOAD, NREN,
    & NU, NUMREG, NUMSEG, PLAM, PC, PCO, PERIOD, PSLAM,
    & TRSBND, SE, STRHIS, SZERO, TSLAM, THIC, TRUNC,
    & TSLAM, VLAM, VSLAM, WOFF, ZROREG)

IF (IOUT .EQ. 15) WRITE(8,*) 'NEWLIF = ', NEWLIF

IF (NLIFET .GE. 100) CALL NBSORT (NEWLIF, LIFE, NLIFET)

200 CONTINUE

150 CONTINUE

IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES

DO 300 J = 1, (NLIFET / 100)
    WRITE(9,*) J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
300 CONTINUE

7 - 157
C ** PRINT EMPIRICAL BLIVES

FIFTY = 0.50E0
WRITE(3,935)
DO 350 J = 1, NBBLIFE
   BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
WRITE(3,936) BLFPER(J), LIFE(BLFPOS(J))
350 CONTINUE
WRITE(3,936) FIFTY, LIFE(NLIFET/2)
ENDIF

C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED

IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) THEN

CALL SORTM (ALLM, NUMR, 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 HEXHCO

900 FORMAT(2X,'Copyright (C) 1990, California Institute of ', 
& 'Technology. U.S. Government',/2X,'Sponsorship under ', 
& 'NASA contract NAS7-918 is acknowledged.', 
& '/3X,'INPUT DATA'. 
& 14X,'DRIVERS',25X,'PARAMETER DISTRIBUTIONS', 
& 48X,'RHO',16X,'THETA')
901 FORMAT(/,2X,'WELD OFFSET (%)',3X,'Be(',F4.2,',',F5.2,')',6X, 
& 'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')', 
& 20X,'U(',F4.2,',',F5.2,')',6X,'U(',F7.5,',',F8.5,')', 
& 4X,'U(',F4.1,',',F5.1,')',/20X,'TEST = ',F4.2)
902 FORMAT(/,2X,'K WELD (OD)',7X,'Be(',F4.2,',',F5.2,')',6X, 
& 'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')', 
& 20X,'U(',F4.2,',',F5.2,')',6X,'U(',F7.5,',',F8.5,')', 
& 4X,'U(',F4.1,',',F5.1,')',/20X,'STRAIN GAGE FACTOR: ',F9.7)
903 FORMAT(/,2X,'INNER DIAMETER',4X,'Be(',F6.4,',',F7.4,')',2X, 
& 'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')', 
& 20X,'INNER PRESSURE',TX,'NORMAL: U(',F6.1,',',F7.1,')', 
& 'U(',F6.1,',',F7.1,')',/20X,'OUTER TEMPERATURE',4X,'NORMAL: U(', 
& F6.1,',',F7.1,')', U(',F5.1,',',F6.1,')', 
& 20X,'STRAIN GAGE FACTOR: ',F9.7)
904 FORMAT(/,2X,'LAMBDA RANDOM',5X,'k: U(',F7.5,',',F8.5,')', 
& 20X,'COEFFICIENT OF VARIATION: ',F9.7)
STOP
END

C******************************* SAMPLE 'HEXHCD' INPUT FILE
C
C 625 ................................ RANDOM NUMBER SEED
C 0 ................................ OUTPUT DUMP CONTROLLER
C 100 ................................ INNER LOOP SIZE
C 200 ................................ OUTER LOOP SIZE
C 2 ................................ S/N VARIATION -- UNIFORM REQUIRED
C 0 ................................ NORMAL MEDIAN NOT REQUIRED
C 0 ................................ MATERIALS PROCESS VARIATION NOT REQUIRED
C 3 ................................ NUMBER OF BLIVES TO BE PROVIDED
C 0.0001 ................................ B.01 LIFE
C 0.001 ................................ B.1 LIFE
C 0.05 ................................ B.5 LIFE
C 0.10 ................................ B.1 LIFE
C 0.50 ................................ B.5 LIFE
C 1.00 ................................ B.10 LIFE
C 0.10 0.10 0.50 0.50 5.0 5.0 WELD OFFSET (A,B) (R1,R2) (T1,T2)
C 0.00 0.00 0.00 0.00 0.0 0.0 WELD OFFSET (C,D) (R3,R4) (T3,T4)
C 0.10 1.33 0.30 0.70 0.8 1.2 WELD OFFSET TEST FOR HYPER-DISTRIBUTION 1
C 1.00 1.33 0.30 0.70 0.8 1.2 WELD OFFSET TEST FOR HYPER-DISTRIBUTION 2
C 0.25 1.74 0.30 0.70 0.5 1.2 K WELD (OD) (A,B) (R1,R2) (T1,T2)
C 0.1885 0.1915 0.50 0.75 0.5 0.5 INNER DIAM (A,B) (R1,R2) (T1,T2)
C 0.0113 0.0137 0.50 0.75 0.5 0.5 WALL THICK (A,B) (R1,R2) (T1,T2)

7 - 159
C 16
C 1.50  3.00  0.15  0.90 .......... LAMBDA NARROW-BAND RANDOM: k: U(A,B),
C  COEFF. OF VAR., STRAIN GAGE FACTOR
C 2.00  3.00  0.20  0.90 .......... LAMBDA SUPERIMPOSED SINE: k: U(A,B),
C  COEFF. OF VAR., STRAIN GAGE FACTOR
C MEAN BOUNDS SIGMA BOUNDS ----- NORMAL DISTRIBUTION PARAMETERS
C  486.  666.  29.  56.5 .......... INNER WALL TEMPERATURE, DEGREES R
C  799.  908.  49.5  48 .......... OUTER WALL TEMPERATURE, DEGREES R
C  3808.  4177.  69.  69 .......... INTERNAL PRESSURE, PSI
C  2.00  3.00 .......... LAMBDA DYNAMIC AERO U(A,B)
C  0.80  1.20 .......... DYNAMIC STRESS ANALYSIS ACCURACY U(A,B)
C  0.90  1.10 .......... AERO STRESS ANALYSIS ACCURACY U(A,B)
C  0.80  1.20 .......... LAMBDA K WELD OFFSET ACCURACY FACTOR
C  0.60  1.40 .......... NEUBER'S RULE MODEL ACCURACY FACTOR
C  -1.38629  0.95166 .......... DAMAGE MODEL ACCURACY U(lnA, lnB)
C 16 .......... NUMBER OF TIME-VARYING LOADS

PSTAT  TSTAT  MSTAT(1)  MSTAT(2)  VSTAT(1)  VSTAT(2)
C 0.00  0.00  -0.07214  0.00  0.00  0.00 .............. STATIC AERO LOADS
C FILE TYPE P()  T()  M(1,)  M(2,)  V(1,)  V(2,)  LOADS
C 'NBP' 1  0.050464  0.00  0.00  0.00  0.00  0.00 .............. RANDOM P
C 'NBT' 1  0.00  0.018395  0.00  0.00  0.00  0.00 .............. RANDOM T
C 'NB2' 1  0.00  0.00  0.00  0.00  0.00  0.00 .............. RANDOM M2
C 'NB3' 1  0.00  0.00  0.00  0.00  0.00  0.00 .............. RANDOM M3
C 'NBV2' 1  0.00  0.00  0.00  0.00  0.00  0.00 .............. RANDOM V2
C 'NBV3' 1  0.00  0.00  0.00  0.00  0.00  0.00 .............. RANDOM V3
C 'SN2' 2  0.01150  0.001927  0.051544  0.000312  0.000596  0.022412 .......... SINE 1
C 'SN2' 2  0.064034  0.001395  0.00  0.00  0.00  0.00 .............. SINE 2
C 'SN3' 2  0.0002284  0.001917  0.050250  0.000457  0.000295  0.021862 .......... SINE 3
C 'SN4' 2  0.023373  0.004778  0.427142  0.005777  0.000809  0.179889 .......... SINE 4
C 'SN5' 2  0.12763  0.000495  0.102889  0.007273  0.005673  0.051970 .......... SINE 5
C 'SN6' 2  0.00  0.00  0.00  0.00  0.00  0.00 .............. SINE 6
C 'AERO1' 3  0.00  0.00  0.00  0.00  0.00  0.00 .............. AERO 1
C 'AERO2' 3  0.00  0.00  0.00  0.00  0.00  0.00 .............. AERO 2
C 'AERO3' 3  0.00  0.00  0.00  0.00  0.00  0.00 .............. AERO 3
C 1.0  1.0  1.0  1.0 .............. K GEOM (OD,ID); K HOOP (OD,ID); KT(2,1) KT(2,2)
C 0.10 .............. WELD OFFSET
C 3640 .......... EXTERNAL (OUTER) PRESSURE (POC, PSI)
C 2 .......... SURFACE LOCATION (I=OUTER, 2=INNER)
C 270 .......... ANGLE (DEGREES)
C 1 .......... STRESS-TIME HISTORY PERIOD (SEC)
C 1 .......... STRESS-TIME HISTORY NOISE FILTER (PSI)
C 100 .......... NUMBER OF POINTS IN STRESS-TIME HISTORY
C 29000000 ................. YOUNG'S MODULUS, COEFF OF THERMAL EXPANSION,
C POISSON'S RATIO
C 0.615  2.0 .............. FK(1), RT(1)
C 0.639  4.80 .......... FK(2), RT(2)
C 0.753  7.20 .......... FK(3), RT(3)
C 0.813  9.60 .......... FK(4), RT(4)
C 1.053  12.80 .......... FK(5), RT(5)
C 0.933  15.80 .......... FK(6), RT(6)
C 0.993  20.00 .......... FK(7), RT(7)
C 1.029  24.00 .......... FK(8), RT(8)
C 1.053  30.00 .......... FK(9), RT(9)
C 1.053  20.00 .......... FK(10), RT(10)
C 6 .......... NUMBER OF SEGMENTS IN STRESS-STRAIN CURVE
C 21.99  0.00 .......... STRESS-STRAIN PRODUCT: SE(1), STRAIN: E(1)
C 55.77  0.003 .......... SE(2), E(2)
C 144.85  0.005 .......... SE(3), E(3)
C 322.73  0.010 .......... SE(4), E(4)
C 1945.90  0.050 .......... SE(5), E(5)
C 50688.0  0.660 .......... SE(6), E(6)
C 70 F, 321 STAINLESS STEEL ALLOY - WELDED .......... MATERIALS DESCRIPTION
C 27900.  76800.  1 13 .......... YIELD & ULTIMATE STRENGTHS, NDIV, NPTS
C 13 1 .......... # PTS IN DIV, STRESS RATIO, REGION
C 2000.  1000 .......... S(1), S(2)
C 40000.  2000 .......... S(2), S(2)
C 40000.  3000 .......... S(3), S(3)
C 40000.  4000 .......... S(4), S(4)

7 - 160
**LIST OF VARIABLES**

- **AERD**: Selected Aero Dynamic Load Scale Factor
- **AERDA**: Aero Dynamic Load Scale Factor Lower Bound
- **AERDB**: Aero Dynamic Load Scale Factor Upper Bound
- **AERS**: Selected Aero Static Load Scale Factor Lower Bound
- **AERSB**: Aero Static Load Scale Factor Upper Bound
- **ALLM()**: 2-D Array Containing MXV Values To Be Sorted For Each Region Angle
- **ANGL****: Angle Theta Measured Counter-Clockwise From The X2-Direction
- **ASTR**: Selected Aero Stress Analysis Accuracy Factor
- **ASTRA**: Aero Stress Analysis Accuracy Factor Lower Bound
- **ASTRB**: Aero Stress Analysis Accuracy Factor Upper Bound
- **B1GK()**: 1-D Array Containing Values of K, Where A = K ** M For Each Region
- **B1GK1**: Equal To B1GK(1) -- Dummy Parameter For Calls To Subroutine
- **BF****: Equal To T1G1(1) -- Dummy Variable For Calls To Subroutine
- **EXP****: Equal To T1G1(1)
- **BBF****: 1-D Array Containing User Specified BfLvS To Be Provided
- **BBFPOS()**: 1-D Array Containing Positions In Life() Of Empirical Blives
- **BZERO**: Value Of BetaAO Randomly Selected From Beta() Interval
- **BEXP****: Coefficient Of Thermal Expansion
- **DI**: Selected Weld Interior Diameter
- **DIA**: Weld Interior Diameter Lower Bound
- **DIB**: Weld Interior Diameter Upper Bound
- **DIR**: Selected Rho For Weld Interior Diameter
- **DIR1**: Weld Interior Diameter - Rho Lower Bound
- **DIR2**: Weld Interior Diameter - Rho Upper Bound
- **DIT**: Weld Interior Diameter - Theta Lower Bound
- **DIT2**: Weld Interior Diameter - Theta Upper Bound
- **D1****: Selected Temperature Difference Between Inner And Outer Surfaces -- Delta T
- **DP****: Equal To PCMUB - PCMU
- **DPC****: Equal To PCSIGB - PCSIGA
- **DS****: Selected Dynamic Stress Analysis Accuracy Factor
- **DSRA**: Dynamic Stress Analysis Accuracy Factor Lower Bound
- **DSRB**: Dynamic Stress Analysis Accuracy Factor Upper Bound
- **D1M**: Weld Interior Diameter
- **D1M1**: Weld Interior Diameter - Rho Lower Bound
- **D1M2**: Weld Interior Diameter - Rho Upper Bound
- **D1T**: Weld Interior Diameter - Theta Lower Bound
- **D1T2**: Weld Interior Diameter - Theta Upper Bound
- **D****: Selected Temperature Difference Between Inner And Outer Surfaces -- Delta T
- **DUM****: Dummy Variable
- **E()**: 1-D Array Which Contains The Strain Values
- **EM****: Elastic Modulus
- **EM1****: Equal To .5 -- Used To Access 50% Point In Life()
- **FNAM****: 1-D Array Containing Unit Number For Stress-Time Histories Files
- **FK()**: 1-D Array Containing Values Of Fk Used To Find Stress Concentration Due To Weld Eccentricity
- **FTEST**: File Test -- Used To Test Existence Of File
- **FTU**: Material Ultimate Strength
- **FTY**: Material Yield Strength
- **GA****: Selected Damage Accumulation Model Accuracy Factor, LambdaAdam
- **GAM****: Gamma Lower Bound
- **GAMB****: Gamma Upper Bound

...
CONTROLS DO LOOP FOR EACH LIFE CALCULATION
II
CONTROLS DO LOOP FOR LOADS
IOUT
CONTROLS DUMP TO SCREEN/PRINTER
J
CONTROLS DO LOOP FOR EACH BI-LIFE
I
CONTROLS DO LOOP FOR EACH HYPER PARAMETER SET
\( \text{KGID} \)
K GEOM (ID)
\( \text{KGOD} \)
SELECTED K GEOM (OD)
\( \text{KRATIO} \)
RATIO OF \( \text{K}^* / \text{K} \), CONSTANT OVER REGIONS AND COMPONENTS
\( \text{KT}() \)
FATIGUE STRESS CONCENTRATION FACTORS -- \( \text{KT}(1,1) = \text{KT AXIAL (OD)} \\
(= \text{KGOD} * \text{KWOD}) ; \text{KT}(1,2) = \text{KT AXIAL (ID)} (= \text{KGID} * \text{KWID}) \\
(= \text{KT}(2,1) = \text{KT HOOP (OD)} ; \text{KT}(2,2) = \text{KT HOOP (ID)} \\
\)
\( \text{KWID} \)
SELECTED K WELD (ID)
\( \text{KWIDA} \)
K WELD (ID) LOWER BOUND
\( \text{KWIDB} \)
K WELD (ID) UPPER BOUND
\( \text{KWIDR} \)
SELECTED RHO FOR K WELD (ID)
\( \text{KWIDR1} \)
K WELD (ID) -- RHO LOWER BOUND
\( \text{KWIDR2} \)
K WELD (ID) -- RHO UPPER BOUND
\( \text{KWIDT} \)
SELECTED THETA FOR K WELD (ID)
\( \text{KWIDT1} \)
K WELD (ID) -- THETA LOWER BOUND
\( \text{KWIDT2} \)
K WELD (ID) -- THETA UPPER BOUND
\( \text{KWOD} \)
SELECTED K WELD (OD)
\( \text{KWODA} \)
K WELD (OD) LOWER BOUND
\( \text{KWODB} \)
K WELD (OD) UPPER BOUND
\( \text{KWODR} \)
SELECTED RHO FOR K WELD (OD)
\( \text{KWODR1} \)
K WELD (OD) -- RHO LOWER BOUND
\( \text{KWODR2} \)
K WELD (OD) -- RHO UPPER BOUND
\( \text{KWODT} \)
SELECTED THETA FOR K WELD (OD)
\( \text{KWODT1} \)
K WELD (OD) -- THETA LOWER BOUND
\( \text{KWODT2} \)
K WELD (OD) -- THETA UPPER BOUND
L
CONTROLS DO LOOP FOR EACH REGION
\( \text{LAMN} \)
SELECTED LAMBDA FOR ONE SIGMA NARROW-BAND RANDOM LOADS
\( \text{LAMNA} \)
LAMBDA FOR NARROW-BAND RANDOM LOADS -- LOWER BOUND OF K
\( \text{LAMNB} \)
LAMBDA FOR NARROW-BAND RANDOM LOADS -- UPPER BOUND OF K
\( \text{LAMNC} \)
LAMBDA FOR NARROW-BAND RANDOM LOADS COEFFICIENT OF VARIATION
\( \text{LAMND} \)
NARROW-BAND RANDOM LOADS STRAIN GAGE ACCURACY FACTOR
\( \text{LAMNK} \)
LAMBDA FOR NARROW-BAND RANDOM LOADS \( \text{k} \) -- INDICATES VARIATION DUE TO SAMPLE SIZE
\( \text{LAMNMU} \)
MEAN OF LAMBDA FOR NARROW-BAND RANDOM LOADS (\( \text{MU} \), NORMAL DISTRIBUTION)
\( \text{LAMNSG} \)
STANDARD DEVIATION OF LAMBDA FOR NARROW-BAND RANDOM LOADS
\( \text{LAMS} \)
SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS
\( \text{LAMSA} \)
LAMBDA FOR SUPERIMPOSED SINE LOADS -- LOWER BOUND OF \( \text{k} \)
\( \text{LAMSB} \)
LAMBDA FOR SUPERIMPOSED SINE LOADS -- UPPER BOUND OF \( \text{k} \)
\( \text{LAMSC} \)
LAMBDA FOR SUPERIMPOSED SINE LOADS COEFFICIENT OF VARIATION
\( \text{LAMSD} \)
SUPERIMPOSED SINE LOADS STRAIN GAGE ACCURACY FACTOR
\( \text{LAMSK} \)
LAMBDA FOR SUPERIMPOSED SINE LOADS \( \text{k} \) -- INDICATES VARIATION DUE TO SAMPLE SIZE
\( \text{LAMSMU} \)
MEAN OF LAMBDA FOR SUPERIMPOSED SINE LOADS (\( \text{MU} \), NORMAL DISTRIBUTION)
\( \text{LAMSSG} \)
STANDARD DEVIATION OF LAMBDA FOR SUPERIMPOSED SINE LOADS
\( \text{LAMSA} \)
SIGNAL, NORMAL DISTRIBUTION)
\( \text{LAMW} \)
SELECTED ACCURACY FACTOR FOR WELD ECCENTRICITY STRESS
\( \text{LAMW} \)
CONCENTRATION FACTOR, \( \text{Koff} \)
\( \text{LAMWA} \)
LAMW LOWER BOUND
\( \text{LAMWB} \)
LAMW UPPER BOUND
\( \text{LDNAME}() \)
1-D ARRAY CONTAINING LOAD NAMES FOR THE TIME-VARYING LOADS
\( \text{LIFE}() \)
1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
\( \text{LIN}()
 -- SORTED VALUES OF THE LEFT-HAND TAIL
\( \text{LNA}() \)
1-D ARRAY CONTAINING \( \ln(A) = \ln(BIGK) \)*MIN FOR EACH REGION
\( \text{LNA}() \)
NORMAL (0,PVAR) DISTRIBUTED FOR EACH REGION
\( \text{LOCAT} \)
LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE DUCT, AND 2 IS THE INTERIOR SURFACE OF THE DUCT
\( \text{LPHIM}() \)
1-D ARRAY CONTAINING \( \ln(\Phi) \)*MIN FOR EACH REGION
\( \text{M}() \)
2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- \( M(1,* \)
\)
\( \text{MAXBLF} \)
MAXIMUM NUMBER OF LIVES TO BE CALCULATED
\( \text{MAXDAT} \)
MAXIMUM NUMBER OF POINTS PER DATA SET PER REGION ALLOWED
\( \text{MAXD} \)
MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
\( \text{MAXLF} \)
MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA CALCULATION
\( \text{MAXM} \)
MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
\( \text{MAXMM} \)
MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
\( \text{MAXSEG} \)
MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN CURVE)
NUMBER OF M'S TO BE USED TO CALCULATE MEDIAN S/N CURVE

1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION

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

2-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

2-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION

1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION

NUMBER OF SLICES TO BE CALCULATED

1-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET

SELECTED NEUBER'S RULE MODEL ACCURACY FACTOR

NEUBER'S RULE MODEL ACCURACY FACTOR LOWER BOUND

NEUBER'S RULE MODEL ACCURACY FACTOR UPPER BOUND

LIFE VALUE RETURNED FROM CALL TO THWELD

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

SIZE OF OUTER LOOP

SIZE OF INNER LOOP

TOTAL NUMBER OF LIVES CALCULATED BY PFM

NUMBER OF TIME-VARYING LOADS

CONTROLS MEDIAN CALCULATION FOR THE NORMAL DISTRIBUTION CASE

0 - NO MEDIAN CALCULATION; 1 - MEDIAN CALCULATION DESIRED

RANGES OF A/D AND S/N DATA SET TIMES UNIFORM, SOMETIMES NORMAL) USED TO OBTAIN SELECTED TEMPERATURES AND PRESSURE

1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET

NUMBER OF POINTS IN STRESS-TIME HISTORY (Number of Random Points)

POISSON'S RATIO

NUMBER OF REGIONS OF INTEREST

NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE

1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS

SELECTED INTERNAL PRESSURE, PSI

SELECTED MEAN OF INTERNAL PRESSURE, PSI (MU, NORMAL DISTRIBUTION)

MEAN OF INTERNAL PRESSURE LOWER BOUND

MEAN OF INTERNAL PRESSURE UPPER BOUND

EXTERNAL PRESSURE, PSI

SELECTED STANDARD DEVIATION OF INTERNAL PRESSURE, PSI (SIGMA, NORMAL DISTRIBUTION)

STANDARD DEVIATION OF INTERNAL PRESSURE LOWER BOUND

STANDARD DEVIATION OF INTERNAL PRESSURE UPPER BOUND

LENGTH OF TIME IN SECONDS OF RANDOM STRESS-TIME HISTORY

WEIBULL(BETA, ETAO) GENERATED RANDOM VARIATE

CONSTANT FOR THE VALUE 3.1415926536

1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS SCALED BY LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY AT_TIE() ) EQUAL TO SQRT(PVAR) -- MATERIALS PROCESS STANDARD DEVIATION

EQUAL TO SQRT(PVAR) -- MATERIALS PROCESS STANDARD DEVIATION

STATIC AXIAL LOAD SCALLED BY AERS -- MATERIALS PROCESS STANDARD DEVIATION

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 VALUES OF R* USED TO FIND STRESS CONCENTRATION DUE TO WELD ECCENTRICITY

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 OF PRODUCT OF STRESS AND STRAIN FOR EACH SEGMENT OF THE STRESS-STRAIN VS STRAIN CURVE

1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

2-D ARRAY CONTAINING STRESS POINTS (STRESS RATIO = -1.0)
FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO
LIFE REGIONS
STRTS() 2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING
STRESS-TIME HISTORIES
SZERO STRESS TENSILE TEST POINT, SO
T() 1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
TEST UNIFORM(0,1) RANDOM VARIATE USED TO DETERMINE
A HYPER-DISTRIBUTION TO SELECT FROM
THIC SELECTED WALL THICKNESS AT BEND (ID) AT WELD, IN
THICA WALL THICKNESS AT BEND (ID) LOWER BOUND
THICB WALL THICKNESS AT BEND (ID) UPPER BOUND
THICR SELECTED RHO FOR WALL THICKNESS AT BEND (ID)
THICR1 WALL THICKNESS AT BEND (ID) - RHO LOWER BOUND
THICR2 WALL THICKNESS AT BEND (ID) - RHO UPPER BOUND
THICT SELECTED THETA FOR WALL THICKNESS AT BEND (ID)
THICT1 WALL THICKNESS AT BEND (ID) - THETA LOWER BOUND
THICT2 WALL THICKNESS AT BEND (ID) - THETA UPPER BOUND
THWELD REAL FUNCTION WHICH CALCULATES THE DUCT LIFE (IN SECONDS)
AT A PLAIN WELD UNDERGOING THERMAL LOADING
SELECTED INNER WALL SURFACE TEMPERATURE (RANKINE)
SELECTED MEAN OF INNER WALL TEMPERATURE, (MU, NORMAL
DISTRIBUTION)
SELECTED MEAN OF INNER WALL TEMPERATURE UPPER BOUND
SELECTED STANDARD DEVIATION OF INNER WALL TEMPERATURE, (SIGMA,
NORMAL DISTRIBUTION)
SELECTED STANDARD DEVIATION OF INNER WALL TEMPERATURE LOWER BOUND
1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS SCALED
BY LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY TYPE())
SELECTED OUTER WALL SURFACE TEMPERATURE (RANKINE)
SELECTED MEAN OF OUTER WALL TEMPERATURE (MU, NORMAL
DISTRIBUTION)
SELECTED MEAN OF OUTER WALL TEMPERATURE UPPER BOUND
SELECTED STANDARD DEVIATION OF OUTER WALL TEMPERATURE, (SIGMA,
NORMAL DISTRIBUTION)
SELECTED STANDARD DEVIATION OF OUTER WALL TEMPERATURE LOWER BOUND
1-D ARRAY CONTAINING VALUES OF SBND * PHI * KRATIO * 2 FOR
EACH REGION AND TRIAL
TRUNC VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
TSLAM STATIC AXIAL LOAD SCALING FACTORS -- TYPE(*) .. 1 INDICATES NARROW-BAND RANDOM;
TYPE(*) .. 2 INDICATES SUPERIMPOSED SINUSOID; TYPE(*) .. 3 INDICATES DYNAMIC AERO
V(1,*) 2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
V(2,*) ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
1 - 2/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
VSLAM() 2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS SCALED
BY LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY TYPE())
VSND() 1-D ARRAY CONTAINING THE STATIC SHEAR LOADS SCALED BY AERD
VSTAT() 1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- VSTAT(1) IS
THE V2 LOAD; VSTAT(2) IS THE V3 LOAD
WOFF SELECTED WELD OFFSET (%)
WOFFA WELD OFFSET LOWER BOUND - HYPER-DISTRIBUTION 1
WOFFB WELD OFFSET UPPER BOUND - HYPER-DISTRIBUTION 1
WOFFC WELD OFFSET LOWER BOUND - HYPER-DISTRIBUTION 2
WOFFD WELD OFFSET UPPER BOUND - HYPER-DISTRIBUTION 2
WOFFE PERCENTAGE OCCURRING IN HYPER-DISTRIBUTION 1
WOFFFL SELECTED WELD OFFSET UPPER BOUND
WOFFLO SELECTED WELD OFFSET LOWER BOUND
WOFFF SELECTED RHO FOR WELD OFFSET
WOFFR SELECTED RHO FOR WELD OFFSET
WOFFR1 WELD OFFSET - RHO LOWER BOUND - HYPER-DISTRIBUTION 1
WOFFR2 WELD OFFSET - RHO UPPER BOUND - HYPER-DISTRIBUTION 2
WOFFS WELD OFFSET - RHO LOWER BOUND - HYPER-DISTRIBUTION 2
WOFFT WELD OFFSET - RHO UPPER BOUND - HYPER-DISTRIBUTION 2
WOFT SELECTED THETA FOR WELD OFFSET
WOFT1 WELD OFFSET - THETA LOWER BOUND - HYPER-DISTRIBUTION 1
SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

INPUTS: NEWLIF, LIFE, NLIFET

OUTPUTS: LIFE

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

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

STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()
DO 200 I = (PLACE + 1) , NUM
    TEMP(I) = LIFE(I-1)
200 CONTINUE
C  INSERT NEWLIF
    LIFE(PLACE) = NEWLIF
C  SHIFT VALUES OF LIFE() FOLLOWING NEWLIF
    DO 300 I = (PLACE + 1) , NUM
      LIFE(I) = TEMP(I)
    300 CONTINUE
C  IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN
400 CONTINUE
RETURN
END

**********************************************************************
SUBROUTINE PRYRV (RAND, RHOI, 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, ' RHO2 = ', RHO2,
RETURN
END

**********************************************************************
*/

SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
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 NAS7-915
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 RAND (FRAC, RAND)
C IF (IOUT .EQ. 15) WRITE (8, *) 'FRAC = ', FRAC
  X = FRAC * (RHO2 - RHO1) + RHO1
CALL RAND (FRAC, RAND)
C IF (IOUT .EQ. 15) WRITE (8, *) 'FRAC = ', FRAC
  Y = FRAC * (THE2 - THE1) + THE1
  IF (IOUT .EQ. 15) WRITE (8, *) 'RHO1 = ', RHO1, ' RHO2 = ', RHO2,
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.

C

7 - 166
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)
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, UI, U2, V1, V2
DOUBLE PRECISION RAND
A = ALPHA - 1.
IF (IOUT .EQ. 15) WRITE(8,*) 'A=', A, ' ALPHA=', ALPHA
10 CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALOG(U1)
V2 = - ALOG(U2)
IF (IOUT .EQ. 15) WRITE(8,*) 'U1=', U1, ' U2=', U2, ' V1=',
& ' V2=', V2
ARG = A * (V1 - ALOG(V1) - 1.)
IF (V2 .LT. ARG) GOTO 10
GAM = ALPHA * V1
IF (IOUT .EQ. 15) WRITE(8,*) 'GAMMA=', GAM
RETURN
END

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

7 - 167

C-3
SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, REFPF, SZERO, ZROREG, & NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC, & KRATIO, PVAR)

INPUTS: READS DATA FROM SPECFD AND RELATD; VARY, MPROC
OUTPUTS: RANGEM, MU, SIG, NF, REFPF, SZERO, ZROREG, NUMREG, NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
SUBPROGRAMS: INIT, REL, SW2SU2, FINDMC, INTRVL, FINDRNG, ADDRREG, CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR
FILES: 5:RELATD-OLD; 6:RELATD-NEW

IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNOODAT, & NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET), & NSETS, NUMREG, REFPF(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), MC(2, MAXREG), & MCHAT(2, MAXREG), NEDN(MAXREG), MU(MAXREG), & MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG), & PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET), & RANR(MAXDAT, 0:MAXSET), RAWFN(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

LIST OF VARIABLES

BIGKHT: EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
BZERO: VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE S/N DATA SET
CZERO: EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, CO
DD(): 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
DELTA(): 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
FTUZ: YIELD 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(RANFN()), ALSO INDEXED FOR REGION
LNSTR(): 3-D ARRAY CONTAINING LN(RAYSTR()), ALSO INDEXED FOR REGION
MAXDAT: MAXIMUM NUMBER OF POINTS IN S/N DATA SET PER REGION ALLOWED
MAXREG: MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET: MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MC(): 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA — MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCHAT(): 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
MCHAP(1,L) = -DO, THE ESTIMATE FOR M AND
MCHAP(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
M() FOR EACH REGION
MEDM() 1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MZERO() FOR EACH REGION
MPROC() Materials Process variation -- CONTROLS MATERIALS PROCESS
VARIATION -- 0 = NO VARIATION; 1 = VARIATION
MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION MEAN FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
EACH REGION -- MZERO(1,L) IS THE UPPER BOUND AND MZERO(2,L)
IS THE LOWER 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
NODAT 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
NPFR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1)))-1 OVER
ALL DATA SETS IN A REGION (Number of Points Per Region)
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
NSETS() NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMREG() NUMBER OF REGIONS OF INTEREST
PVAR() MATERIALS PROCESS VARIATION
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND
RATSTR() 2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
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
SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
SIGA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
VARIANCE FOR EACH REGION
STR() 2-D ARRAY CONTAINING RANSTR() 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
SY() 1-D ARRAY CONTAINING SAMPLE Y COVARIANCE FOR EACH REGION
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
ZROREG Zero Region -- VALUE CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 = ZERO REGION EXISTS, 1 = NO ZERO REGION

OPEN(5, FILE = 'RELATO', STATUS = 'OLD')
OPEN(6, FILE = 'RELATO', STATUS = 'NEW')
C RELATO 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, ZERO, ZEROG, NUMREG, NNODAT, 
& NSETS, NBND, ZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, 
& SIGMA2, KRATIO, LAMN)

C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD, 
& SWHAT2, SUHAT2, NPPR)

C CALCULATE M CONSTRAINT 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, 
& JSERO, 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, 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 = i, NUOREG
WRITE(7,905) L, IZERO(1, L), IZERO(2, L), 
& JZERO(1, L), JZERO(2, L)
25 CONTINUE

WRITE(7,910)
DO 50 L = 1, NUMREG
WRITE(7,915) L, MCHAT(2, L), MCHAT(1, L)
50 CONTINUE

IF (CZERO .GT. 0.0) THEN
WRITE(7,960)
DO 150 L = 1, NUMREG
IF (MCPNT(L) .EQ. 1) THEN
WRITE(7,965) L, MC(1, L)
ELSEIF (MCPNT(L) .EQ. 2) THEN
WRITE(7,970) L, MC(1, L), MC(2, L)
ENDIF
150 CONTINUE
CONTINUE
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 100 L = 1, NUMREG
   WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
100 CONTINUE
WRITE(7,950)

C CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO
CALL MEDIAN (NUMREG, RANGEM, MEDM)

C CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
   CALL EXPCTD (1, MEDM, REFPNP, STR, NF, SZERO, NUMREG, ZROREG, &
   NSND, BIGHRT, 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
      NORMAL VARIATION IS DESIRED
      CALL MUSIG (NUMREG, SX2, DD, SUHAT2, S2HAT2, NPPR, DELTA, MO, &
      SIGMA2, MCHAT, MU, SIG)
   ENDIF

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, ZERO, MCPNT, MC, MCHAT, RANGEM)

C ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
   CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, &
   MPNT, MO, SIGMA2)

C ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
   CALL CONCAV (NUMREG, RANGEM)

C WRITE RESULTS TO FILE DUMP
WRITE(7,975)
DO 350 L = 1, NUMREG
   WRITE(7,980) L, MCHAT(I,L)
350 CONTINUE

IF (CZERO .GT. 0.0) THEN
   WRITE(7,960)
   DO 360 L = 1, NUMREG
      IF (MCPNT(L) .EQ. 1) THEN
         WRITE(7,965) L, MC(I,L)
      ELSEIF (MCPNT(L) .EQ. 2) THEN
         WRITE(7,970) L, MC(1,L), MC(2,L)
      ENDIF
   CONTINUE
ENDIF
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 370 I = 1, NUMREG
  WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
CONTINUE
WRITE(7,950)
WRITE(7,985)
DO 380 L = 1, NUMREG
  WRITE(7,990) L, MU(L), SIG(L)
CONTINUE
ENDIF
C PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS
IF (MPROC .EQ. 1) THEN
  WRITE(7,995) PVAR
ENDIF
C FORMAT STATEMENTS
900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
  & 'Technology. U.S. Government',/2X,'Sponsorship under ',
  & 'NASA Contract NAS7-918 is acknowledged.///,
  & 'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
  & '95% CONFIDENCE INTERVALS ON C AND m ',
  & 'FOR EACH REGION',/)
905 FORMAT(7X,'REGION: ',I1,7X,'Io = (',F12.9,',',F12.9,')',
  & '/,24X,'Jo = (',F12.9,',',F12.9,')')
910 FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
  & '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',
  & '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.///,
  & 'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
  & 'ESTIMATE OF m FOR EACH REGION',
  & 'REGION',12X,'E(m)',/)
980 FORMAT(9X,I1,11X,F10.6)
985 FORMAT(2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS',
  & 'REGION',5X,'MEAN',8X,'STD DEV',/)
990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)
995 FORMAT(///)
& 'INFORMATION',//.7X,E11.5)

RETURN
END

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

C SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
PROGRAMMER: L. NEWLIN
DATE: 5OCT87
VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
         V8.4, V8.5

SUBROUTINE TRMNAT
WRITE (8,*) 'PROGRAM EXECUTION TERMINATED'
STOP
END

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

C SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
PROGRAMMER: L. NEWLIN
DATE: CODE: 21JUN88 COMMENTS: 13JUL89

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

C INPUTS: ---
C OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LINST, REFNP,
            NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2
C
C IMPLICIT NONE
C
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),
               & LINST(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
               & MZERO(2, MAXREG), NF(MAXDAT, MAXREG)
               & RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
               & SIGMA2(MAXREG),
               & STR(MAXDAT, MAXREG)

LIST OF VARIABLES

DELTA()  1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
         SIG() CALCULATION
I         CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT      OUTPUT DUMP CONTROLLER
J         CONTROLS DO LOOP FOR EACH DATA SET
K         CONTROLS DO LOOP FOR EACH POINT IN A REGION
L         CONTROLS DO LOOP FOR EACH REGION
LNNF()    3-D ARRAY CONTAINING LN(RAWNF()), ALSO Indexed FOR REGION

7 - 173
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 I = 1, MAXREG
   DO 425 K = 1, MAXDAT
      DO 450 J = 0, MAXSET
         LNNF(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(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

C INSTR()  3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
C MAXREG  MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C MAXSET  MAXIMUM NUMBER OF REGIONS ALLOWED
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 NORMAL DISTRIBUTION
C MEAN FOR EACH REGION
C NF()  2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C NP()  2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C IN EACH REGION
C NPTS()  1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C RATSTR()  2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
C STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
C RAWNF() 2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C DATA SETS
C RAWSTR() 2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OF TOTAL STRAIN
C DATA (%) FOR ALL S/N DATA SETS
C REFNP() 1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C REFERENCE MATERIAL S/N DATA SET IN EACH REGION
C SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C VARIANCE FOR EACH REGION
C STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C DATA SET BROKEN INTO REGIONS (PSI OR %)
SUBROUTINE RCE "READS" THE DATA FROM SPECIFIED RELAXED, "CONVERTS" THE STRESS DATA TO A STRESS RATIO OF -1.0, AND "ECHOES" THE DATA TO SPECIFIED RELAXED. 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, & LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG, & NUMREG, NNDAT, NSETS, NNBND, CZERO, MPNT, MZERO, & FTUZ, FTYZ, DELTA, MO, SIGMA2, KRAITO, LAMN)

INPUTS: VARY, MPROC
OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NNDAT, NSETS, NNBND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2, KRAITO, LAMN

SUBPROGRAMS: TRMNAT, CONVRT

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, & NNDAT, NF(0:MAXSET, MAXREG), NPTS(0:MAXSET), NSETS, & NUM, NUMREG, REFNP(MAXREG), REG, VARY, ZROREG
REAL CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ, & KRAITO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG), & LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG), & MZERO(2, MAXREG), NBBND(0:MAXREG), NF(MAXDAT, MAXREG), & RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET), & STR(MAXDAT, MAXREG), SZERO
CHARACTER*40 DESCRP(0:MAXSET)

LIST OF VARIABLES

INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO, CONVERSION, AND BREAK UP
EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, CO
1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
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
LAMBDA-N -- RATIO OF VAR (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
MAXDAT
MAXREG
MAXSET
MON()
MPNT()
MPROC
MZERO()
NBND()
NDV()
NFD()
NNODAT
NP()
NP()
NPS()
NNETS
NUM()
NUMREG
RATIO
RATSTR()
RAWF()
RAWSTR()
REFNP()
REG
SIGMA2()
STR()
SZERO
VARY
ZROREG

MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXIMUM NUMBER OF REGIONS ALLOWED
MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MATERIALS PROCESS VARIATION -- CONTROLS MATERIALS PROCESS VARIATION -- 0 -- NO VARIATION; 1 -- VARIATION
2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO, REGION PAIRS DURING INPUT
2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
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
NUMBER OF REGIONS 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
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

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 SPECPO

READ(1,*) DESCRP(0), FTU, FTY, 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, FTY, NPTS(0)
WRITE(3,905)
C STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ

7 - 176
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(I,*) RAWSTR(I,0), RAWNF(I,0)
      CONTINUE
C CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
C IF (RATIO .EQ. -1.0) THEN
C STRESS RATIO IS CORRECT
   DO 120 I = (COUNT + 1), (COUNT + NUM)
      RATSTR(I,0) = RAWSTR(I,0)
   CONTINUE
ELSE
C STRESS RATIO TRANSFORMATION MUST BE DONE
C CALL CONVERT (0, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR, RATIO, FTU, FTY)
C ENDIF
C ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL
DO 130 I = (COUNT + 1), (COUNT + NUM)
   WRITE(3,910) RAWSTR(I, 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)
130 CONTINUE
C BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2, EFCFD, AND PAREST
K = NP(0,REG)
DO 140 I = (COUNT + 1), (COUNT + NUM)
   K = K + 1
   LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
   LNHF(K,0,REG) = ALOG(RAWNF(I,0))
   STR(K,REG) = RATSTR(I,0)
   NF(K,REG) = RAWNF(I,0)
140 CONTINUE
C IF (K .GT. MAXDAT) THEN
   WRITE(8,*), 'ERROR: OVER NUMBER OF POINTS LIMIT IN ',
      'SPECIFIC MATERIAL'
   CALL TRMNAT
ENDIF
NP(0,REG) = K
REPNP(REG) = K
COUNT = COUNT + NUM

100 CONTINUE

IF (NPTS(0) .NE. COUNT) THEN
  WRITE(8,*) 'ERROR: NUMBER OF POINTS PER DIVISION ',
& ' INCORRECTLY SPECIFIED'
  WRITE(6,*) '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) THEN
  WRITE(8,*) 'ERROR: NUMBER OF POINTS PER DMSION ',
& ' INCORRECTLY SPECIFIED'
ENDIF

READ(I,*) SZERO
IF (NINT(SZERO) .GT. 0) THEN
  ZROREG = 0
ELSE
  ZROREG = 1
ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'ERROR: NUMBER OF POINTS PER DMSION ',
& ' INCORRECTLY SPECIFIED'
ENDIF

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,*) NZERO
  150 CONTINUE

READ(1,*) CZERO

DO 160 L = 1, (NUMREG + NNODAT)
  READ(I,*) MPNT(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(6,*) 'ERROR: NUMBER OF POINTS PER DMSION ',
& ' INCORRECTLY SPECIFIED'
ENDIF

WRITE(3,915) NUMREG, NNODAT
IF (IOUT .EQ. 10) WRITE(6,*) 'ERROR: INCORRECTLY SPECIFIED'

DO 170 L = ZROREG, (NUMREG + NNODAT)
  WRITE(3,920) NZERO
  170 CONTINUE

WRITE(3,925) CZERO
IF (IOUT .EQ. 10) WRITE(8,925) CZERO

DO 180 L = 1, (NUMREG + NNODAT)
  WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
  180 CONTINUE

WRITE(3,935)

IF (IOUT .EQ. 10) WRITE(8,935)

IF ((VARY .EQ. 3) .AND. (MPNT(L) .EQ. 0)) THEN
  WRITE(6,*) 'ERROR: NORMAL VARIATION REQUIRES A PRIOR RANGE ON M'
  CALL TRMNAT
ENDIF

180 CONTINUE

IF (VARY .EQ. 3) THEN
  IF (IOUT .EQ. 10) WRITE(8,945)
  WRITE(3,945)
ENDIF
DO 190 L = 1, (NUMREG + NWODAT)
READ(L,*) DELTA(L), MO(L), SIGMA2(L)
WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
IF (IOUT .EQ. 10) 
& WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
& IF ((DELTA(L) .LT. 0.0) .OR. 
& ((DELTA(L) .GT. 0.0) .AND. (MO(L) .LE. 0.0))) THEN
& WRITE(8,*) 'ERROR: BAD VALUE FOR DELTA OR VALUE OF MO '
& 'INCONSISTENT WITH DELTA IN REGION ', L
& CALL TRMNAT
ENDIF
CONTINUE
190 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,*) 'ERROR: OVER LIMIT ON NUMBER OF RELATED DATA SETS'
CALL TRMNAT
ENDIF
WRITE(6,935) NSETS
DO 200 J = 1, NSETS
COUNT = 0
IF (IOUT .EQ. 10) WRITE(8,*) J =', J, ' NSETS =', NSETS
READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)
IF (NPTS(J) .GT. MAXDAT) THEN
WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF POINTS IN ', 
' SET ', J
CALL TRMNAT
ENDIF
WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)
WRITE(6,905)
IF (IOUT .EQ. 10) WRITE(8,905)
DO 300 M = 1, NDIV
READ(5,*) NUM, RATIO, REG
IF (ABS(RATIO) .GT. 1.0) THEN
WRITE(8,*) 'ERROR: INVALID VALUE OF RATIO: ', RATIO
CALL TRMNAT
ENDIF
IF (REG .GT. MAXREG) THEN
WRITE(8,*) 'ERROR: OVER REGION LIMIT IN RELATED MATERIAL ', J
CALL TRMNAT
ENDIF
IF (IOUT .EQ. 10) THEN
WRITE(8,*) NUM = ',', NUM, ' COUNT = ', COUNT
WRITE(8,*) RATIO = ',', RATIO, ' REG = ', REG
ENDIF
DO 310 I = (COUNT + 1), (COUNT + NUM)
C CHECK IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
IF (RATIO .EQ. -1.0) THEN

C STRESS RATIO IS CORRECT
DO 320 I = (COUNT + 1), (COUNT + NUM)
   RATSTR(I,J) = RAWSTR(I,J)
320 CONTINUE

ELSE

C STRESS RATIO TRANSFORMATION MUST BE DONE
CALL CONVRT(J, (COUNT + 1), (COUNT + NUM), RAWSTR, RATSTR, RATIO, FTU, FTY)
ENDIF

C RECORD BOTH S/N DATA SETS TO RELATO
DO 330 I = (COUNT + 1), (COUNT + NUM)
   WRITE(6,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG, RATSTR(I,J), RAWNF(I,J)
   IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,J), RAWNF(I,J), RATIO, REG, RATSTR(I,J), RAWNF(I,J)
330 CONTINUE

K = NP(J,REG)
DO 340 I = (COUNT + 1), (COUNT + NUM)
   K = K + 1
   LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
   LNNF(K,J,REG) = ALOG(RAWNF(I,J))
340 CONTINUE

IF (K .GT. MAXDAT) THEN
   WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF POINTS ', 'IN SET ', J
   CALL TRMNAT
ENDIF

NP(J,REG) = K
COUNT = COUNT + NUM
300 CONTINUE

IF (NPTS(J) .NE. COUNT) THEN
   WRITE(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,12)
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)
THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
STRESS RATIO, R, IS NOT -1.0
PROGRAMMER: L. NEWLIN
DATE: CODE: 6OCT87 COMMENTS: 13JUL89
VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
V8.3, V8.4, V8.5

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

INPUTS: J, NUM1, NUM2, STR, R, FTU, FTY

OUTPUTS: RSTR

IMPLICIT NONE

INTEGER MAXDAT, MAXSET
PARAMETER (MAXDAT = 50, MAXSET = 5)

COMMON IOUT

INTEGER I, IOUT, J, NUM1, NUM2

REAL FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET),
& STR(MAXDAT, 0:MAXSET), TEST

LIST OF VARIABLES

...
SUBROUTINE SW2SU2 CALCULATES, SWHAT2, 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

SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SY2, DD, SWHAT2, SUHAT2, NPPR)

C SUBROUTINE SW2SU2 CALCULATES, SWHAT2, 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
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87
SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SY2, DD, SWHAT2, SUHAT2, NPPR)
C INPUTS: NUMREG, NSETS, NP, LNSTR, LNNF
C OUTPUTS: SX2, SY2, DD, SWHAT2, SUHAT2, NPPR
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), DD(MAXREG), DIFFX(MAXDAT, 0:MAXSET), & DIFFY(MAXDAT, 0:MAXSET), LNNF(MAXDAT, 0:MAXSET, MAXREG), & LNSTR(MAXDAT, 0:MAXSET, MAXREG), MEANX(0:MAXSET), & MEANJ(0:MAXSET), SUHAT2(MAXREG), SWAT2(MAXREG), & SX2(MAXREG), SKY(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) AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
DIFFY() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L) AND MEANJ(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
L CONTROLS DO LOOP FOR EACH REGION
LNNF() 3-D ARRAY CONTAINING LN(RANFN()), 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
C MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MEANX() 1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION L AND DATA SET J (X = Ln S)
MEANY() 1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION L AND DATA SET J (Y = Ln N)
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL DATA SETS IN A REGION (NUMBER OF POINTS PER REGION)
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMREG NUMBER OF REGIONS OF INTEREST
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SWAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION (Y = Ln N)

C INITIALIZE ARRAYS
  DO 50 L = 1, MAXREG
       SY2(L) = 0.0
       SX2(L) = 0.0
       SKY(L) = 0.0
       SWAT2(L) = 0.0
       SUHAT2(L) = 0.0
       BB(L) = 0.0
       DD(L) = 0.0
       NPPR(L) = 0
  50 CONTINUE

  DO 60 J = 0, MAXSET
     DO 70 K = 1, MAXDAT
          DIFFX(K,J) = 0.0
          DIFFY(K,J) = 0.0
     70 CONTINUE
     MEANX(J) = 0.0
     MEANY(J) = 0.0
  60 CONTINUE
CONTINUE

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

DO 100 L = 1, NUMREG
   
   DO 200 J = 0, NSETS
      
      FIRST CALCULATE SAMPLE X AND Y MEANS
      FOR DATA SET J IN REGION L
      MEANY(J) = 0.0
      MEANX(J) = 0.0
      
      IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' J =', J,
         & ' NP =', NP(J,L)
      
      DO 250 K = 1, NP(J,L)
         
         DIFFY(K,J) = LNNF(K,J,L) - MEANY(J)
         DIFFX(K,J) = LNSTR(K,J,L) - MEANX(J)
         
         SY2(L) = SY2(L) + DIFFY(K,J) ** 2
         SX2(L) = SX2(L) + DIFFX(K,J) ** 2
         SXY(L) = SXY(L) + DIFFX(K,J) * DIFFY(K,J)
      
      IF (IOUT .EQ. 10) WRITE(8,*) 'K =', K, DIFFY(K,J),
         & ' DIFFX(K,J)
      
      WRITE(8,*) 'SY2(L) =', SY2(L), ' SX2(L) =', SX2(L),
         & ' SXY(L) =', SXY(L)
      
      ENDIF
   
   NPPR(L) = NPPR(L) + NP(J,L) - 1
   
   IF (IOUT .EQ. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
      & ' SX2(L) =', SX2(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) = SY2(L) / SY2(L)

IF (IOUT .EQ. 10) THEN
   WRITE(8,*) 'NPPLR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
      & ' SX2(L) =', SX2(L)
ENDIF
SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NFPR, IZERO, JZERO, MCHAT)
C
INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NFPR
OUTPUTS: IZERO, JZERO, MCHAT
SUBPROGRAMS: TRMNat
C
IMPLICIT NONE
INTEGER CHITAB, MAXREG, TTAB
PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)
COMMON IOUT
INTEGER I, IOUT, L, NFPR(MAXREG), NUM, NUMREG
REAL ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
& IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
& SX2(MAXREG), SX, T025(TTAB)
DATA (CHI025(I), I = 1, 75) /
& 0.000982669, 0.506356, 0.215795, 0.484419, 0.831211,
& 1.237347, 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, 18.28, 19.04, 19.80, 20.56,
& 21.33, 22.10, 22.87, 23.65, 24.4331,
& 25.21, 25.99, 26.78, 27.57, 28.36,

END
&
&
&
&
&
&

29.95,
33.96,
38.02,
42.12,
46.26,
50.42,

30.75,
34.77,
38.84,
42.95,
47.09,
51.26,

31.55,
35.58,
39.66,
43.77,
47.92,
52.10,

32.3574,
36.39,
40.4817,
44.60
48.75_6,
52.94
/

(CHI025(I)'.I8"53
7
76,
57.80
62.24
66.50
70.78
75.08
79.40
83.73
88.08
92.45
96.82
101.2
,
105.61,
110.02,
114.44,

150)
/
54.62,
58.84,
63.09,
67.35,
71.64,
75.94,
80.27,
84.60,
88.95,
93.32,
97.70
102.0§,
106.49,
110.90,
115.33,

55.46
59.69
63.94
68.21
72.50
76.80
81.13
85.47
89.83
94.19
98.57
102.9,
107.37,
111.79,
116.21,

56.30,
60.54,
64.79,
69.07,
73.36,
77.67,
82.00,
86.34,
90.70,
95.07,
99.45
103.8_,
108.25,
112.67,
117.10,

57.1532,
61.39
65.64_6,
69.92,
74.2219,
78.53,
82.87,
87.21,
91.57,
95.94,
100.33,
104.73,
109.14,
113.56,
117.98
/

(CHI975(I),

75)
/
7.37776
16.0128
23.3367
30.1910
36.7807
43.1944
49.48
55.67
61.77
67.82
73.81
79.75
85.65
91.52
97.35

9.34840,
17.5346,
24.7356,
31.5264,
38.0757,
44.4607,
50.72,
56.89,
62.99,
69.02,
75.00,
80.93,
86.83,
92.69,
98.52,

11.1433
19.0228
26.1190
32.8523
39.3641
45.7222
51.96,
58.12,
64.20,
70.22,
76.19,
82.12,
88.00,
93.86,
99.68,

12.8325,
20.4831,
27.4884,
34.1696,
40.6465,
46.9792,
53.20
59.3417,
65.41
71.4262,
77.38
83.29_6,
89.18,
95.0231,
100.84
/

104.31
110.09
115.84
121.57
127.28
132.98
138.65
144.31
149.96
155.59
161.21
166.82
172.41
178.00
183.58

105.47,
111.24,
116.99,
122.72,
128.42,
134.11,
139.79,
145.44,
151.09,
156.72,
162.33,
167.94,
173.53,
179.12,
184.69,

106.629,
112.39,
118.136,
123.86,
129.561,
135.25,
140.92,
146.57,
152.21,
157.84,
163.46,
169.06,
174.65,
180.23,
185.80
/

29.15,
33.16,
37.21,
41.30,
45.43,
49.59,
DATA

&
&
&
&
&
&
&
&
&
&
&
&
&
&
&
DATA
&
&
&
&
&
&
&
&
&
&
&
&
&
&
&

I = i,
5.02389,
14.4494,
21.9200,
28.8454,
35.4789,
41.9232,
48.23,
54.44,
60.56,
66.62,
72.61,
78.57,
84.48,
90.35,
96.19,

DATA

C
C
C
C
C
C

VALUES
1 -

DATA

C
C

FOR

THE

TABLES

30,

40,

50,

31-39,
--

&
&
&
&

76,

(CHI975(I)'210=I0. 0
107.78
113.54
119.28
125.00
130.70
136.38
142.05
147.70
153.34
158.97
164.58
170.18
175.77
181.35

&
&
&
&
&
&
&
&
&
&
&
&
&
&
&

150)
103.16
108.94
114.69
120.43
126.14
131.84
137.52
143.18
148.83
154.47
160.09
165.70
171.30
176.88
182.46

ABOVE

60,

70,

WERE
80,

41-49,
51-59,
61-69,
CALCULATED
USING
CUBE

T025

/

12.706,
2.365,
2.160,
2.093,
2.060,

4.303,
2.306,
2.145,
2.086,
2.056,

LIST

OF

/

OBTAINED
90,

100

IN
--

THE

Theil,

FOLLOWING
pp.

71-79,
81-89,
91-99,
RULE
APPROXIMATION

718-719
101-150

3.182,
2.262,

2.776,
2.228,

_.571,
.201,

_.447,
.179,

2.131,
2.080,
2.052,

2.120,
2.074,
2.048,

_.110,
.069,
2.045,

_.101,
.064,
2.042,

VARIABLES

7-186

MANNER:

1.960

/


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() 2-D ARRAY CONTAINING SX?(L)/SX2(L) FOR EACH REGION
I CONTROLS LOOP FOR CHI025() AND CHI975()
IZERO() OUTPUT DUMP CONTROLLER
IZERO() 2-D ARRAY CONTAINING I0, THE 95% CONFIDENCE INTERVALS ON Q
FOR EACH REGION
JZERO() 2-D ARRAY CONTAINING J0, THE 95% CONFIDENCE INTERVALS ON M
FOR EACH REGION
L MAXREG CONTROLS DO LOOP FOR EACH REGION
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))**0.5) 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
SUHAT EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
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
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
SX = (NUM*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
C
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
C
DO 75 L = 1, NUMREG
IF (NPPR(L) .GT. CHITAB) THEN
WRITE(8,*) 'ERROR: EXCEEDED LIMIT ON DEGREES OF FREEDOM ',
'IN CHI-SQUARE TABLE, IN REGION ', L
CALL TRMNAT
ENDIF
75 CONTINUE

C ASSIGN VALUES TO NUM, T, SWHAT, SUHAT AND THEN CALCULATE
C CONFIDENCE INTERVALS FOR EACH REGION
C
DO 100 L = 1, NUMREG
NUM = NPPR(L)
IF (NUM .LT. 31) THEN
T = T025(NUM)
ELSE
T = T025(NUM)
ENDIF
SWHAT = SWHAT2(L) ** 0.5
SUHAT = SUHAT2(L) ** 0.5
SX = (NUM*SX2(L)) ** 0.5
C
C CALCULATE ESTIMATED VALUES OF M AND C

7 - 187
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) / CHI025(NUM)) ** 0.5
IZERO(2, L) = MCHAT(1, L) - ARG
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, *) '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) = ',
& 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******************************************************************************************************************************************************************


INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), NUMREG
REAL ARGl, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
& SXY(MAXREG), SY2(MAXREG)

LIST OF VARIABLES

ARGl INTERMEDIATE CALCULATION VARIABLE
ARG2 INTERMEDIATE CALCULATION VARIABLE
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE C COEFFICIENT OF VARIATION, CO
CZERO2 EQUAL TO ZERO ** 2
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION

7 - 188
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) THEN
   WRITE(8,'(1X,A)') 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2
ENDIF
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
      MCPNT(L) = 2
      ELSEIF (ABS(ARG2) .LT. 1.0E-6) THEN
      MCPNT(L) = 1
      MC(1,L) = - SXY(L) / ARG1
      ELSE
      MCPNT(L) = 2
ENDIF
100 CONTINUE
MC(1,L) = (-SKY(L) - ARG2) / ARG1
MC(2,L) = (-SKY(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)
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)

C INPUTS: NSETS, NP, NUMREG, LAMN, MCHAT
C OUTPUTS: PVAR
C IMPLICIT NONE

INTEGER MAXREG, MAXSET
PARAMETER (MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG),
& NUMREG, TOTAL
REAL LAMN, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM

LIST OF VARIABLES

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
MCHAT(2,L) = SUBAT, THE ESTIMATE FOR C
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
SET IN EACH REGION

7 - 190
C NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C NUM()      NUMBER OF POINTS IN EACH DATA SET
C NUMREG     NUMBER OF REGIONS OF INTEREST
C PSIG2()    1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
            VARIATION IN EACH REGION
C PVAR       THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
              CURVE WARRANTED BY THE AVAILABLE INFORMATION
C SUM        WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
C 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,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
ENDIF

PVAR = SUM / FLOAT (TOTAL)
RETURN
END
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
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) = SUHAT, THE ESTIMATE FOR M
   AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
   MC() FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
   MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
   EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
   IS THE UPPER BOUND
NUMREG
   NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
   FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
   RANGEM(2,L) IS THE UPPER BOUND
UPPER
   UPPER BOUND OF INTERSECTION
C INITIALIZE VARIABLES

DO 50 L = 1, MAXREG
   RANGEM(1,L) = 0.0
   RANGEM(2,L) = 0.0
50 CONTINUE
C PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

DO 100 L = 1, NUMREG
   IF (IOUT .EQ. 10) THEN
      WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
      WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
      ENDIF
   IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN
      C 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), &
         ' RANGEM(2,L) = ', RANGEM(2,L), &
         ' JZERO(2,L) = ', JZERO(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

7 - 192
LOWER = AMAX1(JZERO(1,L), MC(1,L))
UPPER = JZERO(2,L)
IF (UPPER .LT. LOWER) THEN
WRITE(*,*) 'ERROR: NO INTERSECTION BETWEEN JO AND MC'
CALL TRNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF

IF (IOUT .EQ. 10) THEN
WRITE(*,*) JZERO(1,L) = ', JZERO(1,L),
WRITE(*,*) 'MC(1,L) = ', MC(1,L),
WRITE(*,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
ENDIF

ELSEIF ((MPNT(L) .EQ. 1) THEN
THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
RANGEM(1,L) = MZERO(1,L)
RANGEM(2,L) = 0.0
IF (IOUT .EQ. 10) THEN
WRITE(*,*) MZERO(1,L) = ', MZERO(1,L),
WRITE(*,*) RANGEM(1,L) = ', RANGEM(1,L),
WRITE(*,*) RANGEM(2,L) = ', RANGEM(2,L)
ENDIF
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN
THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT
USE INTERSECTION BETWEEN JO AND MO
LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
IF (UPPER .LT. LOWER) THEN
WRITE(*,*) 'ERROR: NO INTERSECTION BETWEEN JO AND MO'
CALL TRNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF
IF (IOUT .EQ. 10) THEN
WRITE(*,*) JZERO(1,L) = ', JZERO(1,L),

7 - 193
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 1)) THEN
THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO CO
CONSTRAINT, INTERSECT JO AND MO, ADJUSTING THE LOWER BOUND
BY MC ACCORDINGLY
LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
IF (UPPER .LT. LOWER) THEN
WRITE(8,*) 'ERROR: NO INTERSECTION BETWEEN JO, MO, ',
& CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF
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, ',
& CALL TRMNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF
ELSEIF (IOUT .EQ. 10) THEN
WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
& JZERO(2,L) = ', JZERO(2,L)
WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
& MZERO(2,L) = ', MZERO(2,L)
WRITE(8,*) 'MC(1,L) = ', MC(1,L),
& MC(2,L) = '..,'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

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 SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C WITHOUT DATA
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 2FEB88, FORMAT/COMMENTS: 12AUG91
C
C SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C INPUTS: RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C OUTPUTS: RANGEM, MCHAT, NUMREG
C
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)

LIST OF VARIABLES

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
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) = SUAT, THE ESTIMATE FOR C
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS UPPER BOUND
NNODAT Number of NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

7 - 195
IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG
DO 100 L = 1, NHODAT
   NUMREG = NUMREG + 1
   LL = NUMREG
   IF (IOUT .EQ. 10) WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG,
   &
   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),
       &
       'RANGEM(2,LL) = ', RANGEM(2,LL),
       &
       'MZERO(2,LL) = ', MZERO(2,LL)
     ENDIF
     SPECIFY E(M) OF POSTERIOR FOR SAKE OF
     CALCULATIONS IN SUBROUTINE EXFCD
     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
     ENDF
     IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT = ', MCHAT(1,LL)
     ELSE
       WRITE(8,*), 'ERROR: OVERALL PRIOR RANGE INCORRECTLY ',
       'SPECIFIED IN REGION WITHOUT DATA'
       CALL TRMNAT
     ENDF
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
C FORMAT/COMMENTS: 15SEP89
C VERSION: MATCHCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.3,
C V8.4, V8.5
C
C SUBROUTINE CONCAV (NUMREG, RANGEM)
C
C INPUTS: NUMREG, RANGEM
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT
C
C IMPLICIT NONE
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NUMREG
C REAL RANGEM(2, MAXREG), TESTM

C LIST OF VARIABLES
C IOUT OUTPUT DUMP CONTROLLER

7 - 196
Do 100 L = NUMREG, 2, -1
  IF (RANGEM(2,L-1), EQ. 0.0) THEN
    CALL TRMNAT
  ELSE
    TESTM = AMAX1(RANGEM(I,L), RANGEM(2,L))
    IF (TESTM .LT. RANGEM(I,L-1)) THEN
      CALL TRMNAT
    ELSE
      RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
    ENDIF
  ENDIF
100 CONTINUE

RETURN
END
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)
TO BE USED IN MEDIAN CALCULATION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND

C INITIALIZE ARRAY MEDM
DO 50 L = 1, MAXREG
   MEDM(L) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS FOR EACH REGION
DO 100 L = 1, NUMREG
   IF (RANGEM(2,L) .EQ. 0.0) THEN
      C RANGE IS A POINT
      MEDM(L) = RANGEM(1,L)
   ELSEIF (L .EQ. 1) THEN
      C WE ARE IN REGION ONE -- NOT AFFECTED BY OTHER REGIONS
      -- MEDIAN WILL JUST BE AVERAGE OF RANGEM VALUES
      MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
   ELSE
      C MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT
      LOWERM = AMAX1(RANGEM(1,L), MEDM(L-1))
      MEDM(L) = (LOWERM + RANGEM(2,L)) / 2.0
   ENDIF
   IF (IOUT .EQ. 10) THEN
      WRITE(8,*), 'L = ', L, ' NUMREG = ', NUMREG
      WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
      & ' RANGEM(2,L) = ', RANGEM(2,L),
      & ' LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
   ENDIF
100 CONTINUE

RETURN
END

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

C SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N
C CURVE PARAMETERS
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89

7 - 198
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 1OUT
INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG
REAL
BIGK(0:MAXREG), BIGK1, BZHAT, FACTR, KHAT, MEANZ,
& MEDM(MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
& NF(MAXDAT, MAXREG), RBND(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
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
RBND() 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 RAWSTR() 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
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

7 - 199
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)
    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
    TBIGK(L) = BIGK(L)
150 CONTINUE
    IF (ZROREG .EQ. 0) THEN
      FACTR = 1.0
      CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TBIGK, FACTR, NUMREG)
    ENDIF
C WRITE RESULTS TO FILE
    IF (NCOMPS .EQ. 1) THEN
      WRITE(7,900) NUMREG, BZhat, KHAT
      IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZhat, KHAT
      DO 200 L = ZROREG, NUMREG
        WRITE(7,910) L, MM(L), TBIGK(L), NBND(L), SBND(L)
        IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TBIGK(L), NBND(L), SBND(L)
200 CONTINUE
      ENDIF
C FORMAT STATEMENTS
900 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',/,'NUMBER OF REGIONS:',I4,5X,'E(BETAO) =',F8.4,5X,'E(k) =',F8.4,5X,'E(LIFE BOUND) =',7X)
910 FORMAT(5X,I1,5X,F9.5,5X,E12.5X,F9.5,5X,E9.3,9X,E11.5)
920 FORMAT(///)
930 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',/,'m',7X,'k',7X,'E(k)')
SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88  COMMENTS: 13JUL89

SUBROUTINE MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA,
& MO, SIGMA2, MCHAT, MU, SIG)

C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO, SIGMA2
C OUTPUTS: MCHAT, MU, SIG

C IMPLICIT NONE

INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG, NPPR(MAXREG)
REAL ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
& MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
& SUHAT2(MAXREG), SUMX2, SWHAT2(MAXREG), SX2(MAXREG)

C LIST OF VARIABLES
C ARG INTERMEDIATE CALCULATION VARIABLE
C DD() 1-D ARRAY CONTAINING SX(L)/SX2(L) FOR EACH REGION
C DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C SIG() CALCULATION
C IOUT OUTPUT DUMP CONTROLLER
C L CONTROLS DO LOOP FOR EACH REGION
C MAXREG MAXIMUM NUMBER OF REGION ALLOWED
C MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR
C EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) =
C - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT,
C MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C MEAN FOR EACH REGION
C MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C DISTRIBUTION MEAN FOR EACH REGION
C NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1)))-1 OVER ALL
C DATA SETS IN A REGION (Number of Points Per Region)
C NUMREG NUMBER OF REGIONS OF INTEREST
C SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
C STANDARD DEVIATION FOR EACH REGION
C SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C SUMX2 EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
C SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C (X = Ln S)

C INITIALIZE ARRAYS

RETURN
END
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
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
SIG(L) = SQRT (SUHAT2(L)) / ARG
ELSE
SIG(L) = SQRT (SIGMA2(L) / ARG)
ENDIF
IF (IOUT .EQ. 10) THEN
WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ', MCHAT(1,L)
WRITE(8,*) 'SUHAT2 = ', SUHAT2(L), ' MCHAT2 = ', MCHAT(2,L)
WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
& ' SUMX2 = ', SUMX2
WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
WRITE(8,*) 'SIGMA2 = ', SIGMA2(L), ' SIG = ', SIG(L)
ENDIF
100 CONTINUE
RETURN
END
C**************************************************************************************************
C SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND CO TO
C OBTAIN POSTERIOR RANGES ON M FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 10FEB88 FORMAT/COMMENTS: 12AUG91
C 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

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 C0 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) = SCHAT, 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

C INITIALIZE VARIABLES

DO 50 L = 1, MAXREG
   RANGEM(1,L) = 0.0
   RANGEM(2,L) = 0.0
50 CONTINUE

C PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

DO 100 L = 1, NUMREG

   IF (IOUT .EQ. 10) THEN
      WRITE(8,*) 'L = ', L, ', NUMREG = ', NUMREG
      WRITE(8,*) 'MPNT = ', MPNT(L), ', MCPNT = ', MCPNT(L)
   ENDIF

   IF (MPNT(L) .EQ. 1) THEN
      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 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,*) 'MZERO(2,L) = ', MZERO(2,L),
         &   'RANGEM(1,L) = ', RANGEM(1,L),
         &   'RANGEM(2,L) = ', RANGEM(2,L)
      ENDIF
   ELSEIF ((MPNT(L) .EQ. 1) .AND. (MCPNT(L) .EQ. 0)) THEN
      C 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)
   ELSE
      C THERE IS NO PRIOR INFORMATION ON M
      RANGEM(1,L) = 1.0
      RANGEM(2,L) = 0.0
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
         WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L)
      ENDIF
   ENDIF

100 CONTINUE
C

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 M TO 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 M AND MC'
CALL TRNAT
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 M AND MC'
CALL TRNAT
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 TRNAT
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

7 - 204
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 ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
PROGRAMMER: L. NEWLIN
DATE: CODE: 10FEB88 FORMAT/COMMENTS: 12AUG91

SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
& MZERO, MPNT, MO, SIGMA2)

INPUTS: RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
& MO, SIGMA2

OUTPUTS: RANGEM, MCHAT, MU, SIG, NUMREG

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
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 LL = 1, NMODAT
   NUMREG = NUMREG + 1
   IF (IOUT .EQ. 10) WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG,
      & ' 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),
            & ' RANGEM(2,LL) = ', RANGEM(2,LL),
            & ' MU(LL) = ', MU( 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)
      ELSE
         MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
      ENDIF
      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***************************************************************

C SUBROUTINE PAREST CONTROLS THE CALCULATIONS FOR THE PARAMETER
C ESTIMATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C PROGRAMMER: L. NEWLIN
C DATE: 13FEB89  FORMAT/COMMENTS: 15SEP89
C VERSION: MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
C MATGRM 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 PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG,
      & ZROREG, RAND, NBND, STR, BIGK, BZERO, MM,
      & SBND)
C
C INPUTS: VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
C NBND, STR
C OUTPUTS: BIGK, BZERO, MM, SBND
C SUBPROGRAMS: FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB
C
C***************************************************************
CALL FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
ENDIF

C TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)
CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)
CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C CALCULATE THE VALUES FOR k AND BETAO FROM THE SAMPLE MEAN
C AND VARIANCE
CALL KBETA (MEANZ, SZ2, K, BZERO)

C CALCULATE THE VALUE OF K FOR EACH REGION WHERE A = K ** M
CALL FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C CALCULATE STRESS TIE-POINTS
CALL FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)

C WRITE RESULTS TO FILE
C WRITE(7,900) NUMREG, BZERO
C DO 200 L = ZROREG, NUMREG
C WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
C 200 CONTINUE
C WRITE(7,920)
C FORMAT STATEMENTS
900 FORMAT(///,2X,'SELECTED VALUES OF S/N CURVE PARAMETERS',
& //,2X,'NUMBER OF REGIONS: ',I4,5X,'BETAO = ',F8.4,
& //,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',5X,
& 'STRESS BOUND')
910 FORMAT(5X,11,5X,F9.5,5X,E12.5,5X,E9.3,6X,E11.5)
920 FORMAT(///)

RETURN
END

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

C SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION BY
C SAMPLING OFF THE APPROPRIATE M RANG
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88 COMMENTS: 13JUL89
C VERSION: MATCHR VS, VS.11 : V8.2, V8.3 : V8.4 : V8.5C MATGRM
C SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)
C INPUTS: RAND, NUMREG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: RANDOM, TRNMAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES
IOUT
OUTPUT DUMP CONTROLLER
L
CONTROLS DO LOOP FOR EACH REGION
MAXREG
MAXIMUM NUMBER OF REGIONS ALLOWED
MM()
1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG
NUMBER OF REGIONS OF INTEREST
PICK()
1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RAND
RANDOM NUMBER SEED
RANGEM()
2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
X
UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED OFF THE RANGE ON M

C INITIALIZE MM()
DO 50 L = 0, MAXREG
   MM(MAXREG) = 0.0
50 CONTINUE
C BEGIN CALCULATIONS
DO 100 L = 1, NUMREG
   PICK(1) = 0.0
   PICK(2) = 0.0
   IF (RANGEM(2,L) .EQ. 0.0) THEN
      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
      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


PROGRAMMER: L. GRONDALSKI, L. NEWLIN
DATE: JDEC87

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
RANM CONSTANT FOR LCG
RANSUB INTERNAL CALCULATION
RANT INTERNAL CALCULATION
RANX INTERNAL CALCULATION

USING LCG RANDOM # GENERATOR

RANA = 671093.0
RANC = 790885.0
RANM = 33554432.0

10 RANX = RANA * RAND + RANC
RANDIV = RANX / RANM
RANT = INT(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(0,'(A,F10.0,E10.0)')'RANX =' RANX, 'RAND =', RAND, 'RANDIV =', RANDIV, 'RANT =', RANT, 'RANSUB =', RANSUB, 'RAND =', RAND, 'FRAC =', FRAC
RETURN
END

NOTES: IOUT=2 DUMPS TO SCREEN
C* SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C* SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C* PROGRAMMER: L. NEWLIN
C* DATE: CODE: 7JUN88 COMMENTS: 13FEB89
C* VERSION: MATCHF V8, V8.1, V8.2, V8.3, V8.4, V8.5
C
SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
C INPUTS: RAND, NUMREG, MU, SIG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: NORMGN, TRMNAT
C
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG),
& SIG(MAXREG), X
DOUBLE PRECISION RAND

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
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
    M IS SPECIFIED AS A POINT VALUE
    MM(L) = RANGEM(1,L)
  ELSEIF (L.EQ.10) THEN
    WRITE(*,*) 'RANGEM(I,L) =', RANGEM(I,L)
    MM(L) = RANGEM(I,L)
  ELSE
    CALL NORMGN (RAND, MU(L), SIG(L), X)
  END IF
100 CONTINUE

C
C
C
C
SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

IMPLICIT NONE
COMMON IOUT
DOUBLE PRECISION RAND
REAL FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2
PARAMETER (PI = 3.1415926536)
INTEGER IOUT

100 CONTINUE
RETURN
END

C*******************************************************************************************
C SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C PROGRAM: L. GRONDAULSKY, L. NEWMAN
C DATE: FEBRUARY 1977
C VERSION: MATGRNM V4, MATGRNM V4.4, MATGRNM V4.5
C The random variates are generated using the "Direct Method"
C Abramowitz, M., and Stegun, I. A., editors, Handbook of
C Mathematical Functions, National Bureau of Standards, Applied
C Mathematics Series 55, Issued June 1964, Ninth Printing, November
C 1970 with corrections, pg. 953.
C*******************************************************************************************
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
U2 = FRAC
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*),'U1 =', U1, 'U2 =', U2
Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)
X = SIGMA * Z1 + MU
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*),'Z1 =', Z1, 'Z2 =', Z2, 'X =', X
RETURN
END

SUBROUTINE TRNSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM
THE S/N DATA INTO THE VARIABLE Z = Ln(X)

SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

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
C
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)

LIST OF VARIABLES

I OUT CONTROLS DO LOOP FOR EACH DATA POINT
K OUT CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
L OUT CONTROLS DO LOOP FOR EACH REGION

7 - 213
IL1 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
NBDN() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
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
S/N DATA SET BROKEN INTO REGIONS
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 %)
Z() 1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
Z = F(STR, NF, NBDN)
C INITIALIZE VARIABLES
NP = 0
DO 50 I = 1, MAXDAT
ZZ(I) = 0.0
50 CONTINUE
C BEGIN CALCULATIONS
DO 100 L = 1, NUMREG
MML = MM(L)
IF (IOUT .EQ. 10) WRITE(8,*,'(L,,L, ', MM =', MM(L), ' MML =',
& MML, ' NPTS =', NPTS(L))
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(NBDN(LL-1))
& IF (IOUT .EQ. 10) WRITE(8,*,'(LL =', LL, ' NBDN(LL-1) =',
& NBDN(LL-1), ' MM(LL-1) =', MM(LL-1), ' MM(LL) =',
& MM(LL), ' ZZ =', ZZ(NP))
300 CONTINUE
200 CONTINUE
100 CONTINUE
RETURN
END

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

C SUBROUTINE SMNVAR CALCULATES THE Sample Mean and Variance OF
C Z = F(STR, NF, NBDN, 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,
V8.3, V8.4, V8.5
C SUBROUTINE SMNVAR (NP, ZZ, MEANZ, SZ2)
C INPUTS: NP, ZZ

7-214
C OUTPUTS: MEANZ, SZ2
C
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 DATA, Z = F(STR, NF, NBND, MM)

C INITIALIZE VARIABLES
MEANZ = 0.0
SZ2 = 0.0
C
C CALCULATE THE MEAN OF ZZ(), MEANZ
DO 100 I = 1, NP
   MEANZ = MEANZ + ZZ(I)
   IF (IOUT .EQ. I0) WRITE(8,*)'NP =', NP, ' I =', I,
   & ZZ =', ZZ(I), ' MEANZ =', MEANZ
100 CONTINUE
MEANZ = MEANZ / FLOAT(NP)
IF (IOUT .EQ. I0) WRITE(8,*)' MEANZ =', MEANZ
C
C CALCULATE THE VARIANCE OF ZZ(), SZ2
DO 200 I = 1, NP
   SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
   IF (IOUT .EQ. I0) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
200 CONTINUE
SZ2 = SZ2 / FLOAT(NP - 1)
IF (IOUT .EQ. I0) WRITE(8,*)' SZ2 =', SZ2
RETURN
END

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

C SUBROUTINE KBETA CALCULATES k AND BETAO FROM THE SAMPLE MEAN AND VARIANCE
C OF Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87   COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C V8.4, V8.5
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

LIST OF VARIABLES
BZERO  VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE SPECIFIC MATERIAL S/N DATA SET
IOUT   OUTPUT DUMP CONTROLLER
K     VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
PI    SELF EXPLANATORY CONSTANT
SZ    SZ2 ** 0.5
SZ2   SAMPLE VARIANCE OF THE TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

C PERFORM CALCULATIONS

SZ = SZ2 ** 0.5
BZERO = 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 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)
LIST OF VARIABLES

BIGK()  1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
FOR EACH REGION
BZERO  VALUE OF WEIBULL PARAMETER, BETA, CHARACTERIZING SPECIFIC
        MATERIAL DATA BASE
GAMMA  EULER'S CONSTANT
IOUT   OUTPUT DUMP CONTROLLER
K      VALUE OF K -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
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: 1, K = ', BIGK(1)
   IF (IOUT .EQ. I0) WRITE(8,*) 'BZERO = ', BZERO, ' K = ', K,
   & ' GAMMA = ', GAMMA, ' BIGK(1) = ', BIGK(1)

C CALCULATE K FOR REMAINING REGIONS

DO 100 L = 2, NUMREG
   BIGK(L) = BIGK(L-1) * NBND(L-1)
   & ** ((1.0 / MM(L)) - (1.0 / MM(L-1)))
   WRITE(7,*) 'REGION ', L, ', K = ', BIGK(L)
   IF (IOUT .EQ. I0) WRITE(8,*) 'L = ', L, ', NBND(L-1) = ',
   & ' MM(L-1) = ', MM(L-1), ', MM(L) = ', MM(L),
   & ' BIGK(L) = ', BIGK(L)
100 CONTINUE

RETURN
END
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG, ZROREG
REAL BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
     & SBND(0:MAXREG)

LIST OF VARIABLES
BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
FOR EACH REGION
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
       REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS OF INTEREST
SBND() 1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
       CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
       REGION CONTAINED IN NBND()
ZROREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
       BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO REGION

C INITIALIZATION 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 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: MATCHRC 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, 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.

7 - 218
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

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

SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
& FACTR, NUMREG)
C INPUTS: SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
& FACTR, NUMREG
C OUTPUTS: SZERO, BIGK, MM, NBND, TRSBND, FACTR
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
TRBIGK() 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
  TRBIGK(L) = BIGK(L) * SCLK
  TRSBND(L) = TRSBND(L) * SCLK
  CONTINUE
ELSE
  TRBIGK(0) = SZERO/FACTR
  MM(0) = MM(1) * ((ALOG (BIGK(1)) - ALOG (TRSBND(0)))
          + ALOG (FACTR)) / (ALOG (SZERO) - ALOG (TRSBND(0)))
ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'SZERO = ', SZERO, ' BIGKo-', TRBIGK(0)
  WRITE(8,*) 'FACTOR = ', FACTR, ' BIGKI - 't TRBIGK(1)
  WRITE(8,*) 'MMI - ', MM(1), ' MMO- ', MM(0)
ENDIF

RETURN
END

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: MATCHCR 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, LH, SBND, ZROREG, NUMREG, SZERO
C OUTPUTS: GTLIFE

7 - 220
IMPLICIT NONE
INTEGER IOUT, L, MAXREG, NUMREG, ZROREG
PARAMETER (MAXREG = 3)
COMMON IOUT
REAL GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
& MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

LIST OF VARIABLES

GETLIF  VALUE TO BE ASSIGNED TO GTLIFE -- CYCLES TO FAILURE FOR
   THE REQUIRED STRESS LEVEL
IOUT  OUTPUT DUMP CONTROLLER
KRATIO  RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L  CONTROLS DO LOOP FOR EACH REGION
LNA()  1-D ARRAY CONTAINING VALUES OF LN(A) = M LN K FOR EACH REGION
LNZ  NORMAL(0, PVAR) GENERATED RANDOM VARIATE
LPHIM()  1-D ARRAY CONTAINING VALUES OF M LN PHI FOR EACH REGION WHERE
PHI IS A WEIBULL(BETA0, ETA0) GENERATED RANDOM VARIATE
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
MM()  1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG  NUMBER OF REGIONS OF INTEREST
S  VALUE OF STRESS (PSI) FOR WHICH A VALUE OF LIFE (CYCLES TO
FAILURE) IS REQUIRED
SBND()  1-D ARRAY CONTAINING THE STRESS VALUES (PSI, R = -1.0)
   CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
   CONTAINED IN NBND()
SZERO  STRESS TENSILE POINT, So
TEMP  TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER
   FLOWS
ZROREG  Zero Region -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
   BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO REGION

GETLIF = 0.0

C CALCULATE CYCLES TO FAILURE
IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
   GETLIF = 1.0
ELSE
   DO 100 L = ZROREG, NUMREG
      IF (S .GT. SBND(L)) THEN
         TEMP = LNA(L) + LPHIM(L) + MM(L) * + ALGN (KRATIO) + LNZ)
         IF (TEMP .GT. 86.0) THEN
            TEMP = 86.0
         ENDIF
         GETLIF = EXP (TEMP)
      ENDIF
   ENDIF
   GOTO 150
900 CONTINUE
ENDIF
GOTO 150
GTLIFE = GETLIF
RETURN
END

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

C SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C M FOR EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: 10FEB88

7 - 221
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

5 INC = NUM
10 IF (INC .GT. 1) THEN
20 INC = INC / 2

INORDR = .TRUE.

DO 300 I = 1, (NUM - INC)

IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN

TEMP = ALLM(I,L)

ALLM(I,L) = ALLM(I + INC, L)

ALLM(I + INC, L) = TEMP

INORDR = .FALSE.

ENDIF
300 CONTINUE

IF (.NOT. INORDR) GOTO 20

GOTO 10

ENDIF
400 CONTINUE

RETURN

END

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

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

C FUNCTION THWELD CONTROLS THE CALLS REQUIRED TO CALCULATE A LIFE FOR A
C PLAIN WELD UNDER A THERMAL LOAD
PROGRAMMER: L. NEWLIN
DATE: 11JUL90
VERSION: THDUCT V4, V4.1

FUNCTION THWELD (ALPHA, ANGLE, DLTAT, E, EM, FTU, FTY, DI, K, &
  KRATIO, LAMW, FK, RT, LNA, LN2, LOCAT, LPHIH, &
  M, MM, NSTAT, NEUB, NLOAD, NRAN, NU, NUMREG, &
  NUMSEG, P, PC, PCO, PERIOD, PSTAT, SBND, SE, &
  STRHIS, SZERO, T, THIC, TRUNC, TSTAT, V, &
  VSTAT, WOFF, ZROREG)

INPUTS: ALPHA, ANGLE, DLTAT, E, EM, FTU, FTY, DI, K, LAMW, FK, RT, LNA, &
  LOCAT, LPHIH, M, MM, NSTAT, NEUB, NLOAD, NRAN, NU, NUMREG, NUMSEG, P, PC, PCO, PERIOD, PSTAT, SBND, SE, STRHIS, SZERO, T, THIC, TRUNC, TSTAT, V, VSTAT, WOFF

OUTPUTS: THWELD

SUBPROGRAMS: M4L1, M4L2, NARBN2, TRMNAT

C IMPLICIT NONE
C INTEGER MAXLD, MAXM, MAXREG, MAXSEG
C PARAMETER (MAXLD = 16, MAXM = 24000, MAXREG = 3, MAXSEG = 10)
C COMMON IOUT
C INTEGER IOUT, LOCAT, NLOAD, NUMSEG, NRAN, NUMREG, ZROREG
C REAL
&
   ALPHA, ANGLE, DI, DLTAT, E(MAXSEG), EM, FATLIF, FK(10), &
   FTY, FTU, K(2, 2), KRATIO, LAMW, LNA(0:MAXREG), LN2, &
   LPHIH(0:MAXREG), M(2, MAXLD), MM(0:MAXREG), NSTAT(2), &
   NEUB, NU, P(HAMK), PC, PCO, PERIOD, PSTAT, RT(10), &
   SBND(0:MAXREG), SE(MAXSEG), STRAMP(4, MAXLD), &
   STATIC(4), SZERO, T(MAXLD), THIC, &
   THWELD, TRUNC, TSTAT, WOFF, V(2, MAXLD), VSTAT(2)

LIST OF VARIABLES

ALPHA COEFFICIENT OF THERMAL EXPANSION
ANGLE ANGLE THETA IN RADIANS
DI INTERIOR DIAMETER
DLTAT TEMPERATURE DIFFERENCE BETWEEN THE INNER AND OUTER SURFACES
E() 1-D ARRAY CONTAINING THE STRAIN VALUES
EM YOUNG'S MODULUS BEFORE YIELD
FATLIF 1-D ARRAY CONTAINING VALUES OF FK USED TO FIND STRESS
FK() 1-D ARRAY CONTAINING VALUES OF FK USED TO FIND STRESS
FTY YIELD STRENGTH
FTU ULTIMATE STRENGTH
IOUT OUTPUT DUMP CONTROLLER
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
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
LAMW ACCURACY FACTOR OF FK - R/T CURVE
LNA() 1-D ARRAY CONTAINING VALUES OF LN(A) = Ln(K**M) FOR EACH REGION
LN2 NORMAL (0,PVAR) GENERATED FROM RANDOM VARIATE
LOCAT LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE
2 IS THE INTERIOR SURFACE OF THE DUCT

LPHIM() 1-D ARRAY CONTAINING VALUES OF Ln(PHI)m FOR EACH REGION
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
MAXM MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSEG MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN)
MM() 1-D ARRAY CONTAINING VALUES FOR m FOR EACH REGION
MSTAT() 1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE M2 LOAD; M(2) IS THE M3 LOAD
NEUB NEUBER'S RULE MODEL ACCURACY FACTOR
NLLOAD NUMBER OF TIME-VARYING LOADS
NRAN NUMBER OF POINTS IN STRESS-TIME HISTORY
NU POISSON'S RATIO
NUMREG NUMBER OF REGIONS OF INTEREST
NUMSEG NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
P() 1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
PC LIMIT PRESSURE ON INSIDE OF THE TUBE
PCO LIMIT PRESSURE ON OUTSIDE OF THE TUBE
PERIOD LENGTH OF TIME IN SECONDS FOR RANDOM STRESS-TIME HISTORY
PSAT STATIC AXIAL LOAD
RT() 1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS CONCENTRATION DUE TO WELD ECCENTRICITY
SND() 1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CORRECTED FOR PHI, KRATIO AND LNZ
SE() 1-D ARRAY CONTAINING THE STRESS AND STRAIN PRODUCTS
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
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
SZERO STRESS TENSILE POINT (PSI)
T() 1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
THIC WALL THICKNESS AT DUCT OUTER RADIUS
THWLD FUNCTION WHICH CALCULATES THE LIFR FOR THE PLAIN WELD UNDER THERMAL LOADS
TRUNC VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
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
ZROREG ZERO REGION--VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE --0 - ZERO REGION EXISTS, 1-NO ZERO REGION

IF (LOCAT .EQ. 1) THEN
CALL M4L1 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMM, M, MSTAT, 
& NLLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP, T, 
& THIC, TSTAT, V, VSTAT, WOFF, FK, RT)
CALL NARBN2 (E, EM, FATLIF, FTY, 1.0, KRATIO, LNA, LNZ, 
& LPHIM, MM, NRAN, NEUB, NLOAD, NUMREG, NUMSEG, 
& SND, PERIOD, SE, STATIC, STRAMP, STRHIS, SZERO, 
& TRUNC, ZROREG)
ELSE IF (LOCAT .EQ. 2) THEN
CALL M4L2 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMM, M, MSTAT, 
& NLLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP, T, 
& THIC, TSTAT, V, VSTAT, WOFF, FK, RT)
ENDIF

7 - 224
CALL NARBN2 (E, EM, FATLIF, FTU, FTY, 1.0, KRATIO, LNA, LHZ,
& \LPHIM, MM, NRAM, NEUB, NLOAD, NUMREG, NUMSEG,
& \SBND, PERIOD, SE, STATIC, STRAMP, STRHIS, SZERO,
& \TRUNC, ZROREG)

ELSE

WRITE(8,*) 'ERROR: INVALID LOCATION SPECIFICATION'
CALL TRNWAT

ENDIF

THWELD = FATLIF
RETURN
END

***************************************************************************
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)

INPUTS: ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT, NLOA D, NU, P,
PC, PCO, PSTAT, STATIC, T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT

OUTPUTS: STATIC, STRAMP

C IMPLICIT NONE

COMMON IOUT

INTEGER I, IOUT, J, MAXLD, NLOAD

REAL PI

PARAMETER (MAXLD = 16, PI = 3.1415926536)

REAL ALPHA, ANGLE, AREA, DI, DLTAT, EM, FK(10), GEOM, IFK,
& K(2, 2), KKOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU,
& P(MAXLD), PC, PCO, PSTAT, RDIFF, RI, RI2, RO, RO2,
& ROT, RT(10), SIGIA(MAXLD), SIGIB(MAXLD), SKT1, SKT2,
& STATIC(4), STRHAT, STR1A, STR2A, STR1B, STR2B, STRIC,
& STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD),
& VSTAT(2), WOFF

LIST OF VARIABLES

ALPHA COEFFICIENT OF THERMAL EXPANSION
ANGLE ANGLE THETA IN RADIANS
AREA CROSS SECTION AREA OF DUCT WALL
DI INTERIOR DIAMETER
DLTAT TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
EM YOUNG'S MODULUS PRIOR TO YIELD
FK() 1-D ARRAY CONTAINING VALUES OF FK USED TO FIND STRESS
    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
CONTROLLER

CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING
INTERPOLATION

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

STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD

2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS

MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED

MOMENT OF INERTIA

1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE M
2 LOAD; M(2) IS THE M3 LOAD

NUMBER OF TIME-VARYING LOADS

POISSON'S RATIO

1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS

LIMIT PRESSURE ON INSIDE OF THE VESSEL

LIMIT PRESSURE ON OUTSIDE OF THE VESSEL

SELF EXPLANATORY CONSTANT

STATIC AXIAL LOAD

EQUAL TO RO2 - R12

INTERIOR RADIUS

INNER RADIUS SQUARED

OUTER RADIUS

OUTER RADIUS SQUARED

EQUAL TO R / T (R Over T)

1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
CONCENTRATION DUE TO WELD ECCENTRICITY

1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO FORCE
FOR THE TIME-VARYING LOADS

1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO BENDING
FOR THE TIME-VARYING LOADS

STRESS CONCENTRATION FACTOR FOR AXIAL STRESS

STRESS CONCENTRATION FACTOR FOR HOOP STRESS

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

THE STATIC AXIAL STRESS DUE TO THERMAL GRADIENT

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

WALL THICKNESS AT DUCT OUTER RADIUS

STATIC TORQUE LOAD

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

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

WELD OFFSET

CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO
ECCENTRICITY OF THE WELD

RI = DI / 2.0
ROT = (DI + THIC) / (2.0 * THIC)

DO 50 J = 2, 10
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

7 - 226
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 EXTERIOR SURFACE

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

TEMPERATURE STRESS

STHMA = ((EM * ALPHA * DLTAT) / (2.00 * (1.00 - NU) & * LOG (RO / RI))) * GEOM

AXIAL STRESS CALCULATIONS

STRIA = PSTAT / AREA
STRIB = (MSTAT(1) * COS (ANGLE) + MSTAT(2) * SIN (ANGLE)) * RO & / MI
STRIC = (PC - PCO) * RI2 / RDIFF
STATIC(1) = (STRIA + STRIB + STRIC) * SKT1 * KOFF + STHMA

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

SHEAR STRESS

STATIC(4) = TSTAT * RO / (2.0 * MI) - (2.0 / AREA & * (VSTAT(1) * COS (ANGLE) + VSTAT(2) * SIN (ANGLE)))

IF (IOUT .EQ. 25) THEN
  WRITE(8,*) 'RO2 = ', RO2, ' RI2 = ', RI2
  WRITE(8,*) 'RDIFF = ', RDIFF, ' GEOM = ', GEOM
  WRITE(8,*) 'STATIC STRESS VALUES'
  WRITE(8,*) 'AXIAL STRESSES'
  WRITE(8,*) 'STRIA = ', STRIA, ' STRIB = ', STRIB
  WRITE(8,*) 'STRIC = ', STRIC, ' STHMA = ', STHMA
  WRITE(8,*) 'STATIC(1) = ', STATIC(1)
ENDIF
WRITE(8,*), 'HOOP STRESSES'
WRITE(8,*), 'STRA2 = ', STRA2, ' STRA = ', STRA,
& ' STHMA = ', STHMA
WRITE(8,*), 'STATIC(2) = ', STATIC(2)
WRITE(8,*), 'RADIAL STRESS', ' STATIC(3) = ', STATIC(3)
WRITE(8,*), 'SHEAR STRESS', ' STATIC(4) = ', STATIC(4)
WRITE(8,*)
ENDIF

DO 100 I = 1, NLOAD
C AXIAL STRESS CALCULATIONS

SIG1A(I) = P(I) / AREA
SIG1B(I) = (M(1,I) * COS (ANGLE) + M(2,I) * SIN (ANGLE))
& * RO / MI
STRAMP(1,I) = (SIG1A(I) + SIG1B(I)) * SKT1 * KOFF
C
C HOOP (2) AND RADIAL (3) STRESSES ARE ZERO
C 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,*), 'SIG1A = ', SIG1A(I), ' SIG1B = ', SIG1B(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)
WRITE(8,*)
ENDIF

100 CONTINUE

IF (IOUT.EQ.25) THEN
WRITE(8,*), 'I' AXIAL HOOP RADIAL SHEAR'
WRITE(8,*), 'STATIC(1), STATIC(2), STATIC(3), STATIC(4)
DO 300 I = 1, NLOAD
WRITE(8,*), I, STRAMP(1,I), STRAMP(2,I), STRAMP(3,I),
& STRAMP(4,I)
300 CONTINUE
ENDIF
RETURN
END
INPUTS: ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT, NLOAD, NU, P, PC, PCO, PSTAT, T, THIC, TSTAT V, VSTAT, WOFF, FK, RT

OUTPUTS: STATIC, STRAMP

IMPLICIT NONE

COMMON IOUT
INTEGER I, IOUT, J, MAXLD, NLOAD
REAL PI
PARAMETER (MAXLD = 16, PI = 3.1415926536)
REAL ALPHA, ANGLE, AREA, DLTAT, EM, FK(10), GEOM, IFK, DI, K(2, 2), KOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU, P(2, MAXLD), PC, PCO, PSTAT, RDIFF, RI, RI2, RO, RO2, RT(10), SIGIA(MAXLD), SIGIB(MAXLD), SKTI, STHMA, STRIB, STR2A, STR2B, STRIA, STR2B, C, STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD), VSTAT(2), WOFF

LIST OF VARIABLES

ALPHA COEFFICIENT OF THERMAL EXPANSION
ANGLE ANGLE THETA IN RADIANS
AREA CROSS SECTION AREA OF DUCT WALL
DI INTERIOR DIAMETER
DLTAT TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
EM YOUNG’S MODULUS PRIOR TO YIELD
FK() 1-D ARRAY CONTAINING VALUES OF FK USED TO FIND STRESS CONCENTRATION DUE TO WELD ECCENTRICITY
GEOM INTERMEDIATE THERMAL STRESS CALCULATION VARIABLE
I CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND AERODYNAMIC LOADS
IFK INTERPOLATED VALUE OF FK CORRESPONDING TO THE VALUE OF R/T
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING INTERPOLATION
K() FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL DIRECTION; K(2,2) IS FOR DUCT INTERIOR FOR HOOP DIRECTION
KOFF STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
LAMW ACCURACY FACTOR OF FK - R/T CURVE
M() 2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*) ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
MAXLD MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
MI MOMENT OF INERTIA
MSTAT() 1-D ARRAY CONTAINING THE STATIC TIME-VARYING LOADS -- M(1) 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
PC LIMIT PRESSURE ON INSIDE OF THE VESSEL
PCO LIMIT PRESSURE ON OUTSIDE OF THE VESSEL
PI SELF EXPLANATORY CONSTANT
PSTAT STATIC AXIAL LOAD
RDIFF EQUAL TO RO2 - RI2
RI INTERIOR RADIUS
RI2 INNER RADIUS squared
RO OUTER RADIUS
RO2 OUTER RADIUS squared
R/T 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
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
C

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
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

C

CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO
      ECCENTRICITY OF THE WELD
PR = 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

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 RADI APPROPRIATE TO LOCATION
      THIS IS THE INTERIOR SURFACE
SKT1 = K(1,2)
SKT2 = K(2,2)
RO = RI + THIC

IF (IOUT .EQ. 25) THEN
      WRITE(8,*) 'AREA = ', AREA, ' MI = ', MI
      WRITE(8,*) 'K(1,2) = ', K(1,2), ' SKT1 = ', SKT1
      WRITE(8,*) 'K(2,2) = ', K(2,2), ' SKT2 = ', SKT2
      WRITE(8,*) 'THIC = ', THIC, ' RO = ', RO
      WRITE(8,*) 'ALPHA = ', ALPHA, ' NU = ', NU
      WRITE(8,*) 'DLTAT = ', DLTAT, ' EM = ', EM
      WRITE(8,*)
ENDIF

RI2 = RI ** 2

7 - 230
\[ RO_2 = RO \cdot \sqrt{2} \]
\[ RD\text{DIFF} = RO_2 - RI_2 \]
\[ \text{GEOM} = 1.00 - 2.00 \times \log(RO / RI) \times RO_2 / RD\text{DIFF} \]

\[ \text{TEMPERATURE STRESS} \]
\[ \text{ST\text{HMA}} = \left( EM \times \text{ALPHA} \times D\text{LTAT} / (2.00 \times (1.00 - NU) \times \log(RO / RI)) \right) \times \text{GEOM} \]

\[ \text{AXIAL STRESS CALCULATIONS} \]
\[ \text{ST\text{RIA}} = \text{PSTAT} / \text{AREA} \]
\[ \text{ST\text{RIB}} = \left( \text{MSTAT}(1) \times \cos(\text{ANGLE}) + \text{MSTAT}(2) \times \sin(\text{ANGLE}) \right) \times \text{RI} / \text{MI} \]
\[ \text{ST\text{RIC}} = \left( \text{PC} - \text{PCO} \right) \times RI_2 / RD\text{DIFF} \]
\[ \text{STATIC}(1) = \left( \text{ST\text{RIA}} + \text{ST\text{RIB}} + \text{ST\text{RIC}} \right) \times \text{SKT1} \times \text{KOFF} + \text{ST\text{HMA}} \]

\[ \text{HOOP (2) AND RADIAL (3) STRESS CALCULATIONS} \]
\[ \text{ST\text{R2A}} = \text{PC} \times (RI_2 + RO_2) / RD\text{DIFF} \]
\[ \text{ST\text{R2B}} = -2.0 \times \text{PCO} \times RO_2 / RD\text{DIFF} \]
\[ \text{STATIC}(2) = \left( \text{ST\text{R2A}} + \text{ST\text{R2B}} \right) \times \text{SKT2} + \text{ST\text{HMA}} \]
\[ \text{STATIC}(3) = -\text{PC} \]

\[ \text{SHEAR STRESS} \]
\[ \text{STATIC}(4) = \text{TSTAT} \times \text{RI} / \left( 2.0 \times \text{MI} \right) \times \left( 2.0 / \text{AREA} \times \left( \text{VSTAT}(1) \times \cos(\text{ANGLE}) + \text{VSTAT}(2) \times \sin(\text{ANGLE}) \right) \right) \]

IF (IOUT.EQ.25) THEN  
  WRITE(8,*), 'RO_2 = ', RO_2, 'RI_2 = ', RI_2 
  WRITE(8,*), 'RD\text{DIFF} = ', RD\text{DIFF}, 'GEOM = ', GEOM 
  WRITE(8,*), 'STATIC STRESS VALUES' 
  WRITE(8,*), 'AXIAL STRESSES' 
  WRITE(8,*), 'ST\text{RIA} = ', ST\text{RIA}, 'ST\text{RIB} = ', ST\text{RIB} 
  WRITE(8,*), 'ST\text{RIC} = ', ST\text{RIC} 
  WRITE(8,*), 'ST\text{HMA} = ', ST\text{HMA} 
  WRITE(8,*), 'HOOP STRESSES' 
  WRITE(8,*), 'ST\text{R2A} = ', ST\text{R2A}, 'ST\text{R2B} = ', ST\text{R2B} 
  WRITE(8,*), 'ST\text{HMA} = ', ST\text{HMA} 
  WRITE(8,*), 'RADIAL STRESS', 'STATIC(3) = ', STATIC(3) 
  WRITE(8,*), 'SHEAR STRESS', 'STATIC(4) = ', STATIC(4) 
ENDIF

DO 100 I = 1, NLOAD

\[ \text{AXIAL STRESS CALCULATIONS} \]
\[ \text{SIGIA}(I) = P(I) / \text{AREA} \]
\[ \text{SIGIB}(I) = \left( \text{M}(I,1) \times \cos(\text{ANGLE}) + \text{M}(2,I) \times \sin(\text{ANGLE}) \right) \times \text{RI} / \text{MI} \]
\[ \text{STR\text{AMP}}(1,I) = \left( \text{SIGIA}(I) + \text{SIGIB}(I) \right) \times \text{SKT1} \times \text{KOFF} \]

\[ \text{HOOP (2) AND RADIAL (3) STRESSES ARE ZERO} \]

\[ \text{BECAUSE PRESSURES ARE CONSTANT} \]
\[ \text{STR\text{AMP}}(2,I) = 0.0 \]
\[ \text{STR\text{AMP}}(3,I) = 0.0 \]

\[ \text{SHEAR STRESS} \]
\[ \text{STR\text{AMP}}(4,I) = \left( T(I) \times \text{RI} / \left( 2.0 \times \text{MI} \right) \right) \times \left( 2.0 / \text{AREA} \times \left( \text{V}(I,1) \times \cos(\text{ANGLE}) + \text{V}(2,I) \times \sin(\text{ANGLE}) \right) \right) \]

IF (IOUT.EQ.25) THEN
WRITE(8,*) 'STRESS VALUES FOR I = ', I
WRITE(8,*) 'AXIAL STRESSES'
WRITE(8,*) 'SIGMA = ', SIGMA(I), ' SIGMA(I)
WRITE(8,*) 'SIGMA(1) = ', STRAMP(I,1)
WRITE(8,*) 'SIGMA(2) = ', STRAMP(I,2)
WRITE(8,*) 'SIGMA(3) = ', STRAMP(I,3)
WRITE(8,*) 'SHEAR STRESS', STRAMP(I,4)
WRITE(8,*) 'RADIAL STRESS', STRAMP(I,5)
WRITE(8,*) 'HOOP STRESSES', STRAMP(I,6)
WRITE(8,*) 'AXIAL STRESSES'
WRITE(8,*) 'I OUT', IOUT
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(I,1), STRAMP(I,2), STRAMP(I,3), STRAMP(I,4)
300 CONTINUE
ENDIF
RETURN
END

***************************************************************************
C SUBROUTINE NARBN2 CALCULATES THE FATIGUE LIFE WHEN A RANDOMLY DISTRIBUTED
C LOAD IS PRESENT USING A SIMULATED NARROW BAND STRESS-TIME HISTORY
C PROGRAMMER: L. NEWLIN
C DATE: 11JUL90
C VERSION: THDUCT V4, V4.1
***************************************************************************

SUBROUTINE NARBN2 (E, EM, FATLIF, FTU, FTY, KT, KRATIO, LNA,
 & LNZ, LPHIM, MM, M, NEUB, NLOAD, NUMREG,
 & NUMSEG, SBND, PERIOD, SE, STATIC, STRAMP,
 & STRHIS, SZERO, TRUNC, ZROREG)

C INPUTS: E, EM, FTU, FTY, KT, KRATIO, LNA, LNZ, LPHIM, MM, M, NEUB,
 NLOAD, NUMREG, NUMSEG, SBND, PERIOD, SE, STATIC, STRAMP,
 STRHIS, SZERO, TRUNC, ZROREG

C OUTPUTS: FATLIF
C SUBPROGRAMS: RAINF2
C IMPLICIT NONE
INTEGER MAXLD, MAXM, MAXREG, MAXSEG
PARAMETER (MAXLD = 16, MAXM = 24000, MAXREG = 3, MAXSEG = 10)
COMMON IOUT
INTEGER I, IOUT, J, M, NLOAD, NUMREG, NUMSEG, ZROREG
REAL EM, E(MAXSEG), FATLIF, FTU, FTY, KRATIO, KT,
 & LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MM(0:MAXREG),
 & NEUB, PERIOD, RAINF2, S(4, MAXM), SBND(0:MAXREG),
 & SE(MAXSEG), SEFF(MAXM), STATIC(4), STRAMP(4), MAXLD,
 & STRHIS(MAXLD, MAXM), SZERO, TRUNC

LIST OF VARIABLES
E() 1-D ARRAY CONTAINING THE STRAIN VALUES
EM YOUNG'S MODULUS BEFORE YIELD
FATLIF VALUE OF FATIGUE LIFE CALCULATED
FTY YIELD STRENGTH
FTU ULTIMATE STRENGTH
CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND 
AERODYNAMIC LOADS

IOUT OUTPUT DUMP CONTROLLER

KRASTO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS

LNA() 1-D ARRAY CONTAINING VALUES OF Ln(A) = Ln(K**m) FOR EACH REGION

LPHI() 1-D ARRAY CONTAINING VALUES OF Ln(PHI)m FOR EACH REGION

M NUMBER OF POINTS IN STRESS-TIME HISTORY

MAXLD MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED

MAXM MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY

MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED

MM() 1-D ARRAY CONTAINING VALUES FOR m FOR EACH REGION

NEUB NEUBER'S RULE MODEL ACCURACY FACTOR

NLOAD NUMBER OF TIME-VARYING LOADS

NUMREG NUMBER OF REGIONS OF INTEREST

PERIOD TIME IN SECONDS FOR ONE PERIOD OF STRESS-TIME HISTORY

RAINP2 FUNCTION WHICH CALCULATES THE TIME TO FAILURE FOR A GIVEN 
UNI-AXIAL STRESS-TIME HISTORY

S() 2-D ARRAY CONTAINING THE TOTAL COMPONENT STRESS-TIME HISTORIES

SBND() 1-D ARRAY CONTAINING STRESS VALUES (PSI, R .. -I.0) 
CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH 
REGION CORRECTED FOR PHI, KRASTO AND LNZA

SE() 1-D ARRAY CONTAINING THE STRESS-STRAIN PRODUCT

SEFF() 1-D ARRAY CONTAINING THE EFFECTIVE (OR UNI-AXIAL) STRESS-TIME 
HISTORY RESULTING FROM THE COMBINATION OF STATIC, RANDOM, AND 
SINUSOIDAL LOADS FOR ALL FOUR COMPONENTS

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

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

SZERO STRESS TENSILE POINT (PSI)

TRUNC VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY 
ZREG ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP 
BEGINNING VALUES -- 0 - ZERO REGION EXISTS, 1 - NO ZERO 
REGION

DO 50 J = 1, M

S(1,J) = STATIC(1)
S(2,J) = STATIC(2)
S(3,J) = STATIC(3)
S(4,J) = STATIC(4)

50 CONTINUE

DO 100 I = 1, NLOAD

DO 150 J = 1, M

S(1,J) = S(1,J) + STRHIS(I,J) * STRAMP(1,I)
S(4,J) = S(4,J) + STRHIS(I,J) * STRAMP(4,I)

150 CONTINUE

100 CONTINUE

DO 400 J = 1, M

SEFF(J) = (S(1,J) / ABS(S(1,J))) * SQRT((
& (S(1,J) - S(2,J))**2 + (S(1,J) - S(3,J))**2 +
& (S(2,J) - S(3,J))**2 + (6.0 * S(4,J))**2) / 2.0)

400 CONTINUE

IF (IOUT.EQ. 25) THEN
DO 125 J = 1, M
  WRITE(8,*) J, 'S: ', S(1, J), S(2, J), S(3, J), S(4, J)
  WRITE(8,*) 'SEFF = ', SEFF(J)
125  CONTINUE
END

FATLIF = RAINF2 (E, EM, FTU, FTY, KT, KRATIO, LNA, LNZ, &
  LPHIM, M, MM, NEUB, PERIOD, SBND, NUMREG, &
  NUMSEG, SE, SEFF, SZERO, TRUNC, ZROREG)

IF (IOUT .EQ. 25) & WRITE(8,*)
  OD = ', PERIOD, ' FATLIF = ', FATLIF
RETURN
END

C******************************************************************
FUNCTION RAINF2 CALCULATES THE TIME (in seconds) TO FAILURE FOR
THE GIVEN UNI-AXIAL (OR EFFECTIVE) STRESS-TIME HISTORY

PROGRAMMER: L. NEWLIN
DATE: 20SEP89
VERSION: 4.2+ (CONSISTENT WITH THDUCT V3.8, V3.9, V4, & V4.1,
MATCHR V8.4, V8.5)
Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918
is acknowledged.
C******************************************************************

FUNCTION RAINF2 (E, EM, FTU, FTY, KT, KRATIO, LNA, LNZ, LPHIM, &
  M, MM, NEUB, PERIOD, SBND, NUMREG, NUMSEG, SE, &
  SEFF, SZERO, TRUNC, ZROREG)

C INPUTS: E, EM, FTU, FTY, KT, KRATIO, LNA, LNZ, LPHIM, M, MM, NEUB, PERIOD, &
  SBND, NUMREG, NUMSEG, SE, SEFF, SZERO, TRUNC, ZROREG

C OUTPUTS: RAINF2

C IMPLICIT NONE

COMMON IOUT
COMMON / COUNT / TOHIGH

INTEGER MAXREG, MAXM, MAXSEG
PARAMETER (MAXREG = 3, MAXM = 24000, MAXSEG = 10)

INTEGER BIG1, I, INDEX(MAXM), IOUT, J, JMAX, K, M, N, NEWTOT, &
  NUMREG, NUMSEG, OVER, TOHIGH, ZROREG

REAL ARGH, CHKFT, E(MAXSEG), EE(MAXM), EM, FTU, FTY, &
  GYLIFE, INVLF(MAXM), KRATIO, KT, LIFE(MAXM), &
  LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MM(0:MAXREG), &
  NEUB, NEUBER, PERIOD, RAINF2, S(MAXM), SALT(MAXM), &
  SBND(0:MAXREG), SE(MAXSEG), SEFF(MAXM), SEFFM(2, MAXM), &
  SFINAX, SM, SMAXF(MAXM), SP(MAXM), STR(MAXM), SUMDAM, &
  SZERO, TEST1(MAXM), TEST2(MAXM), TRUNC

LIST OF VARIABLES
RAINF2 CYCLES TO FAILURE FOR THE GIVEN STRESS LEVELS

input variables:
E()  1-D ARRAY CONTAINING THE STRAIN VALUES
EM() YOUNG'S MODULUS BEFORE YIELD
FTU ULTIMATE TENSILE STRENGTH (PSI)
FTY YIELD TENSILE STRENGTH (PSI)
IOUT  OUTPUT DUMP CONTROLLER
KRATIO  RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
KT  FATIGUE CONCENTRATION FACTOR
LNA()  1-D ARRAY CONTAINING VALUES OF Ln(A) = Ln(K**N) FOR EACH REGION
LNZ  NORMAL (0,PVAR) GENERATED FROM RANDOM VARIATE
LPHIM()  1-D ARRAY CONTAINING VALUES OF Ln(PHI)*M FOR EACH REGION WHERE PHI IS A WEIBULL(ETAO, ETAO) GENERATED RANDOM VARIATE
M  TOTAL NUMBER OF STRESS DATA POINTS PER PERIOD
MM()  1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
MAXM  MAXIMUM NUMBER OF POINTS IN STRESS TIME HISTORY
MAXREG  MAXIMUM NUMBER OF REGIONS ALLOWED
NUMREG  NUMBER OF REGIONS OF INTEREST
NUMSEG  NUMBER OF SEGMENTS OF INTEREST
PERIOD  TIME IN SECONDS FOR ONE PERIOD
SBND  TIME BETWEEN SECTIONS
SEF(N)  1-D ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
SALTF(I)  EFFECTIVE STRESSES BEFORE FILTERING/RAINFLOW
SEFFM(2,N)  EFFECTIVE STRESSES AFTER RESEQUENCING/FILTERING/RAINFLOW
SEFFM(2,1)  = sigma max,eff,i
SEFFM(2,i)  = sigma min,eff,i
SEFMAX  LARGEST EFFECTIVE STRESS
SM  SM = EQUIVALENT MEAN STRESS
SMEANF(N)  SMEANF(I) = sigma mean,eff,i
SP(M+1)  RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
STR(N)  STR(I) = EQUIVALENT (COMBINED) STRESS,I
SUNDAM  SUM OF ALL THE DAMAGE FRACTIONS
TEST1{}  1-D ARRAY USED IN FILTERING THE STRESSES
TEST2{}  1-D ARRAY USED IN FILTERING THE STRESSES
TOHIGH  COUNTER FOR AMOUNT OF STRESSES IN EXCESS OF FTU

intermediate variables:

ARGM  INTERMEDIATE CALCULATION VARIABLE EQUAL TO KT/(1 - SM/FTU)
BIG1  VALUE OF i FOR SEFMAX
CHKFT  RAINFL VALUE OF LIFE
EE()  HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
GETLIF  FUNCTION WHICH CALCULATES THE NUMBER OF CYCLES TO FAILURE
INDEX()  COUNTER FOR EFFECTIVE STRESSES
INVLF(N)  INVLF(I) = 1/LIFE(I); DAMAGE FRACTION
I,J,K  COUNTERS FOR VARIOUS DO LOOPS
JMAX  INDEX (LOCATION) OF SEFMAX IN SEFF()
LIFE(N)  LIFE(I) = CALCULATED LIFE FOR STRESS LEVEL STR(I)
N  NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
NEUBER  NEUBER'S RULE MODEL ACCURACY FACTOR
NEUBER' s RULE MODEL ACCURACY FACTOR
NEWTOT  TOTAL NUMBER OF EFFECTIVE STRESS VALUES AFTER FILTERING
OVER  FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
RAINF  FUNCTION WHICH CALCULATES TIME TO FAILURE FOR A GIVEN UNI-AXIAL STRESS-TIME HISTORY
S(NEWTOT)  FILTERED EFFECTIVE STRESSES
SALTF(I)  = sigma alternating, eff,i
SEFFM(2,N)  EFFECTIVE STRESSES AFTER RESEQUENCING/FILTERING/RAINFLOW
SEFFM(1,I)  = sigma max,eff,i
SEFFM(2,1)  = sigma min,eff,i
SEFMAX  LARGEST EFFECTIVE STRESS
SM  SM = EQUIVALENT MEAN STRESS
SMEANF(N)  SMEANF(I) = sigma mean,eff,i
SP(M+1)  RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
STR(N)  STR(I) = EQUIVALENT (COMBINED) STRESS,I
SUNDAM  SUM OF ALL THE DAMAGE FRACTIONS
TEST1{}  1-D ARRAY USED IN FILTERING THE STRESSES
TEST2{}  1-D ARRAY USED IN FILTERING THE STRESSES
TOHIGH  COUNTER FOR AMOUNT OF STRESSES IN EXCESS OF FTU

dump input data
if (iout.eq.20) then
write(8,*) 'rainfl inputs'
write(8,*) 'm :',m, ' period: ',period
write(8,*) 'kt :',kt, ' ftu :',ftu,' fty :',fty
write(8,*) 'numreg :',numreg, ' zroreg :',zroreg
write(8,*) 'szero :',szero, ' kratio :',kratio, ' lnz :',lnz
write(8,*) 'lma(1), lphim(i), mm(i), sbnd(i),
write(8,*) 'EM :',EM, ' TRUNC :',TRUNC, ' NEUB :',NEUB,
write(8,*) 'E(1) :', E
C INITIALIZE ARRAYS

DO 50 I = 1, MAXM
   SP(I) = 0.0
   S(I) = 0.0
   EE(I) = 0.0
   SEFFM(1,I) = 0.0
   SEFFM(2,I) = 0.0
   SALTF(I) = 0.0
   SMEANF(I) = 0.0
   STR(I) = 0.0
   LIFE(I) = 0.0
   INVLIF(I) = 0.0
   INDEX(I) = 0
   TEST1(I) = 0.0
   TEST2(I) = 0.0
50 CONTINUE

SM = 0.0
TOHIGH = 0

C********************* BEGIN RESEQUENCE *********************
C RESEQUENCE effective stresses (needed for rainflow analysis);
C largest effective stress is placed at beginning and end of SP(M+1)
C find SEFMAX, the largest sigma,eff, and JMAX, its location within SEFF(M)

SEFMAX = -1.0E+20
DO 200 I = 1, M
   IF (SEFF(I) .GT. SEFMAX) THEN
      SEFMAX = SEFF(I)
      JMAX = I
   ENDIF
200 CONTINUE

C assign all points from JMAX out, to the beginning of SP()
DO 210 I = 1, M-JMAX+1
    J = JMAX-1 + I
    SP(I) = SEFF(J)
210 CONTINUE

C assign points before JMAX to the end of SP()
J = 0
DO 220 I = M-JMAX+2, M
    J = J + 1
    SP(I) = SEFF(J)
220 CONTINUE

END RESEQUENCE *********************

C*********************** BEGIN FILTER ***********************
C FILTER the resequenced effective stresses, leaving only peaks and
C valleys (excursions larger than TRUNC are deleted during rainflow
C counting) in (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
300 CONTINUE
   if (iout .eq. 20) then
      do 305 i = 2, m
         write(8,*) 'test1 = ', test1(i), ' test2 = ', test2(i)
      continue
   endif
   K = 1
   INDEX(1) = 1
   DO 310 I = 2, M
      IF ((TEST1(I) .NE. 0) .AND. (TEST2(I) .LE. 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,*), test1(i), test2(i)
      write(8,*), 'newtot:
      write(8,*), s(newtot), (s(i), i=1, newtot)
   endif
C***************************************************************************
C***************************************************************************
C***************************************************************************
C***************************************************************************
CONTINUE

C N equals the final number of cycles found
N = I

if (iout.eq.20) then
  write(8,*),'N :',n
  write(8,*),'seffm(2,n) :'
do 12 i=1,n
  write(8,*), seffm(1,i), seffm(2,i)
12 continue
endif

IF (N .EQ. 0) THEN
  TRUNCATION FILTER TOO LARGE -- NO CYCLES LEFT
  SUMDAM = 1.0E-36
  GOTO 710
ENDIF

C calculate alternating and mean effective stresses

DO 500 I=1,N
  SALTF(I) = (SEFFM(1,I) - SEFFM(2,I)) / 2.0
  SMEANF(I) = (SEFFM(1,I) + SEFFM(2,I)) / 2.0
500 CONTINUE

if (iout.eq.20) then
  write(8,*),'saltf(n) :',(saltf(i),i=1,n)
  write(8,*),'smean_(n) ',(smean_(1),i=1,n)
endif

C******* Determine Equivalent Mean Stress, SM(N), (two methods) *********

BIG1 = N
OVER = 0

C We are calculating the equivalent mean stress using neuber's rule
SM = NEUBER (EM, SALTF(BIG1), SMEANF(BIG1), NUMSEG, E, SE, &
       NEUB, OVER)

IF (OVER .EQ. 1) THEN
  SUMDAM = 1.0
  OVER = 0
  GOTO 710
ENDIF

if (iout.eq.20) write(8,*),'sm : ',' sm

C calculate equivalent stresses, STR(N)

ARGM = KT / (1.0 - SM / FTU)

DO 530 I = 1, N
  STR(I) = SALTF(I) * ARGM
  IF (STR(I) .GE. FTU) TOHIGH = TOHIGH + 1
530 CONTINUE

if (iout.eq.20) write(8,*),'str(n) :',(str(i),i=1,n)

C calculate lives and damage fractions: LIFE (N) and INVLIF(N)

DO 600 I=1,N
LIFE(I) = GTLIFE(STR(I), MM, LNA, LPHIM, KRATIO, LNZ,
SBND, ZOREG, 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),',', invlif(n):', invlif(i)
14   continue
endif

LIFE(I) = GTLIFE(STR(I), MM, LNA, LPHIM, KRATIO, LNZ,
SBND, ZOREG, NUMREG, ZZERO)

C Miner's Rule -- sum the damage fractions
C
SUMDAM = 0.0
DO 700 I=1,N
    SUMDAM = SUMDAM + INVLIF(I)
700 CONTINUE
710 CONTINUE

if (iout.eq.20) write(8,*),'sumdam:',sumdam

C calculate fatigue life (time to failure) in seconds
C
RAINF2 = PERIOD / SUMDAM
if (iout.eq.15) then
    chkft = period / sumdam
    write(8,*),'rainfl life',chkft
    write(8,*)
endif

RETURN
END

***************************************************************************
FUNCTION NEUBER (EM, SALT, SMEAN, NUMSEG, E, SE, NEUB, OVER)
C INPUTS: EM, SALT, SMEAN, NUMSEG, E, SE, 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, PRODCT, SALT,
& SE(MAXSEG), SMEAN, ST, TEMP
C
C
********************************************
* NEUBER USES NEUBER'S RULE AND THE STRESS-STRAIN CURVE TO CALCULATE THE *
* MEAN STRESS. PROGRAM ASSUMES THAT THE STRESS STRAIN CURVE IS PIECEWISE LINEAR WITH AT MOST FIVE SECTIONS. *
* PROGRAMMER: L. NEWLIN
* DATE: 13SEP88
* VERSION: THDUCT V3.7, V3.8, V3.9, V4, V4.1
C********************************************
LIST OF VARIABLES

E() STRAIN VALUES FOR EACH SEGMENT
EM YOUNG'S MODULUS BEFORE YIELD
EPSLON CALCULATED STRAIN (WHERE PLASTIC=ELASTIC DEFORMATION)
I CONTROLS DO LOOP FOR EACH SEGMENT
IOUT OUTPUT DUMP CONTROLLER
MAXSEG MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN)
NEUB NEUBER'S RULE MODEL ACCURACY FACTOR
NEUBER TOTAL EQUIVALENT MEAN STRESS
NUMSEG NUMBER OR SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
OVER FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
PRODCT STRESS STRAIN PRODUCT (WHERE PLASTIC-ELASTIC DEFORMATION)
SALT TOTAL ALTERNATING STRESS
SE() 1-DIMENSIONAL ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
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
      ENDIF
    ENDIF
  CONTINUE
ENDIF

800 CONTINUE
ENDIF

IF (ABS(TEMP) .LT. 1.0E-04) THEN
  OVER = 1
  WRITE(8,*), 'THE VALUE PRODCT EXCEEDED STRESS-STRAIN CURVE'
ENDIF
TEMP = TEMP * ABS(ST) / ST
NEUBER = TEMP

IF (IOUT.EQ.25) THEN
  WRITE(8,*), 'VALUES FROM NEUBER'
  WRITE(8,*), 'INPUT VALUES'
  WRITE(8,*), 'EM = ', EM, ' 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
Section 7.2
Low Cycle Fatigue Failure Program

The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the LCF analysis code TRBPWA are given here. The pertinent LCF methodology is given in Section 2.2.2.2. The overall description of the program and the flowcharts are given in Section 5.2. The user's guide for running TRBPWA is given in Section 6.2.

7.2.1 TRBPWA Program

7.2.1.1 Program Tree Structure
The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for TRBPWA using Uniform variation on the materials shape parameter $m$ is given in Figure 7-5, while the tree structure for the truncated Normal case is given in Figure 7-6. 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.

7.2.1.2 List of Subprograms
A list of subprograms and their purposes is given in Table 7-5. The section numbers where the subprograms are described by means of flowcharts are given next to the names.
Figure 7-5  Tree Structure for Program TRBPWA for the Uniform Variation in Materials Shape Parameter $m$
Figure 7-6  Tree Structure for Program TRBPWA 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>ADDREG(^1)</td>
<td>4.1.3.9</td>
<td>Adds the (m) ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.</td>
</tr>
<tr>
<td>ADDRGN(^1)</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>BETAGN(^2)</td>
<td>4.4.5</td>
<td>Generates Beta((a, b, \rho, \theta)) random variates.</td>
</tr>
<tr>
<td>CONCAV(^3)</td>
<td>4.1.3.10</td>
<td>Adjusts the upper bound of the posterior ranges on (m) to be consistent with concavity constraints.</td>
</tr>
<tr>
<td>CONVRT(^4)</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>EXPCTD(^5)</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.</td>
</tr>
<tr>
<td>FINDM(^6)</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 for each life region implied by the constraint on the coefficient of variation of fatigue strength (C) by using Equations 2-28 through 2-32.</td>
</tr>
<tr>
<td>FINDMN(^6)</td>
<td>4.1.5.2</td>
<td>Obtains the value of (m) for each life region by sampling from the appropriate truncated Normal distribution on (m).</td>
</tr>
<tr>
<td>FINDSB</td>
<td>4.1.5.7</td>
<td>Calculates the life region &quot;tie-points&quot; or stress values which correspond to the &quot;life boundaries&quot; conditional on the randomly selected (m) for each region. Also calculates (K), characterizing the specific material S/N data set, which is a function of (\beta_0) and (k).</td>
</tr>
<tr>
<td>FNDRNG(^7)</td>
<td>4.1.3.8</td>
<td>Combines the 95% confidence interval, (J_0), with the implicit and explicit constraints on (m) to obtain posterior credibility ranges on (m) for each life region.</td>
</tr>
<tr>
<td>GAM</td>
<td>4.4.4</td>
<td>Generates Gamma((\alpha, 1)) random variates.</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.</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>4.1.3.7</td>
<td>Calculates (\sigma^2), Equation 2-49, the extent of departures from the multiple heat median S/N curve warranted by the information available.</td>
</tr>
<tr>
<td>INFAGG(^8)</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>NAME</td>
<td>SECTION</td>
<td>PURPOSE</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</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 $l_o$ for $C$, and $J_o$ for $m$, for each region by using Equations 2-24 and 2-26.</td>
</tr>
<tr>
<td>KBETA</td>
<td>4.1.5.5</td>
<td>Calculates $k$ and $\beta_o$ from the sample mean and variance of $Z$, where $Z$ is a function of stress, life, the life region boundaries, and the $m$'s by using Equation 2-42.</td>
</tr>
<tr>
<td>KOMO$^9$</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.</td>
</tr>
<tr>
<td>MUSIG$^{10}$</td>
<td>4.1.3.13</td>
<td>Calculates the posterior Normal distribution parameters: mean $m_o$ and standard deviation $\sigma_o$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NORMGN$^{11}$</td>
<td>4.4.3</td>
<td>Generates Normal($\mu$, $\sigma^2$) random variates.</td>
</tr>
<tr>
<td>NORRNG$^7$</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$^{12}$</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$^{13}$</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>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 TRBPWD and RELATD; calls CONVRT to transform the stress data to a stress ratio of $-1.0$; and echoes the data to TRBPWO 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 stress, life, the life region boundaries, and the $m$'s by using Equation 2-42.</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>4.1.3.4</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; to be used in the credibility range calculations.</td>
</tr>
</tbody>
</table>
### Table 7-5 List of Subprograms For Program TRBPWA (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRBPWA</td>
<td>5.2.2</td>
<td>The main routine that controls the logical flow and performs the driver transformation calculations of the low cycle fatigue turbine disk program.</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>4.1.11</td>
<td>Performs premature program termination, when required.</td>
</tr>
<tr>
<td>TRNSFM¹⁵</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(β, η(β)) random variates.</td>
</tr>
</tbody>
</table>

¹ No data regions to the right are discussed on Page 2-17.
² The Beta distribution is discussed on Page 2-25.
³ Concavity constraints are discussed on Pages 2-13 through 2-14.
⁴ The stress transformation is discussed on Page 2-7.
⁵ The median S/N curve parameter estimation calculations are described on Pages 2-15 through 2-18.
⁶ Selection of the \( \{m_j\} \) parameters is discussed on Page 2-15.
⁷ Combining information to obtain the posterior credibility ranges on \( m \) is discussed on Page 2-13.
⁸ The information aggregation calculations are discussed on Pages 2-6 through 2-14.
⁹ Extension of the S/N curve to the left is discussed on Page 2-17.
¹⁰ Calculation of the truncated Normal distribution parameters is discussed on Page 2-14.
¹¹ The Normal distribution is discussed on Page 2-23.
¹² The parameter estimation calculations are discussed on Pages 2-15 through 2-18.
¹³ The Uniform distribution is discussed on Page 2-23.
¹⁴ The need for saving m's is discussed on Page 2-15.
¹⁵ The S/N data transformation is discussed on Page 2-16.
### 7.2.1.3 Description of Variables

A list of variables used in the ATD-HPFTP second stage turbine disk LCF code, TRBPWA, is given in Table 7-6. 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 7-6 List of Variables For Program TRBPWA
(Footnotes are at the end of the table)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLM(MAXMM, MAXREG)</td>
<td>RE</td>
<td>2-D array containing the materials model shape parameters (m’s) for each life region to be used in the truncated Normal median S/N curve calculation.¹</td>
</tr>
<tr>
<td>BIGK(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the materials model location parameter K, Equation 2-12, where A = Kᵐ.</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>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 βₒ, Equation 2-11, which characterizes the intrinsic variation of the S/N data set.</td>
</tr>
<tr>
<td>CG</td>
<td>RE</td>
<td>C_G (psi/°°F) in Equation 2-98, the sensitivity of stress to the deviation from nominal thermal gradient ΔGᵀ, due to the deviation from nominal coolant fluid temperature ΔT_f.</td>
</tr>
<tr>
<td>CG1</td>
<td>RE</td>
<td>C_G₁ in Equation 2-96, the sensitivity of thermal gradient to the deviation from nominal coolant fluid temperature ΔT_f for ΔT_f &lt; 0.</td>
</tr>
<tr>
<td>CG2</td>
<td>RE</td>
<td>C_G₂ in Equation 2-96, the sensitivity of thermal gradient to the deviation from nominal coolant fluid temperature ΔT_f for ΔT_f ≥ 0.</td>
</tr>
</tbody>
</table>

² Improved stress robustness in S/N calculation provided by using 3-D array containing models' shape parameters for each life region.
Table 7-6  List of Variables For Program TRBPWA (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM</td>
<td>RE</td>
<td>$C_m$ (psi/°F) in Equation 2-97, the sensitivity of stress to the deviation from nominal metal temperature $\Delta T_m$, due to the deviation from nominal coolant fluid temperature $\Delta T_f$.</td>
</tr>
<tr>
<td>CMF</td>
<td>RE</td>
<td>$C_{mf}$ (psi/°F) in Equation 2-95, the sensitivity of metal temperature to the deviation from nominal coolant fluid temperature $\Delta T_f$.</td>
</tr>
<tr>
<td>CS</td>
<td>RE</td>
<td>$C_S$ in Equation 2-94, the rotational speed correction factor, (selected speed / reference speed)$^2$.</td>
</tr>
<tr>
<td>DELGT</td>
<td>RE</td>
<td>$\Delta G_T$ (°F) in Equation 2-96, the deviation from the nominal thermal gradient in the blade attachment area as a result of the randomly selected $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTA</td>
<td>RE</td>
<td>$\Delta T_f$ lower bound of Beta distribution 1.</td>
</tr>
<tr>
<td>DELTB</td>
<td>RE</td>
<td>$\Delta T_f$ upper bound of Beta distribution 1.</td>
</tr>
<tr>
<td>DELTC</td>
<td>RE</td>
<td>$\Delta T_f$ lower bound of Beta distribution 2.</td>
</tr>
<tr>
<td>DELTD</td>
<td>RE</td>
<td>$\Delta T_f$ upper bound of Beta distribution 2.</td>
</tr>
<tr>
<td>DELTE</td>
<td>RE</td>
<td>Decimal equivalent percentage weight occurring in Beta distribution 1 of the deviation from nominal coolant fluid temperature $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTF</td>
<td>RE</td>
<td>$\Delta T_f$ (°F) in Equation 2-95, the randomly selected deviation from nominal coolant fluid temperature.</td>
</tr>
<tr>
<td>DELTHI</td>
<td>RE</td>
<td>Upper bound of the randomly selected Beta distribution for the deviation from nominal coolant fluid temperature $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTILO</td>
<td>RE</td>
<td>Lower bound of the randomly selected Beta distribution for the deviation from nominal coolant fluid temperature $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTM</td>
<td>RE</td>
<td>$\Delta T_m$ (°F) in Equation 2-95, the deviation from the nominal metal temperature in the blade attachment area as a result of the randomly selected $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTR</td>
<td>RE</td>
<td>Randomly selected Beta distribution location parameter $\rho$ for the deviation from the nominal coolant fluid temperature $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTR1</td>
<td>RE</td>
<td>$\rho$ Uniform distribution lower bound of Beta distribution 1 of $\Delta T_f$.</td>
</tr>
</tbody>
</table>
Table 7-6  List of Variables For Program TRBPWA (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTR2</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution 1 of $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTR3</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution 2 of $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTR4</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution 2 of $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTT</td>
<td>RE</td>
<td>Randomly selected Beta distribution shape parameter $\theta$ for the deviation from the nominal coolant fluid temperature $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTT1</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution 1 of $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTT2</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution 1 of $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTT3</td>
<td>RE</td>
<td>Uniform distribution lower bound of Beta distribution 2 of $\Delta T_f$.</td>
</tr>
<tr>
<td>DELTT4</td>
<td>RE</td>
<td>Uniform distribution upper bound of Beta distribution 2 of $\Delta T_f$.</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 that calculates the fatigue cycles to failure at a given stress, given by Equation 2-48.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls inner DO loop.</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.²</td>
</tr>
<tr>
<td>K</td>
<td>INT</td>
<td>Controls outer DO loop.</td>
</tr>
<tr>
<td>KD</td>
<td>RE</td>
<td>$K_d$ in Equation 2-102, the stress factor to adjust for using 2-D stress analyses.</td>
</tr>
<tr>
<td>KRATIO</td>
<td>RE</td>
<td>Ratio of MED $K$/MED $K$ in Equation 2-48. KRATIO is constant over life regions for the materials model.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>KT</td>
<td>RE</td>
<td>$K_t$ in Equation 2-92, the local stress concentration factor used in stress analysis to obtain the reference stress from the equivalent elastic stress.</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>LAMG</td>
<td>RE</td>
<td>$\lambda_g$ in Equation 2-98, the factor characterizing the sensitivity of stress due to thermal gradient.</td>
</tr>
<tr>
<td>LAMKD</td>
<td>RE</td>
<td>$\lambda_{K_d}$ in Equation 2-103, the randomly selected model accuracy factor for the parameter $K_d$.</td>
</tr>
<tr>
<td>LAMKDA</td>
<td>RE</td>
<td>Uniform distribution lower bound.</td>
</tr>
<tr>
<td>LAMKDB</td>
<td>RE</td>
<td>Uniform distribution upper bound.</td>
</tr>
<tr>
<td>LAMKT</td>
<td>RE</td>
<td>$\lambda_{K_t}$ in Equation 2-103, the randomly selected model accuracy factor for the parameter $K_t$.</td>
</tr>
<tr>
<td>LAMKTA</td>
<td>RE</td>
<td>Uniform distribution lower bound.</td>
</tr>
<tr>
<td>LAMKTB</td>
<td>RE</td>
<td>Uniform distribution upper bound.</td>
</tr>
<tr>
<td>LAMM</td>
<td>RE</td>
<td>$\lambda_m$ in Equation 2-97, the factor characterizing the sensitivity of stress due to metal temperature.</td>
</tr>
<tr>
<td>LIFE(MAXLIF)</td>
<td>RE</td>
<td>1-D array containing values of the lives generated by program TRBPWA. The lives are sorted values for the left-hand tail simulated failure distribution.</td>
</tr>
<tr>
<td>LNZ</td>
<td>RE</td>
<td>$\ln(Z)$ in Equation 2-48, the Normal(0, PVAR) random variate for the materials process variation aspect of the materials model.</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 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.$^1$ The maximum number of $m$'s is 20,000.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</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>MEDM(MAXMM)</td>
<td>RE</td>
<td>1-D array containing the empirical median m for each life region of the S/N curve.³</td>
</tr>
<tr>
<td>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 φ.</td>
</tr>
<tr>
<td>MM(0:MAXREG)</td>
<td>RE</td>
<td>mj in Equation 2-12, 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.⁴</td>
</tr>
<tr>
<td>MU(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution mean of the materials shape parameter m for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>NBLIFE</td>
<td>INT</td>
<td>Number of B-lives to be obtained from the simulated failure distribution.²</td>
</tr>
<tr>
<td>NBND(0:MAXREG)</td>
<td>RE</td>
<td>Ni, i+1 in Equation 2-35, 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>NEWLIFE</td>
<td>RE</td>
<td>Fatigue life value (cycles) returned from call to function GTLIFE.</td>
</tr>
<tr>
<td>NF(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values from the array RAWNF() for the specific material S/N data set partitioned into life regions.</td>
</tr>
<tr>
<td>NHYPER</td>
<td>INT</td>
<td>The outer loop size.</td>
</tr>
<tr>
<td>NLIFE</td>
<td>INT</td>
<td>The inner loop size.</td>
</tr>
<tr>
<td>NLIFET</td>
<td>INT</td>
<td>Total number of lives calculated by program TRBPWA. Value of NHYPER * NLIFE.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</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>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, 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, the materials intrinsic variation, or scatter, given by a Weibull(( \beta_o ), ( \eta_o(\beta_o) )) random variate.</td>
</tr>
<tr>
<td>PSIG</td>
<td>RE</td>
<td>( \sigma ) in Equation 2-48, the value of SQRT(PVAR).</td>
</tr>
<tr>
<td>PVAR</td>
<td>RE</td>
<td>( \sigma^2 ) in Equation 2-48, characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.</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>REFSPEED</td>
<td>RE</td>
<td>Reference Speed ( \omega_o ) (rpm) in Equation 2-94, the speed corresponding to the nominal mechanical stress SMM due to rotor speed.</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>SG</td>
<td>RE</td>
<td>( S_G ) (psi) in Equation 2-98, the nominal stress due to the thermal gradient alone, at reference speed REFSPD and nominal coolant fluid temperature.</td>
</tr>
<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>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>SMM</td>
<td>RE</td>
<td>$S_{Mo}$ (psi) in Equation 2-94, the nominal mechanical stress due to centrifugal effects only at the reference speed REFSPD.</td>
</tr>
<tr>
<td>SPDMU</td>
<td>RE</td>
<td>Mean $\mu$ of Normally distributed speed (rpm).</td>
</tr>
<tr>
<td>SPDSIG</td>
<td>RE</td>
<td>Standard deviation $\sigma$ of Normally distributed speed (rpm).</td>
</tr>
<tr>
<td>SPEED</td>
<td>RE</td>
<td>$\omega$ (rpm) in Equation 2-94, the randomly selected rotational speed for rotational speed correction factor CS.</td>
</tr>
<tr>
<td>SR</td>
<td>RE</td>
<td>$S_R$ (psi) in Equation 2-103, the value of the maximum reference stress used to calculate life. The stress ratio is assumed to be greater than or equal to 0.</td>
</tr>
<tr>
<td>STM</td>
<td>RE</td>
<td>$S_{mo}$ (psi) in Equation 2-97, the nominal stress due to the metal temperature alone, at the reference speed REFSPD and nominal coolant fluid temperature.</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>SZERO</td>
<td>RE</td>
<td>Stress tensile test point, $S_0$ (psi). ^8</td>
</tr>
<tr>
<td>TEST</td>
<td>RE</td>
<td>Uniform(0, 1) random variate used to determine Beta distribution for $\Delta T_f$.</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 $S_0$. ^8</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 &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 $S_0$. ^8</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; 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, the randomly selected process variation shift factor given by a Lognormal(0, PVAR) random variate.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| ZROREG        | INT  | 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.$^8$

1. The need for saving $m$'s is discussed on 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 on Page 2-15.
4. See Section 2.1.2.3 for a discussion on process variation in materials.
5. $m^*$ of the posterior density of $m$ is discussed on Page 2-14.
6. The posterior credibility ranges $\pi(m)$ are discussed on Page 2-13.
7. $\sigma_*$ of the posterior density of $m$ is discussed on Page 2-14.
8. Extension of the S/N curve to the left using the tensile point is discussed on Page 2-17.
### 7.2.1.4 Program TRBPWA Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program TRBPWA Listing</td>
<td></td>
</tr>
<tr>
<td>Temporal Order, Uniform</td>
<td></td>
</tr>
<tr>
<td>Distribution</td>
<td>7-256</td>
</tr>
<tr>
<td>Program TRBPWA Listing</td>
<td></td>
</tr>
<tr>
<td>Temporal Order, Truncated</td>
<td></td>
</tr>
<tr>
<td>Normal Distribution</td>
<td>7-257</td>
</tr>
<tr>
<td>TRBPWA</td>
<td>7-259</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-266</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-267</td>
</tr>
<tr>
<td>GAM</td>
<td>7-267</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-268</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-269</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-274</td>
</tr>
<tr>
<td>INIT</td>
<td>7-275</td>
</tr>
<tr>
<td>RCE</td>
<td>7-276</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-283</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-284</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-287</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-290</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-292</td>
</tr>
<tr>
<td>FNDARNG</td>
<td>7-293</td>
</tr>
<tr>
<td>ADDREG</td>
<td>7-297</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-298</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-299</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-300</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-302</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-304</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-306</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-308</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-310</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-312</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-314</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-315</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-316</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-317</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-318</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-319</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-320</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-321</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-322</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-323</td>
</tr>
</tbody>
</table>

TRBPWA Version 3
Program TRBPWA Listing Temporal Order, Uniform Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRBPWA</td>
<td>7-259</td>
</tr>
<tr>
<td>INFA 4G</td>
<td>7-269</td>
</tr>
<tr>
<td>INIT</td>
<td>7-275</td>
</tr>
<tr>
<td>RCE</td>
<td>7-276</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-283</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-284</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-290</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-287</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-292</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-293</td>
</tr>
<tr>
<td>ADDREG</td>
<td>7-297</td>
</tr>
<tr>
<td>CONCA V</td>
<td>7-298</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-299</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-300</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-315</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-316</td>
</tr>
<tr>
<td>KBE T A</td>
<td>7-317</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-318</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-319</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-321</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>PITYRV</td>
<td>7-266</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-312</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-308</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-310</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-315</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-316</td>
</tr>
<tr>
<td>KBE T A</td>
<td>7-317</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-318</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-319</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-314</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-267</td>
</tr>
<tr>
<td>GAM</td>
<td>7-267</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-314</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>PITYRV</td>
<td>7-266</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-320</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-321</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-322</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-268</td>
</tr>
</tbody>
</table>
Program TRBPWA Listing Temporal Order, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRBPWA</td>
<td>7-259</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-269</td>
</tr>
<tr>
<td>INIT</td>
<td>7-275</td>
</tr>
<tr>
<td>RCE</td>
<td>7-276</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-283</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-284</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-290</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-302</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-292</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-304</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-306</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-298</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-266</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-308</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-312</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-314</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-315</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-316</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-317</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-318</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-319</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-314</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>BETAGN</td>
<td>7-267</td>
</tr>
<tr>
<td>GAM</td>
<td>7-267</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-314</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>PRYRV</td>
<td>7-266</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-320</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-312</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-321</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-322</td>
</tr>
<tr>
<td>INSORT</td>
<td>7-268</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-323</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-300</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-315</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-316</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-317</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-318</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-319</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-321</td>
</tr>
</tbody>
</table>

7 - 258
C PROGRAM TRBPWA CONTROLS THE FLOW OF LOGIC OF THE LOW CYCLE
C FATIGUE ANALYSIS OF THE ATD-HPTP 2ND STAGE TURBINE DISK
C PROGRAMMER: L. NEWLIN
C DATE: 9SEP91
C VERSION: 3 (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
C ********************************************************************************
C
PROGRAM TRBPWA
C
SUBPROGRAMS: INFAGG, PAREST, PRYRV, BETAGN, NORMGN, WEIBGN,
GTLIFE, INSORT, TRGPN, SORTM, EXPECTD
FILES: 1:TRBPWD-OLD; 5:RELATD-OLD; 6:RELATO-NEW;
7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
NOTE: 5 & 6 ARE OPENED IN 'INFAGG'
C
C IMPLICIT NONE

INTEGER MAXBLF, MAXDAT, MAXLIF, MAXMM, MAXREG
PARAMETER (MAXBLF=MAXMM - 20001, MAXDAT=50, MAXLIF = 10000,
MAXMM = 20001, MAXREG = 3)

COMMON I
INTEGER BLFPIS(MAXBLF), I, IOUT, J, K, L, M, MCOUNT, MID,
MPROC, NBLIFE, NHYPER, NLLIFE, NLLIFET, NMED,
INTS(MAXREG), NSYM, NUMREG, VARY, ZR

DOUBLE PRECISION RAND

REAL ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1,
& BLFPER(MAXBLF), BZERO, CG, CQ1, CQ2, CM, CPU, CS,
& DELT, DELTA, DELTB, DELTC, DELTD, DELTE, DELTF, DELTHI,
& DELTLO, DELTN, DELTR, DELTR1, DELTR2, DELTR3, DELTR4,
& DELTI, DELTI1, DELTI2, DELTI3, DELTI4, FACTR, FIFTY,
& FTU, FYT, GTLIFE, K, KRATIO, KT, LAMG, LAMK, LAMKM,
& LAMKB, LAMKT, LAMKTA, LAMKTS, LAMM, LIFE(MAXLIF), LNZ,
& MEDM(MAXMM), MINPM, MN(0:MAXREG), MU(MAXREG),
& NBND(0:MAXREG), NEWLIF, NF(MAXDAT, MAXREG), NMED,
& SIG(MAXREG), SM, SPDMU, SPDSDG, SPEED, SR, STM,
& STR(MAXDAT, MAXREG), SZERO, TEST, TRBIGK(0:MAXREG),
& TRSBND (0:MAXREG), Z
C ** SEE BOTTOM OF PROGRAM FOR LIST OF VARIABLES

OPEN (1, FILE = 'TRBPWD', STATUS = 'OLD')
OPEN (3, FILE = 'TRBPWO', 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, TRBPWA = 15) = ', IOUT
READ(1,*) NLLIFE
WRITE(8,*) ', INNER LOOP SIZE = ', NLLIFE
READ(1,*) NHYPER
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 = ', 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

C

7 - 259
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'
ENDIF

IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) 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 TRBPWD
READ(1,*) DELTA, DELTB, DELTR1, DELTR2, DELT1, DELT2,
     DELTC, DELTD, DELTR3, DELTR4, DELT3, DELT4,
     DELTE
READ(1,*) SPDMU, SPDSIG
READ(1,*) LAMKDA, LAMKDB
READ(1,*) LAMKTA, LAMKTB
READ(1,*) KD, KT, SHM, REFSPD
READ(1,*) STM, CMF, CH
READ(1,*) SG, CG1, CG2, CG

C ** ECHO DATA TO TRBPWO
WRITE(3,900) DELTA, DELTB, DELTR1, DELTR2, DELT1, DELT2,
        DELTC, DELTD, DELTR3, DELTR4, DELT3, DELT4,
        DELTE
WRITE(3,902) SPDMU, SPDSIG
WRITE(3,903) LAMKDA, LAMKDB
WRITE(3,904) LAMKTA, LAMKTB
WRITE(3,905) KD, KT, SHM, REFSPD
WRITE(3,906) STM, CMF, CH
WRITE(3,907) SG, CG1, CG2, CG

C ** CALL TO PERFORM THE INFORMATION AGGREGATION MODEL ASPECTS
C OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZOREG, NUMREG,
     1, NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)
IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)
MCOUNT = 0

C ** INITIALIZE VARIABLES
DO 10 K = 1, MAXLIFE
   LIFE(K) = 1.0E+36
10 CONTINUE
NLIFET = NHYPER * NLIFE

C ** OUTER LOOP -- THIS LOOP SAMPLES HYPER-PARAMETER SETS
DO 150 K = 1, NHYPER

C ** CALL PRYRV TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS
CALL RANDOM (TEST, RAND)
IF (TEST .LE. DELTE) THEN
CALL PRYV (RAND, DELTR1, DELTR2, DELTT1, DELTT2, DELTR, DELTT)
DELTL0 = DELTA
DELTHI = DELTB
ELSE
CALL PRYV (RAND, DELTR3, DELTR4, DELTT3, DELTT4, DELTR, DELTT)
DELTL0 = DELTC
DELTHI = DELTD
ENDIF
IF (IOUT .EQ. 15) THEN
WRITE(* , *) 'TEST =', 'TEST, ' DELTE =', DELTE
WRITE(* , *) 'DELTL0 =', ' DELTL0, ' DELTHI =', DELTHI
ENDIF
C ** CALL PAREST TO PERFORM THE PARAMETER ESTIMATION ASPECT
OF THE MATERIALS CHARACTERIZATION MODEL CALCULATIONS
CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND, NBND, STR, BIGK, BZERO, MM, SBND)
C ** OBTAIN MATERIALS PROCESS VARIATION IF DESIRED
CALL NORMGN (RAND, 0.0, PSIG, 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 FAILURE LIVES
DO 200 I = 1, NLIFE
C ** SELECT DRIVERS FOR CALCULATING LIFE
CALL BETAGN (RAND, DELTR, DELTT, DELTL0, DELTHI, DELTF)
CALL NORMGN (RAND, SPDMU, SPDSIG, SPEED)
CALL PRYV (RAND, LAMKDA, LAMKDB, LAMKTA, LAMKTB, LAMED, LAMKT)
MINPHI = 1.0E+36
DO 225 M = 1, NSYM
   CALL WEIBGN (BZERO, RAND, PHI)
   MINPHI = MIN (PHI, MINPHI)
225 CONTINUE
PHI = MINPHI
IF (VARY .EQ. 0) PHI = 1.0
IF (IOUT .EQ. 15) THEN
   WRITE(* , *) 'DELTF =', ' DELTF, ' SPEED =', 'SPEED
   WRITE(* , *) 'LAMKDA =', ' LAMKDB, ' LAMKTA =', ' LAMKTB
   WRITE(* , *) 'PHI =', ' PHI
ENDIF
C FIND SO
FACTR = PHI * KRATIO * Z
DO 230 L = ZROREG, NUMREG
   TRSBND(L) = FACTR * SBND(L)
   TRBIGK(L) = BIGK(L)
230 CONTINUE
TRSBND(0) = SBND(0)
IF (ZROREG .EQ. 0) CALL KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK, FACTR, NUMREG)
IF (IOUT .EQ. 15) THEN
  WRITE(8,*),'ZROREG = ', ZROREG, ' NUMREG = ', NUMREG
  WRITE(8,*),'KRATIO = ', KRATIO, ' FACTR = ', FACTR
  WRITE(8,*),'PHI = ', PHI, ' Z = ', Z
  DO 250 L = ZROREG, NUMREG
    WRITE(8,*),'L = ', L, ' MM = ', MM(L)
    WRITE(8,*),'SBND = ', SBND(L), ' TRSBND = ', TRSBND(L)
    WRITE(8,*),'BIGK = ', BIGK(L), ' TRBIGK(L)'
  CONTINUE
ENDIF

C ** PERFORM DRIVER TRANSFORMATION
CS = (SPEED / REFSPD) ** 2
DELM = CMF * DELTF
LAMM = 1.0 + (CM * DELTM / STM)
IF (DELTF.LT.0.0) THEN
  DELGT = CG1 * DELTF
ELSE
  DELGT = CG2 * DELTF
ENDIF
LAMG = 1.0 + (CG * DELGT / SG)
IF (IOUT .EQ. 15) THEN
  WRITE(8,*),'CS = ', CS
  WRITE(8,*),'DELM = ', DELTM, ' LAMM = ', LAMM
  WRITE(8,*),'DELGT = ', DELGT, ' LAMG = ', LAMG
ENDIF
SR = LAMKD * LAMKT * (CS * SMM + STM + LAMG * SG)
& * KD / KT
IF (IOUT .EQ. 15) WRITE(8,*),'SR = ', SR

C ** CALL GTLIFE TO CALCULATE FATIGUE LIFE
NEWLIF = GTLIFE (SR, MM, TRBIGK, PHI, KRATIO, LNZ, TRSBND, ZROREG, NUMREG, SZERO)
& IF (IOUT .EQ. 15) WRITE(8,*),'SR = ', SR,
& IF (NLIFET .GE. 100) CALL INSORT (NEWLIF, LIFE, NLIFET)
200 CONTINUE
150 CONTINUE
IF (NLIFET .GE. 100) THEN
C ** PRINT SORTED LIVES
  DO 300 J = I, (NLIFET / 100)
    WRITE(9,*), J, FLOAT(J)/FLOAT(NLIFET), LIFE(J)
  CONTINUE
C ** INITIALIZE VARIABLES
  DO 325 J = 1, MAXBLF
    BLFPOS(J) = 0
  CONTINUE
C ** PRINT EMPIRICAL BLIVES
  FIFTY = 0.50E0
  WRITE(3,910)
  DO 350 J = 1, NLIFE
    BLFPOS(J) = NINT (BI_PER(J)* (NLIFET))
    WRITE(3,920) BLFPER(J), LIFE(BLFPOS(J))
  CONTINUE
  WRITE(3,920) FIFTY, LIFE(NLIFET/2)
300 CONTINUE
325 CONTINUE
350 CONTINUE
WRITE(3,920)

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 = I, NUMREG
        MEDM(L) = ALLM(MID,L)
    CONTINUE

    CALL EXPCTD (I, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND, BIGK1, BZERO)

ENDIF

C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO TRBPWO

900 FORMAT(2X, 'Copyright (C) 1990, California Institute of', &
' NASA Contract NAS7-918 is acknowledged.', &
' ///, 33X, 'INPUT DATA', &
' ///, 14X, 'DRIVERS', 25X, 'PARAMETER DISTRIBUTIONS', &
' ///, 48X, 'RHO', 16X, 'THETA')

901 FORMAT(/, 2X, 'DELTA Tf', 8X, 'Be(', F6.1, ', ', F7.1, ')', 3X, &
' U(', F7.5, ',', F8.5, ')', 4X, 'U(', F4.1, ',', F5.1, ')', &
' 18X, 'Be(', F6.1, ',', F7.1, ')', 3X, 'U(', F7.5, ',', F8.5, ')', &
' 4X, 'U(', F4.1, ',', F5.1, ')', /, 18X, 'TEST = ', F4.2)

902 FORMAT(/, 2X, 'SPEED (RPM)', 5X, 'NORMAL: MEAN = ', 2X, F6.0, 5X, &
' STAND. DEV. = ', '2X, F5.0)

903 FORMAT(/, 2X, 'LAMBDA Kd', 7X, 'U(', F8.5, ',', F9.5, ')')

904 FORMAT(/, 2X, 'LAMBDA Kt', 7X, 'U(', F8.5, ',', F9.5, ')')

905 FORMAT(/, 28X, 'OTHER LOADS INPUT', &
' ///, 2X, 'STRESS ADJUSTMENT, Kd', 37X, F5.3, &
' ///, 2X, 'STRESS CONCENTRATION, Kt', 34X, F5.3, &
' ///, 2X, 'MECHANICAL STRESS (PSI)', 30X, F8.1, &
' ///, 2X, 'ROTATIONAL SPEED (RPM)', 32X, F6.0)

906 FORMAT(/, 2X, 'STRESS DUE TO METAL TEMPERATURE (PSI)', 16X, F8.1, &
' ///, 2X, 'SENSITIVITY OF METAL TEMPERATURE TO DELTA Tf', 14X, &
' F7.5, ///, 2X, 'SENSITIVITY OF STRESS DUE TO Tmetal (PSI/F)', &
' 14X, F5.2)

907 FORMAT(/, 2X, 'STRESS DUE TO THERMAL GRADIENT (PSI)', 17X, F8.1, &
' ///, 2X, 'SENSITIVITY OF THERMAL GRADIENT TO DELTA Tf', &
' 7X, 'FOR DELTA Tf < 0', 37X, F5.3, &
' 7X, 'FOR DELTA Tf > 0', 36X, F5.3, &
' ///, 2X, 'SENSITIVITY OF STRESS DUE TO THERM. GRAD. (PSI/F)', &
' 5X, F8.2)

910 FORMAT(/, 2X, 'B LIVES: EMPIRICAL', /)

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

STOP
END

C******************************************************************************
C SAMPLE 'TRBPWO' INPUT FILE
C******************************************************************************
C 75 CONSORTNUM RANDOM NUMBER SEED
C 0 IOUTPUT DUMP CONTROLLER
C 100 INNER LOOP SIZE
C 200 OUTER LOOP SIZE

7 - 263
<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLM</td>
<td>2-D array containing M values to be sorted for each region</td>
</tr>
<tr>
<td>BIGK1</td>
<td>Dummy variable used during calls to EXPCTD</td>
</tr>
<tr>
<td>BLFPER</td>
<td>1-D array containing user specified lives to be provided</td>
</tr>
<tr>
<td>BLFPOS</td>
<td>1-D array containing position in life() of empirical lives</td>
</tr>
<tr>
<td>CM</td>
<td>Value of Weibull parameter, beta, characterizing S-N data set</td>
</tr>
<tr>
<td>SG</td>
<td>SENSITIVITY OF STRESS DUE TO VARIATION OF THERMAL GRADIENT DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE (PSI/deg F)</td>
</tr>
<tr>
<td>SG1</td>
<td>SENSITIVITY OF THERMAL GRADIENT DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE FOR DELTA T &amp; lt; 0</td>
</tr>
<tr>
<td>SG2</td>
<td>SENSITIVITY OF THERMAL GRADIENT DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE FOR DELTA T &amp; ge; 0</td>
</tr>
<tr>
<td>CM</td>
<td>SENSITIVITY OF STRESS DUE TO VARIATION OF METAL TEMPERATURE DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE (PSI/deg F)</td>
</tr>
<tr>
<td>CMF</td>
<td>SENSITIVITY OF STRESS DUE TO VARIATION OF METAL TEMPERATURE DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE (PSI/deg F)</td>
</tr>
<tr>
<td>CS</td>
<td>ROTATIONAL SPEED CORRECTION FACTOR -- EQUAL TO (SELECTED SPEED / REFERENCE SPEED) ** 2</td>
</tr>
<tr>
<td>DELGT</td>
<td>DELTA THERMAL GRADIENT (deg F) IS THE DEVIATION FROM NOMINAL THERMAL GRADIENT IN THE BLADE ATTACHMENT AREA AS A RESULT OF THE SELECTED DELTA T</td>
</tr>
<tr>
<td>DELTA</td>
<td>DELTA T THERMAL GRADIENT (deg F) IS THE DEVIATION FROM NOMINAL THERMAL GRADIENT IN THE BLADE ATTACHMENT AREA AS A RESULT OF THE SELECTED DELTA T</td>
</tr>
<tr>
<td>DELTB</td>
<td>DELTA T LOWER BOUND -- HYPER-DISTRIBUTION 1</td>
</tr>
<tr>
<td>DELTD</td>
<td>DELTA T UPPER BOUND -- HYPER-DISTRIBUTION 1</td>
</tr>
<tr>
<td>DELT</td>
<td>DELTA T LOWER BOUND -- HYPER-DISTRIBUTION 2</td>
</tr>
<tr>
<td>DELTD</td>
<td>DELTA T UPPER BOUND -- HYPER-DISTRIBUTION 2</td>
</tr>
<tr>
<td>DELTE</td>
<td>PERCENTAGE OCCURRING IN HYPER-DISTRIBUTION 1 -- DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE, DELTA T</td>
</tr>
<tr>
<td>DELT</td>
<td>SELECTED DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE, DELTA T</td>
</tr>
</tbody>
</table>

7 - 264
DELTHI  SELECTED LAMBDA UPPER BOUND FOR DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE, DELTA Tf
DELTL0  SELECTED LAMBDA LOWER BOUND FOR DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE, DELTA Tf
DELTM  DELTA Tf IS THE DEVIATION FROM THE NOMINAL METAL TEMPERATURE IN THE BLADE ATTACHMENT AREA AS A RESULT OF THE SELECTED DELTHI
DELTR  SELECTED RHO FOR DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE, DELTA Tf
DELTR1  DELTA Tf - RHÓ LOWER BOUND - HYPER-DISTRIBUTION 1
DELTR2  DELTA Tf - RHÓ UPPER BOUND - HYPER-DISTRIBUTION 2
DELTR3  DELTA Tf - RHÓ LOWER BOUND - HYPER-DISTRIBUTION 3
DELTR4  DELTA Tf - RHÓ UPPER BOUND - HYPER-DISTRIBUTION 4
DELTT  SELECTED THETA FOR DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE, DELTA Tf
DELTT1  DELTA Tf - THETA LOWER BOUND - HYPER-DISTRIBUTION 1
DELTT2  DELTA Tf - THETA UPPER BOUND - HYPER-DISTRIBUTION 2
DELTT3  DELTA Tf - THETA LOWER BOUND - HYPER-DISTRIBUTION 3
DELTT4  DELTA Tf - THETA UPPER BOUND - HYPER-DISTRIBUTION 4
FACTR  EQUAL TO FACTOR = PHI * KRATIO * Z
FIFTY  EQUAL TO .5 -- USED TO ACCESS 50% POINT IN LIFE() GTRLIFE  FUNCTION WHICH CALCULATES THE CYCLES TO FAILURE AT A GIVEN STRESS I  CONTROLS DO LOOP FOR EACH LIFE CALCULATION IOUT  OUTPUT DUMP CONTROLER C  CONTROLS DO LOOP FOR EACH LIFE K  CONTROLS DO LOOP FOR EACH HYPER-PARAMETER SET KD  STRESS FACTOR TO ADJUST FOR USING 2-D ANALYSIS KRATIO  RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS KT  STRESS CONCENTRATION FACTOR USED IN ANALYSIS TO OBTAIN REFERENCE STRESSES L  CONTROLS DO LOOP FOR EACH REGION LAMB  THERMAL GRADIENT SENSITIVITY FACTOR LAMKD  SELECTED STRESS CONCENTRATION FACTOR Kd ACCURACY LAMKDA  Kd ACCURACY FACTOR LOWER BOUND LAMKDB  Kd ACCURACY FACTOR UPPER BOUND LAMKT  SELECTED STRESS CONCENTRATION FACTOR Kt ACCURACY LAMKTA  Kt ACCURACY FACTOR LOWER BOUND LAMKTB  Kt ACCURACY FACTOR UPPER BOUND LAMM  METAL TEMPERATURE SENSITIVITY FACTOR LIFE()  1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM -- SORTED VALUES OF THE LEFT-HAND TAIL LNZ  NORMAL(0,PSVAR) GENERATED RANDOM VARIATE M  CONTROLS SYMMETRY DO LOOP MAXLF  MAXIMUM NUMBER OF LIVES TO BE CALCULATED MAXDAT  MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED MAXLIF  MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, LIFE() ALPHA CALCULATION MAXM  MAXIMUM NUMBER OF M'S TO BE SORTED MAXREG  MAXIMUM NUMBER OF REGION ALLOWED MCOUNT  NUMBER OF M'S TO BE USED TO CALCULATE NORMAL MEDIAN S/N CURVE MEDM()  1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION C  POINT 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 SELECTED VALUES OF M FOR EACH REGION MPROC  Materials Process Variation -- CONTROLS MATERIALS PROCESS VARIATION -- 0 - NO VARIATION, 1 - VARIATION MU()  1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION MLIFE  NUMBER OF LIVES TO BE CALCULATED MBND()  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 GTRLIFE NF()  2-D ARRAY CONTAINING RMNF() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS C  SIZE OF OUTER LOOP NLIFE  SIZE OF INNER LOOP CONSUM  TOTAL NUMBER OF LIVES CALCULATED BY PFM CMED  CONTROLS MEDIAN CALCULATION FOR THE NORMAL DISTRIBUTION CASE -- 0 - NO MEDIAN CALCULATION; 1 - MEDIAN CALCULATION DESIRED CNPTS()  1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
SUBROUTINE PTRYR (RAND, RHO1, RHO2, THE1, THE2, X, Y)

COMMON IOUT

DOUBLE PRECISION RAND

REAL  FRAC, RHO1, RHO2, THE1, THE2, X, Y
INTEGER IOUT
CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
X = FRAC * (RHO2 - RH01) + RH01
CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
Y = FRAC * (THE2 - THE1) + THE1
IF (IOUT .EQ. 15) WRITE(8,*) 'RH01 =', RH01, ' RH01 =', RH01, 
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, RH0, THE1A, B, X)
COMMON IOUT
DOUBLE PRECISION RAND
REAL A, B, GAM, RH0, THE1A, W, X, Y1, Y2
INTEGER IOUT
IF (IOUT .EQ. 15) WRITE(8,*) 'RAND =', RAND, ' RH0 =', RH0,
& ' THE1A =', A, ' B =', B, ' X =', X
Y1 = GAM((RH0 * THE1A + 1.), RAND)
Y2 = GAM((1. - RH0) * THE1A + 1.), RAND)
W = Y1 / (Y1 + Y2)
IF (IOUT .EQ. 15) WRITE(8,*) 'Y1 =', Y1, ' Y2 =', Y2, ' W =', W

C TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION
X = W * (B - A) + A
IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X
RETURN
END

C***************************************************************************
C The random variates are generated using an "Acceptance/Rejection Method"
C Fishman, George S., "Sampling From the Gamma Distribution on a
C Computer," Communications of the ACM, Volume 19, Number 7, July 1976,
C***************************************************************************

REAL FUNCTION GAM (ALPHA, RAND)
SUBPROGRAM: RANDOM
COMMON IOUT
INTEGER IOUT
REAL A, ALPHA, ARG, U1, U2, V1, V2

DOUBLE PRECISION RAND

C

IF (IOUT .EQ. 15) WRITE(8,*') A =', A, ' ALPHA =', ALPHA

10 CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALOG(U1)
V2 = - ALOG(U2)
C
IF (IOUT .EQ. 15) WRITE(8,*') U1 =', U1, ' U2 =', U2, ' V1 =', V1, ' V2 =', V2
C
ARG = A * (V1 - ALOG(V1) - 1.)
IF (V2 .LT. ARG) GOTO 10
C
GAM = ALPHA * V1
C
IF (IOUT .EQ. 15) WRITE(8,*') GAMMA =', GAM

RETURN
END

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

C SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C PROGRAMMER: L. NEWLIN
C DATE: 20JUN90
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 IMPLICIT NONE
C INTEGER MAXLIF
C PARAMETER (MAXLIF = 10000)
C COMMON IOUT
C INTEGER I, IOUT, NLIFET, NUM, PLACE
C REAL LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

LIST OF VARIABLES
I
| CONTROLS DO LOOP FOR INSERTION |
IOUT | OUTPUT DUMP CONTROLLER |
LIFE() | 1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE |
PPM TO BE SORTED |
MAXLIF | MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, Theta, Alpha, |
CALCULATION |
NEWLIF | LIFE VALUE TO BE INSERTED INTO LIFE() |
NLIFET | TOTAL NUMBER OF LIVES CALCULATED BY PPM |
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

7 - 268
C FIND POSITION IN LIFE() FOR NEWLIF

IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

DO 100 I = 1, NUM
   IF (NEWLIF .LT. LIFE(I)) THEN
      PLACE = I
      GOTO 110
   ENDIF
100 CONTINUE
110 CONTINUE

C STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

DO 200 I = (PLACE + 1), NUM
   TEMP(I) = LIFE(I-1)
200 CONTINUE

C INSERT NEWLIF

LIFE(PLACE) = NEWLIF

C SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

DO 300 I = (PLACE + 1), NUM
   LIFE(I) = TEMP(I)
300 CONTINUE

C IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN

400 CONTINUE

RETURN
END

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

C SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
FOR THE STRESS FORMULATION
PROGRAMMER: L. NEWLIN
DATE: 13JUL89
FORMAT/COMMENTS: 12AUG91
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 SPECFD AND RELATD; VARY, MPROC
OUTPUTS: RANGEM, MU, SIG, NF, REFNP, SZERO, ZROREG, NUMREG, &
NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
SUBPROGRAMS: INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDRREG, &
CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR
FILES: 5:RELATD-OLD; 6:RELATD-NEW

C IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT

INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT, &
NP(0:MAXSET, MAXREG), NPFR(MAXREG), NPTS(0:MAXSET),
LIST OF VARIABLES

BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
FTUZ, FTUZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
KAMT, LAMN, LNFN(MAXDAT, 0:MAXSET, MAXREG),
LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
MCHAT(2, MAXREG), MCHAT(MAXREG), MO(MAXREG), MZERO(2, MAXREG),
MZERO(MAXREG), NBNDF(MAXREG, 0:MAXREG), NF(MAXDAT, MAXREG),
PVAR, RANGEM(2..MAXREG), RAWSTR(MAXDAT, 0:MAXSET),
RANF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT),
SUHAT2(MAXREG), SWHAT2(MAXREG), SX(MAXREG),
SX2(MAXREG), SY2(MAXREG), ZERO

NSETS, NUMREG, REFN( MAXREG ), VARY, ZZERO

REAL
BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
FTUZ, FTUZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
KAMT, LAMN, LNFN(MAXDAT, 0:MAXSET, MAXREG),
LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
MCHAT(2, MAXREG), MCHAT(MAXREG), MO(MAXREG), MZERO(2, MAXREG),
MZERO(MAXREG), NBNDF(MAXREG, 0:MAXREG), NF(MAXDAT, MAXREG),
PVAR, RANGEM(2..MAXREG), RAWSTR(MAXDAT, 0:MAXSET),
RANF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT),
SUHAT2(MAXREG), SWHAT2(MAXREG), SX(MAXREG),
SX2(MAXREG), SY2(MAXREG), ZERO

VALUE
EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
VALUE OF WEIBULL PARAMETER, BETA, CHARACTERIZING THE S/N
DATA SET
EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
COEFFICIENT OF VARIATION, CO
1-D ARRAY CONTAINING SKY(L)/SK2(L) FOR EACH REGION
AND SIG() CALCULATION
ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
OUTPUT DUMP CONTROLLER
2-D ARRAY CONTAINING IO, THE 95% CONFIDENCE INTERVALS ON C
FOR EACH REGION
2-D ARRAY CONTAINING IO, THE 95% CONFIDENCE INTERVALS ON M
FOR EACH REGION
RATIO OF K*K, CONSTANT OVER REGIONS AND COMPONENTS
CONTROL DO LOOP FOR EACH REGION
CONSTANT OVER REGIONS AND COMPONENTS
3-D ARRAY CONTAINING LN(RANF()) (ALSO INDEXED FOR REGION
3-D ARRAY CONTAINING LN(RANF()) (ALSO INDEXED FOR REGION
MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXIMUM NUMBER OF REGIONS ALLOWED
2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
--- MC(L,1) IS THE LOWER BOUND AND MC(L,2) 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(L,1) = -DD, THE ESTIMATE FOR M AND
MCHAT(L,2) = SUHAT, THE ESTIMATE FOR C
1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MC() FOR EACH REGION
1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
MEAN FOR EACH REGION
1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
MZERO() FOR EACH REGION
"MATERIALS PROCESS VARIATION -- CONTROLS MATERIALS PROCESS
VARIATION --- 0 = NO VARIATION, 1 = VARIATION
1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION MEAN FOR EACH REGION
2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
EACH REGION --- MZERO(L,1) IS THE LOWER BOUND AND MZERO(L,2)
IS THE UPPER BOUND
1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE MAXREG
REGIONS OF INTEREST
2-D ARRAY CONTAINING RAWF() (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
SET IN EACH REGION
2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
SET 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
NUMREG  NUMBER OF REGIONS OF INTEREST
PVAR MATERIALS PROCESS VARIATION
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
         FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
         RANGEM(2,L) IS THE UPPER BOUND
RATSTR() 2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR
         STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
RAWNF()  2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
         DATA SETS
RAWSTR()  2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OR TOTAL STRAIN
         DATA (%) FOR ALL S/N DATA SETS
REFNP()   1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
         (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
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)
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
ZROREG   ZEROREGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
         BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

OPEN(5, FILE = 'RELATO', STATUS = 'OLD')
OPEN(6, FILE = 'RELATO', STATUS = 'NEW')
C RELATO 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, NODAT,
            &                                    NSETS, NNODE, CZERO, MPNT, MZERO, NUT2, PVT2, DELTA, MO,
            &                                    SIGMA, KRATIO, LAMN)
C CALCULATE RESIDUAL VARIANCES
            CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SXY, SY2, DD,
            &                                    SWHAT2, SUHAT2, NPFR)
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, NPFR, 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 RANGES FOR REGIONS WITHOUT DATA
CALL ADDREG (RANGEM, MCHAT, MNODAT, 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
WRITE(7,910)
DO 50 L = 1, NUMREG
WRITE(7,915) L, MCHAT(2,L), MCHAT(1,L)
50 CONTINUE
IF (CZERO .GT. 0.0) THEN
WRITE(7,960)
DO 150 L = 1, NUMREG
IF (MCPNT(L) .EQ. 1) THEN
WRITE(7,965) L, MC(1,L)
ELSEIF (MCPNT(L) .EQ. 2) THEN
WRITE(7,970) L, MC(1,L), MC(2,L)
ENDIF
150 CONTINUE
ENDIF
WRITE(7,990)
WRITE(7,930)
DO 100 L = 1, NUMREG
WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
100 CONTINUE
WRITE(7,950)

C CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO
CALL MEDIAN (NUMREG, RANGEM, MEDM)

C CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
CALL EXPCTD (1, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG, NBND, BIGHT, 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)
200 CONTINUE
ENDIF
ELSE
C NORMAL VARIATION IS DESIRED
C CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION
CALL MUSIG (NUMREG, SX2, DD, SHAT2, Sihat2, 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 = i, NUMREG
WRITE(7,980) L, MCHAT(I,L)
350 CONTINUE
IF (CZERO .GT. 0.0) THEN
WRITE(7,960) L, NUMREG
DO 360 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
360 CONTINUE
ENDIF
WRITE(7,920)
WRITE(7,930)
DO 370 L = 1, NUMREG
WRITE(7,940) L, RANGEM(1,L), RANGEM(2,L)
370 CONTINUE
WRITE(7,950)
WRITE(7,985) L, NUMREG
DO 380 L = 1, NUMREG
WRITE(7,990) L, MU(L), SIG(L)
380 CONTINUE
ENDIF
C PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS
IF (MPROC .EQ. 1) THEN
WRITE(7,995) PVAR
ENDIF
C FORMAT STATEMENTS

7 - 273
C SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED

C PROGRAMMER: L. NEWLIN
C DATE: 5OCT87

SUBROUTINE TRMNAT
WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'
STOP
END

7 - 274
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)

INPUTS: ---
OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
& NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

COMMON IOUT
INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
& NPTS(0:MAXSET), REFNP(MAXREG)
REAL DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
& LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
& MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
& RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
& RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
& STR(MAXDAT, MAXREG)

LIST OF VARIABLES

DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
I OUT CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
J OUT OUTPUT DUMP CONTROLLER
K CONTROLS DO LOOP FOR EACH DATA SET
L CONTROLS DO LOOP FOR EACH POINT IN A REGION
LNNF() 3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN 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
NP() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NPTS() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
RATSTR() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
RAWNF() 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 RAW STRESS DATA (PSI) OF TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
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

7 - 275
S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

C

DO 100 J = 0, MAXSET
NP(N(J), L) = 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
LNNF(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(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

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

C SUBROUTINE RCE "READS" THE DATA FROM SPECFD AND RELATD; "CONVERTS" THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO SPECFD AND RELATD. RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS SPECIFIED BY USER
C
C PROGRAMMER: L. NEWLIN
C DATE: 21JUN88
C
C*****SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, & LNNF, REFNP, STR, NF, SZERO, ZROREG, & NUMREG, NNODAT, NSETS, NBD, CZERO, MPNT, MZERO, & FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN)*****
C
C INPUTS: VARY, MPROC
C OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, NNODAT, NSETS, NBD, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2, KRATIO, LAMN

7 - 276
IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET

PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

COMMON IOUT

INTEGER COUNT, I, IOUT, J, K, L, M, MPNT, MZERO, N, NUM, NUMREG,
& NODAT, NP(0:MAXSET, MAXREG), NPTS(0:MAXSET), NSETS,
& NUM, NUNREG, REPNP(MAXREG), REG, VARY, ZROREG

REAL CZERO, DELTA(MAXREG), FTU, FTUZ, FTY, FTYZ,
& KJ, LAMN, LNF(MAXDAT, 0:MAXSET, MAXREG), LNSR(MAXDAT, 0:MAXSET, MAXREG),
& MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
& RAT, RATSTR(MAXDAT, 0:MAXSET, MAXREG), RAWNP(MAXDAT, 0:MAXSET),
& RATIO, RATIOSTR(MAXDAT, 0:MAXSET), SIGNAL(MAXREG),
& STR(MAXDAT, MAXREG), SZERO

CHARACTER*40 DESCRP (0:MAXSET)

LIST OF VARIABLES

COUNT INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO, CONVERSION, AND BREAK UP
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, CO
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
DESCRP() 1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
FTU ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
FTUZ ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTY YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTYZ YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
I CONTROL DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
L CONTROLS DO LOOP FOR EACH REGION
LAMN LAMBDA-N -- RATIO OF VAR (LN N given S) / (m**2 C-N-2), CONSTANT OVER ALL REGIONS AND COMPONENTS
LNNF() 3-D ARRAY CONTAINING LN(RAWNP()), ALSO INDEXED FOR REGION
LNSR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
M CONTROLS DO LOOP FOR EACH DATA DIVISION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MPROC Materials Process variation -- CONTROLS MATERIALS PROCESS VARIATION -- 0 - NO VARIATION; 1 - VARIATION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NDIV NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO, REGIONS PAIRS DURING INPUT
NF() 2-D ARRAY CONTAINING RAWNP() (CYCLES TO FAILURE) FOR EACH SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NODAT 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
RANF() 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
REPNP() 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 %)
ZERO 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)
IF (IOUT .EQ. 11) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)
WRITE(3,905)
IF (IOUT .EQ. 11) 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 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), RANF(I,0)
 110 CONTINUE

C CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
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, PTY)

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)
130  CONTINUE

C BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2, EXPECTD, AND PAREST

K = NP(0,REG)
DO 140 I = (COUNT + 1), (COUNT + NUM)
    K = K + 1
    LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
    LNRF(K,0,REG) = ALOG(RAWRF(I,0))
    STR(K,REG) = RATSTR(I,0)
    NRF(K,REG) = RAWRF(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 TRNAT
ENDIF

DO 150 L = ZROREG, (NUMREG + NNODAT)
  READ(1,*) NBND(L)
150 CONTINUE

READ(1,*) CZERO

DO 160 L = 1, (NUMREG + NNODAT)
  READ(1,*) MPNT(L), MZERO(1,L), MZERO(2,L)
160 CONTINUE

WRITE(3,913)
  IF (ZROREG .EQ. 0) WRITE(3,914) 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. I0) WRITE(8,920) NBND(L)
170 CONTINUE

WRITE(3,925) CZERO
  IF (IOUT .EQ. 10) WRITE(8,925) CZERO

DO 180 L = 1, (NUMREG + NNODAT)
  WRITE(3,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
  IF (IOUT .EQ. 10) THEN
    WRITE(8,930) L, MPNT(L), MZERO(1,L), MZERO(2,L)
    IF ((VARY .EQ. 3) .AND. MPNT(L .EQ. 0)) THEN
      WRITE(8,*) 'RANGE ON M'
      CALL TRNAT
    ENDIF
  ENDIF
180 CONTINUE

IF (VARY .EQ. 3) THEN
  IF (IOUT .EQ. 10) WRITE(8,945)
  READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
  WRITE(3,945)
  IF (IOUT .EQ. 10) WRITE(8,945)
  DO 190 L = 1, (NUMREG + NNODAT)
    READ(1,*) DELTA(L), MO(L), SIGMA2(L)
    WRITE(3,950) L, DELTA(L), MO(L), SIGMA2(L)
    IF (IOUT .EQ. 10) THEN
      WRITE(8,950) L, DELTA(L), MO(L), SIGMA2(L)
      IF (((DELTA(L) .LT. 0.0) .OR. (MO(L) .LE. 0.0))) THEN
        WRITE(8,*) 'ERROR: BAD VALUE FOR DELTA OR VALUE OF MO ',
        'INCONSISTENT WITH DELTA IN REGION ', L
      ENDIF
    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 RELATO
C AND THEN ECHO TO RELATO

READ(5,*) NSETS
IF (NSETS .GT. MAXSET) THEN
   WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF RELATED DATA SETS'
   CALL TRMNAT
ENDIF
WRITE(6,935) NSETS
DO 200 J = 1, NSETS
   COUNT = 0
   IF (IOUT . EQ. 10) WRITE(8,*) 'J = ', J, ' NSETS = ', NSETS
   READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)
   IF (NPTS(J) . GT. MAXDAT) THEN
      WRITE(8,*) 'ERROR: OVER LIMIT ON NUMBER OF POINTS IN ', 'SET ', J
      CALL TRMNAT
   ENDIF
   WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
   IF (IOUT . EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)
WRITE(6,905)
   IF (IOUT . EQ. 10) WRITE(8,905)
DO 300 M = 1, NDIV
   READ(5,*) NUM, RATIO, REG
   IF (ABS(RATIO) . GT. 1.0) THEN
      WRITE(8,*) 'ERROR: INVALID VALUE OF RATIO: ', RATIO
      CALL TRMNAT
   ENDIF
   IF (REG . GT. MAXREG) THEN
      WRITE(8,*) 'ERROR: OVER REGION LIMIT IN RELATED MATERIAL ', J
      CALL TRMNAT
   ENDIF
   IF (IOUT . EQ. 10) THEN
      WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
      WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
   ENDIF
   DO 310 I = (COUNT + 1), (COUNT + NUM)
      READ(5,*) RAWSTR(I,J), RAWNF(I,J)
   CONTINUE
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)
      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 ',
    '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

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',5X,'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',
&    'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
&    (CYCLES): '/)

920 FORMAT(10X,E9.3)

925 FORMAT(/,2X,'EXOGENOUS INFORMATION',///,2X,'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
&    ///,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
&    ///,2X,'REGION',3X,'# OF POINTS',5X,'LOWER BOUND',
&    5X,'UPPER BOUND',//)

930 FORMAT(6X,I1,11X,I1,12X,F7.4,9X,F7.4)

935 FORMAT(20X,'NUMBER OF DATA SETS:',2X,I2,///,17X,
&    'NOTE: ALL Kt ASSUMED TO BE 1.0',///,23X,
&    'TRANSFORMED DATA')
C***SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)
C
C***DESCRIPTION:
C
C    THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C    STRESS RATIO, R, IS NOT -1.0
C
C***DATE:  CODE:  6OCT87  COMMENTS:  13JUL89
C
C***VERSION:  MATER V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C
C***PARAMETERS:
C    J, NUM1, NUM2, STR, RSTR, R, FTU, FTY
C
C***INPUTS:  J, NUM1, NUM2, STR, R, FTU, FTY
C
C***OUTPUTS:  RSTR
C
C***LIST OF VARIABLES
C    FTU    YIELD STRENGTH OF MATERIAL (PSI)
C    FTY    ULTIMATE 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    NUM1   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 * (I - R)/2 , TO BE COMPARED WITH FTY
C
C Kt IS ASSUMED TO BE ONE
C
DO 100 I = NUM1, NUM2
   TEST = STR(I,J) * (1.0 - R)/2.0
   IF (IOUT.EQ.10) WRITE(8,*) 'I=',I,' J=',J,' TEST=',TEST
IF (TEST .GE. FTY) THEN
  RSTR(I,J) = TEST
  IF (IOUT.EQ.10) WRITE(8,*') '1:RSTR() = ', RSTR(I,J)
ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
  RSTR(I,J) = TEST/(1.0 - ((FTY - TEST)/FTU))
  IF (IOUT.EQ.10) WRITE(8,*') '2:RSTR() = ', RSTR(I,J)
ELSE
  RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
                      /(2.0 * FTU)))
  IF (IOUT.EQ.10) WRITE(8,*') '3:RSTR() = ', RSTR(I,J)
END IF
100 CONTINUE
RETURN
END

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

C SUBROUTINE SW2SU2 CALCULATES SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C PROGRAMMER: L. NEWLIN
C DATE: 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
C DESCRIPTION:
C SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SY2, DD, SWHAT2, SUHAT2, NPPR)
C INPUTS: NUMREG, NSETS, NP, LNSTR, LNNF
C OUTPUTS: SX2, SY2, SY2, DD, SWHAT2, SUHAT2, NPPR
C IMPLICIT NONE
C INTEGER MAXDAT, MAXREG, MAXSET
C PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
C COMMON IOUT
C INTEGER IOUT, J, K, L, NP(0:MAXSET, MAXREG), NPPR(MAXREG),
C & NSETS, NUMREG
C REAL BB(MAXREG), DD(MAXREG), DIFFX(MAXDAT, 0:MAXSET),
C & DIFFY(MAXDAT, 0:MAXSET), LNNF(MAXDATA, 0:MAXSET, MAXREG),
C & MEANX(0:MAXSET), MEANX(0:MAXSET), MEANX(0:MAXSET),
C & SY2(MAXREG), SY2(MAXREG), SY2(MAXREG)
C LIST OF VARIABLES
C BB() 1-D ARRAY CONTAINING SX2(L)/SY2(L) FOR EACH REGION
C DD() 1-D ARRAY CONTAINING SX2(L)/SX2(L) FOR EACH REGION
C DIFFX() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNSTR(K,J,L)
C AND MEANX(J) FOR EACH POINT IN EACH DATA SET FOR REGION L
C DIFFY() 2-D ARRAY CONTAINING THE DIFFERENCE BETWEEN LNNF(K,J,L)
C IOUT 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
LNNF() 3-D ARRAY CONTAINING LN(RATSTR()); ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RSTR()); ALSO INDEXED FOR REGION
MAXREG MAXIMUM NUMBER OF POINTS PER S/N DATA SET (PER REGION) ALLOWED
MAXSET MAXIMUM NUMBER OF REGIONS ALLOWED
MEANX() 1-D ARRAY CONTAINING SAMPLE X MEAN FOR POINTS FROM REGION
L AND DATA SET J (X = Ln S)
MEANY() 1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
L AND DATA SET J (Y = Ln N)
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
SET IN EACH REGION
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1)-1)) OVER
ALL DATA SETS IN A REGION (Number of Points Per Region)
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUMREG NUMBER OF REGIONS OF INTEREST
SUHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(SX2 = Ln S)
SXY() 1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR
EACH REGION (X = Ln S, Y = Ln N)
SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
(Y = Ln N)

C INITIALIZE ARRAYS

DO 50 L = 1, MAXREG
SY2(L) = 0.0
SX2(L) = 0.0
SXY(L) = 0.0
SUHAT2(L) = 0.0
SWHAT2(L) = 0.0
BB(L) = 0.0
DD(L) = 0.0
NPPR(L) = 0
50 CONTINUE

DO 60 J = 0, MAXSET
DO 70 K = 1, MAXDAT
DIFFY(K,J) = 0.0
DIFFX(K,J) = 0.0
70 CONTINUE
MEANY(J) = 0.0
MEANX(J) = 0.0
60 CONTINUE

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

DO 100 L = 1, NUMREG

DO 200 J = 0, NSETS
C FIRST CALCULATE SAMPLE X AND Y MEANS FOR DATA SET J IN REGION L
MEANX(J) = 0.0
MEANY(J) = 0.0
IF (IOUT .EQ. 10) WRITE(8,E10) WRITE(8,E10) 'MEANX(J)', MEANX(J),
& 'MEANY(J)' MEANY(J)

DO 250 K = 1, NP(J,L)
MEANX(J) = MEANX(J) + LNNF(K,J,L)
MEANX(J) = MEANX(J) + LNSTR(K,J,L)
IF (IOUT .EQ. 10) WRITE(8,E10) 'MEANX(J)' MEANX(J),
& 'LNNF =', LNNF(K,J,L),
& 'LNSTR =', LNSTR(K,J,L)
250 CONTINUE

MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
IF (IOUT .EQ. 10) WRITE(8,E10) 'MEANX(J)' MEANX(J),
& 'MEANY(J)' MEANY(J)
7 - 285
NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY, OF X AND Y FOR EACH REGION BY SUMMING OVER EACH DATA SET IN REGION L

DO 300 K = 1, NP(J,L)
  DIFFY(K,J) = LNYF(K,J,L) - MEANY(J)
  DIFFX(K,J) = LNXF(K,J,L) - MEANX(J)
  SY2(L) = SY2(L) + DIFFY(K,J) ** 2
  SX2(L) = SX2(L) + DIFFX(K,J) ** 2
  SKX(L) = SKX(L) + DIFFX(K,J) * DIFFY(K,J)
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'K = ', K, ' DIFFX(K,J) = ', DIFFX(K,J),
  ' DIFFY(K,J) = ', DIFFY(K,J),
  ' SY2(L) = ', SY2(L), ' SX2(L) = ', SX2(L),
  ' SKY(L) = ', SKY(L)
ENDIF
CONTINUE

NPFR(L) = NPFR(L) + NP(J,L) - 1
IF (IOUT .EQ. 10) WRITE(8,*), 'NPFR(L) = ', NPFR(L)
CONTINUE

IF (SKY(L) .GE. 0.0) THEN
  LIFE WILL INCREASE WITH INCREASING STRESS -- INVALID FOR OUR MODEL
  WRITE(8,*), 'ERROR: SKY > 0 IN REGION', L
  CALL TRNAT
ENDIF
NPFR(L) = NPFR(L) - 1
IF (NPFR(L) .LE. 0) THEN
  WRITE(8,*), 'ERROR: TOO FEW POINTS FOR REGRESSION IN ',
  'REGION ', L
  CALL TRNAT
ENDIF

SY2(L) = SY2(L) / FLOAT(NPFR(L))
SX2(L) = SX2(L) / FLOAT(NPFR(L))
SKY(L) = SKY(L) / FLOAT(NPFR(L))

NOW CALCULATE THE RESIDUAL VARIANCES, SHAT2, SUHAT2, FOR EACH REGION FROM THE Y ON X AND X ON Y REGRESSIONS

DD(L) = SXY(L) / SX2(L)
BB(L) = SKY(L) / SY2(L)
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'NPFR(L) = ', NPFR(L), ' SY2(L) = ', SY2(L),
  ' SX2(L) = ', SX2(L),
  ' SKY(L) = ', SKY(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)
  CONTINUE

DO 500 K = 1, NP(J,L)
  SHAT2(L) = SHAT2(L)
  ++ (DIFFX(K,J) - DD(L) * DIFFX(K,J)) ** 2
  SUHAT2(L) = SUHAT2(L)
  ++ (DIFFX(K,J) - BB(L) * DIFFX(K,J)) ** 2
  IF (IOUT .EQ. 10) WRITE(8,*), 'K = ', K, ' SHAT2(L) = ',
  ' SUHAT2(L) = ', SUHAT2(L)
CONTINUE

400 CONTINUE
500 CONTINUE

SHAT2(L) = SHAT2(L) / FLOAT(NPFR(L))
SUHAT2(L) = SUHAT2(L) / FLOAT(NPFR(L))
IF (IOUT .EQ. 10) WRITE(8,*), 'NPFR(L) = ', NPFR(L),
  ' SHAT2(L) = ', SHAT2(L), ' SUHAT2(L) = ', SUHAT2(L)
100 CONTINUE
**SUBROUTINE INTRVL** calculates the 95% confidence interval, \( I_0 \), on \( C \); and the 95% confidence interval, \( J_0 \), on \( M \).

**PROGRAMMER:** L. NEWLIN

**DATE:** CODE: 5OCT87

**COMMENTS:** 15SEP89

**VERSION:** MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

**SUBROUTINE INTRVL** (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO, MCHAT)

**INPUTS:** NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR

**OUTPUTS:** IZERO, JZERO, MCHAT

**SUBPROGRAMS:** TRMNAT

**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, 75) /**

& 0.000982069, 0.506356, 0.215795, 0.484419, 0.831211, 1.237347,

& 1.68987, 2.17973, 2.70039, 3.24624, 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, 17.5318,

& 18.28, 19.04, 19.80, 20.56, 21.33,

& 22.10, 22.87, 23.65, 24.4331, 25.21,

& 25.99, 26.78, 27.57, 28.36, 29.15,

& 29.95, 30.75, 31.57, 32.354, 33.15,

& 33.96, 34.77, 35.58, 36.39, 37.21,

& 38.02, 38.84, 39.66, 40.4817, 41.30,

& 42.12, 42.95, 43.77, 44.60, 45.43,

& 46.26, 47.09, 47.92, 48.7576, 49.59,

& 50.42, 51.26, 52.10, 52.94, 53.8017,

& (CHI025(I), I = 76, 150) /**

& 54.62, 55.46, 56.30, 57.1411, 57.9871,

& 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, 75.0586,

& 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, 89.83, 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, 105.62,

& 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, 118.8617,

& (CHI975(I), I = 1, 75) /**

& 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,

& 17.5318, 18.28, 19.04, 19.80, 20.56, 21.33,

& 22.10, 22.87, 23.65, 24.4331, 25.21,

& 25.99, 26.78, 27.57, 28.36, 29.15,

& 29.95, 30.75, 31.57, 32.354, 33.15,

& 33.96, 34.77, 35.58, 36.39, 37.21,

& 38.02, 38.84, 39.66, 40.4817, 41.30,

& 42.12, 42.95, 43.77, 44.60, 45.43,

& 46.26, 47.09, 47.92, 48.7576, 49.59,

& 50.42, 51.26, 52.10, 52.94, 53.8017,

& (CHI975(I), I = 76, 150) /**

& 54.62, 55.46, 56.30, 57.1411, 57.9871,

& 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, 75.0586,

& 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, 89.83, 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, 105.62,

& 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, 118.8617,
VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:

1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719


-- CALCULATED USING CUBE RULE APPROXIMATION

<table>
<thead>
<tr>
<th>DATA (CHI975(I))</th>
<th>I = 76, 150 /</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.00</td>
<td>103.10</td>
</tr>
<tr>
<td>107.79</td>
<td>106.94</td>
</tr>
<tr>
<td>113.54</td>
<td>115.84</td>
</tr>
<tr>
<td>119.28</td>
<td>121.57</td>
</tr>
<tr>
<td>125.00</td>
<td>127.29</td>
</tr>
<tr>
<td>130.70</td>
<td>132.98</td>
</tr>
<tr>
<td>136.38</td>
<td>138.65</td>
</tr>
<tr>
<td>142.05</td>
<td>144.31</td>
</tr>
<tr>
<td>147.71</td>
<td>149.66</td>
</tr>
<tr>
<td>153.34</td>
<td>155.59</td>
</tr>
<tr>
<td>158.97</td>
<td>161.21</td>
</tr>
<tr>
<td>164.58</td>
<td>166.82</td>
</tr>
<tr>
<td>170.17</td>
<td>172.41</td>
</tr>
<tr>
<td>175.77</td>
<td>178.00</td>
</tr>
<tr>
<td>181.35</td>
<td>183.58</td>
</tr>
</tbody>
</table>

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(L)/SX2(L) FOR EACH REGION

I OUT OUTPUT DUMP CONTROLLER

IZER() 2-D ARRAY CONTAINING I0, 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) = SUHAT2(L) = SUHAT, THE ESTIMATE FOR C

NPFR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()(-1))(-1)) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)

NUM NUMBER OF REGIONS OF INTEREST

NUMBER 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 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)

SWHAT EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS

VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:

1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719


-- CALCULATED USING CUBE RULE APPROXIMATION

<table>
<thead>
<tr>
<th>DATA (CHI975(I))</th>
<th>I = 76, 150 /</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.00</td>
<td>103.10</td>
</tr>
<tr>
<td>107.79</td>
<td>106.94</td>
</tr>
<tr>
<td>113.54</td>
<td>115.84</td>
</tr>
<tr>
<td>119.28</td>
<td>121.57</td>
</tr>
<tr>
<td>125.00</td>
<td>127.29</td>
</tr>
<tr>
<td>130.70</td>
<td>132.98</td>
</tr>
<tr>
<td>136.38</td>
<td>138.65</td>
</tr>
<tr>
<td>142.05</td>
<td>144.31</td>
</tr>
<tr>
<td>147.71</td>
<td>149.66</td>
</tr>
<tr>
<td>153.34</td>
<td>155.59</td>
</tr>
<tr>
<td>158.97</td>
<td>161.21</td>
</tr>
<tr>
<td>164.58</td>
<td>166.82</td>
</tr>
<tr>
<td>170.17</td>
<td>172.41</td>
</tr>
<tr>
<td>175.77</td>
<td>178.00</td>
</tr>
<tr>
<td>181.35</td>
<td>183.58</td>
</tr>
</tbody>
</table>
C SWHAT2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS

C SX 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(X = Ln S)

C SX2() VALUE OF T025() USED IN CALCULATIONS

C T TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION

C T025() 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(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

C CONFIDENCE INTERVALS

DO 100 L = 1, NUMREG
NUM = NPPR(L)
IF (NUM .LT. 31) THEN
T = T025(NUM)
ELSE
T = T025(NUM)
ENDIF
SWHAT = SWHAT2(L) ** 0.5
SUHAT = SUHAT2(L) ** 0.5
SX = (NUM * SX2(L)) ** 0.5

C CALCULATE ESTIMATED VALUES OF M AND C

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

C CALCULATE CONFIDENCE INTERVALS

IZERO(1,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI975(NUM)) ** 0.5
IZERO(2,L) = MCHAT(2,L) * (FLOAT(NUM) / CHI025(NUM)) ** 0.5
JZERO(1,L) = MCHAT(1,L) - ARG
JZERO(2,L) = MCHAT(1,L) + ARG

C WRITE OUT INTERVALS

IF (IOUT .EQ. 10) 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(2,L) =', IZERO(2,L)
& WRITE(8,*), 'JZERO(1,L) =', ' JZERO(2,L) =', JZERO(2,L)
& WRITE(8,*), 'MCHAT(1,L) =', ' MCHAT(2,L) =', MCHAT(2,L)
7 - 289
SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SKY, SY2, MCPNT, MC)

C INPUTS: NUMREG, CZERO, SX2, SKY, SY2
C
C OUTPUTS: MCPNT, MC
C
C IMPLICIT NONE

INTEGER MAXREG
PARAMETER (MAXREG = 3)

COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), NUMREG

REAL ARGI, ARG2, CZERO, CZERO2,
    MC(2, MAXREG), SX2(MAXREG),
    SKY(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
       (X = Ln S)
SKY()  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
      WRITE(8,*) 'ERROR: CO TOO LOW'
      CALL TRASH
      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
    ENDF
  100 CONTINUE

IF (IOUT .EQ. 10) THEN
  DO 200 L = 1, NUMREG
  WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
  WRITE(8,*) 'ARG1 = ', ARG1, ' ARG1, ' ARG2 = ', ARG2
  WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
  200 CONTINUE
ENDIF

RETURN
END
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: MATCHRM V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)

C INPUTS: NSETS, NP, NUMREG, LAMN, MCHAT
C OUTPUTS: PVAR

C IMPLICIT NONE
C
INTEGER MAXREG, MAXSET
PARAMETER (MAXREG = 3, MAXSET = 5)
C
COMMON IOUT
INTEGER IOUT, J, L, NP(0:MAXSET, MAXREG), NSETS, NUM(MAXREG),
& NUMREG, TOTAL
REAL LAMN, MCHAT(2, MAXREG), PSIG2(MAXREG), PVAR, SUM

C
C
C IOUT
C J
C L
C LA_N
C
C MAXREG
C MAXSET
C MCHAT ( )
C
C
C NP()
C
C NSETS
C
_(_)
C
C
C _IG2()
C
C
C AR
C
C SUM
C
C TOTAL

LIST OF VARIABLES

JOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
L CONTROLS DO LOOP FOR EACH REGION
LAMN LAMBDA-N -- RATIO OF VAR (Lj 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) = SUBAT, THE ESTIMATE FOR C
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
SET IN EACH REGION
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUM() EQUAL TO NJ-1 FOR EACH REGION WHERE NJ IS THE SUM OF THE
 NUMBER OF POINTS IN EACH DATA SET
NUMREG NUMBER OF REGIONS OF INTEREST
PSIG2() 1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
 VARIATION IN EACH REGION
PVAR THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
 CURVE WARRANTED BY THE AVAILABLE INFORMATION
SUM WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
 AVERAGE
TOTAL SUM OF NUM() OVER ALL REGIONS

C

INITIALIZE VARIABLES

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)
CONTINUE
NUM(L) = NUM(L) - 1
TOTAL = TOTAL + NUM(L)
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 = i, 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 FNDREG 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           V8.4, V8.5
C SUBROUTINE FNDREG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
C                     & MCHAT, RANGEM)
C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C OUTPUTS: RANGEM
C SUBPROGRAMS: TRMNAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
      & MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

LIST OF VARIABLES
IOUT OUTPUT DUMP CONTROLLER
JZERO() 2-D ARRAY CONTAINING JO, THE 95% CONFIDENCE INTERVALS ON M
         FOR EACH REGION
L  CONTROLS DO LOOP FOR EACH REGION
LOWER LOWER BOUND OF INTERSECTION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MC() 2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
      REGION CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA
      -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
      BOUND
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
         FOR EACH REGION -- MCHAT(1,L) = DD(L), THE ESTIMATE
         FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C

7 - 293
c MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION
MNT() 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,*) 'MNT = ', MNT(L), ' MCPNT = ', MCPNT(L)
ENDIF

IF ((MCPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C THERE IS NO EXOGENOUS INFORMATION
ASSUME RANGE TO BE JZERO:
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 ((MCPNT(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 JZERO ACCORDINGLY
LOWER = AMAX1(JZERO(1,L), MC(1,L))
UPPER = JZERO(2,L)
IF (UPPER .LT. LOWER) THEN
WRITE(8,*) 'ERROR: NO INTERSECTION BETWEEN JZERO AND MC'
CALL TRNAT
ELSE
RANGEM(1,L) = LOWER
RANGEM(2,L) = UPPER
ENDIF
ELSEIF ((MCPNT(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 JZERO ACCORDINGLY
LOWER = AMAX1(JZERO(1,L), MC(1,L))

ENDIF
ELSEIF ((MCPNT(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 JZERO ACCORDINGLY
LOWER = AMAX1(JZERO(1,L), MC(1,L))

ENDIF

7 - 294
UPPER = AMIN1(JZERO(2,L), MC(2,L))
IF (UPPER .LT. LOWER) THEN
  WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN Jo AND MC'
  CALL TRMNAT
ELSE
  RANGEM(1,L) = LOWER
  RANGEM(2,L) = UPPER
ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  & 'JZERO(2,L) = ', JZERO(2,L),
  & 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
  WRITE(8,*), 'LOWER = ', LOWER, ' UPPER = ', UPPER
  WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
  & 'RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

ELSEIF (MPNT(L) .EQ. 1) THEN
  THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER INFORMATION:
  ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
  RANGEM(1,L) = MZERO(1,L)
  RANGEM(2,L) = 0.0
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L),
  & 'RANGEM(1,L) = ', RANGEM(1,L),
  & 'RANGEM(2,L) = ', RANGEM(2,L)
ENDIF
ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN
  THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT
  USE INTERSECTION BETWEEN Jo AND Mo
  LOWER = AMAX1(JZERO(1,L), MZERO(1,L))
  UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
IF (UPPER .LT. LOWER) THEN
  WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN Jo AND Mo'
  CALL TRMNAT
ELSE
  RANGEM(1,L) = LOWER
  RANGEM(2,L) = UPPER
ENDIF

IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  & 'JZERO(2,L) = ', JZERO(2,L),
  & 'MZERO(1,L) = ', MZERO(1,L),
  & 'MZERO(2,L) = ', MZERO(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. (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
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  & 'JZERO(2,L) = ', JZERO(2,L),
  & 'MZERO(1,L) = ', MZERO(1,L),
  & 'MZERO(2,L) = ', MZERO(2,L),
  & 'MC(1,L) = ', MC(1,L),
  & 'MC(2,L) = ', MC(2,L),
  & '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
  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 TRNSNT
  ELSE
    RANGEM(1,L) = LOWER
    RANGEM(2,L) = UPPER
  ENDIF
ENDIF
IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'JZERO(1,L) = ', JZERO(1,L),
  & 'JZERO(2,L) = ', JZERO(2,L),
  & 'MZERO(1,L) = ', MZERO(1,L),
  & 'MZERO(2,L) = ', MZERO(2,L),
  & 'MC(1,L) = ', MC(1,L),
  & 'MC(2,L) = ', MC(2,L),
  & '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 TRNSNT
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(L).LT. RANGEM(1,L)) .OR. (MCHAT(L).GT. RANGEM(2,L)) )
    WRITE(8,*), 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
    & 'ON m IN REGION ', L
  ENDIF
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
REAL MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)

LIST OF VARIABLES
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT( ) 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1, L) = DD(L), THE ESTIMATE FOR M AND MCHAT(2, L) = SUHAT, THE ESTIMATE FOR C
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1, L) IS THE LOWER BOUND AND MZERO(2, L) IS UPPER BOUND
NNODAT NUMBER OF NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1, L) IS THE LOWER BOUND AND RANGEM(2, L) IS THE UPPER BOUND

IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG
DO 100 L = 1, NNODAT
   NUMREG = NUMREG + 1
   LL = NUMREG
   IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG, &
      'LL =', LL, ' MPNT(LL) =', MPNT(LL)
   IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
      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)
      ENDIF
   ELSE
      WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL), &
            'MZERO(1,LL) =', MZERO(1,LL)
   ENDIF
   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
      IF (MCHAT(2,LL) .EQ. 0.0) THEN
         MCHAT(1,LL) = RANGEM(1,LL)
      ELSE
         MCHAT(1,LL) = RANGEM(1,LL)
      ENDIF
   ENDIF
100 CONTINUE
MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
ENDIF

IF (IOUT .EQ. IOOT) WRITE(8,*) 'MCHAT =', MCHAT(I,LL)
ELSE
WRITE(8,*) 'ERROR: OVERALL PRIOR RANGE INCORRECTLY SPECIFIED IN REGION WITHOUT DATA'
CALL TRTMAT
ENDIF

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: TRTMAT
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
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

DO 100 L = NUMREG, 2, -1

IF (RANGEM(2,L-1) .EQ. 0.0) THEN
RANGE IS A POINT IN REGION L-1
IF (RANGEM(1,L-1) .GT. AMAX1(RANGEM(1,L), RANGEM(2,L))) THEN
WRITE(8,*) 'ERROR: POSTERIOR INTERVAL IN REGION ', L,
& ' IS INCONSISTENT WITH POINT POSTERIOR IN REGION ',L-1
CALL TRTMAT
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

7 - 298
C SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER JO HAS BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR CO
C PROGRAMMER: L. NEWLIN
C DATE: CODE: SOCT87 COMMENTS: IDEC87
C
SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)
C INPUTS: NUMREG, RANGEM
C IOUTPUT: MEDM
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON I
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) TO BE USED IN MEDIAN CALCULATION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MEDM() 1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

C INITIALIZE ARRAY MEDM
DO 50 L = 1, MAXREG
MEDM(L) = 0.0
50 CONTINUE
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
        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
    ENDIF
    IF (IOUT .EQ. 10) THEN
        WRITE(*,*) 'L = ', L, ' NUMREG = ', NUMREG
        WRITE(*,*) 'RANGEM(1,L) = ', RANGEM(1,L),
        ' RANGEM(2,L) = ', RANGEM(2,L),
        ' LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
    ENDIF
100 CONTINUE
RETURN
END

SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES OF THE S/N CURVE PARAMETERS
PROGRAMMER: L. NEWLIN
DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89
Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
& ZROREG, NBND, BIGKI, BZHAT)
INPUTS: NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
OUTPUTS: BIGKI, BZHAT
SUBPROGRAMS: TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

IMPLICIT NONE
INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NCOMPS, NF, NPTS(MAXREG), NUMREG, ZROREG
REAL BIGK(0:MAXREG), BIGKI, BZHAT, FACTR, KAT, MEANZ,
& MEDM(MAXREG), MN(0:MAXREG), NBND(0:MAXREG),
**LIST OF VARIABLES**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIGK( )</td>
<td>1-D array containing values of K, where A = K ** M for each region.</td>
</tr>
<tr>
<td>BIGKI</td>
<td>Equal to BIGK(1).</td>
</tr>
<tr>
<td>BZHAT</td>
<td>FACTR × A scale factor = PHI × KRATIO × Z</td>
</tr>
<tr>
<td>FACTR</td>
<td>E(FACTR)</td>
</tr>
<tr>
<td>FINAL ( )</td>
<td>Controls do loop for each region</td>
</tr>
<tr>
<td>MAXDAT</td>
<td>Maximum number of points in S/N data set (per region) allowed.</td>
</tr>
<tr>
<td>MAXREG</td>
<td>Maximum number of regions allowed.</td>
</tr>
<tr>
<td>MEANZ</td>
<td>Sample mean of transformed data, Z = F(STR, NF, NBND, MM).</td>
</tr>
<tr>
<td>MEDM( )</td>
<td>1-D array containing values of the median M for each region.</td>
</tr>
<tr>
<td>MM( )</td>
<td>1-D array containing values of M for each region.</td>
</tr>
<tr>
<td>NBND( )</td>
<td>1-D array containing upper bounds (cycles) for the NUMREG regions of interest</td>
</tr>
<tr>
<td>NUMCOMPS</td>
<td>Number of components -- 1 for stress and strain when decomposed data unavailable -- 2 for decomposed strain data.</td>
</tr>
<tr>
<td>NF( )</td>
<td>2-D array containing RAWNF() (cycles to failure) for the specific material S/N data set broken into regions</td>
</tr>
<tr>
<td>NP</td>
<td>TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET</td>
</tr>
<tr>
<td>NPTS( )</td>
<td>1-D array containing number of points in each region for the specific material S/N data set</td>
</tr>
<tr>
<td>NUMREG</td>
<td>NUMBER OF REGIONS OF INTEREST</td>
</tr>
<tr>
<td>SBND( )</td>
<td>1-D array containing the stress values (psi, R = -1.0) corresponding to the &quot;life boundary&quot; values for each region contained in NBND()</td>
</tr>
<tr>
<td>STR( )</td>
<td>2-D array containing RATSTR() for the specific material S/N data set</td>
</tr>
<tr>
<td>SZERO</td>
<td>STRESS TENSILE TEST POINT, SO</td>
</tr>
<tr>
<td>TRBIGK( )</td>
<td>BIGKI chosen to facilitate region do loop</td>
</tr>
<tr>
<td>ZZ( )</td>
<td>1-D array containing transformed S-N data, Z = F(STR, NF, NBND, MM).</td>
</tr>
</tbody>
</table>

**C INITIATE 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 BETA0 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 FINDSB (NUMREG, ZR0REG, NBND, BIGK, MM, SBND)

C CALCULATE K0 AND M0 FOR THE NO DATA REGION TO THE LEFT IF REQUIRED

DO 150 L = ZR0REG, NUMREG
   TRBIGK(L) = BIGK(L)
150 CONTINUE
   IF (ZR0REG .EQ. 0) THEN
      FACTR = 1.0
      CALL KOMO (SZERO, BIGK, MM, NBND, SBND, TRBIGK,
                     & FACTR, NUMREG)
   ENDIF

C WRITE RESULTS TO FILE
   IF (NCOMPS .EQ. 1) THEN
      WRITE(7,900) NUMREG, BZHAT, KHAT
      IF (IOUT .EQ. 10) WRITE(8,900) NUMREG, BZHAT, KHAT
   DO 200 L = ZR0REG, 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)
   CONTINUE
   WRITE(7,920)
   ELSE
      WRITE(7,930) MM(L), BIGK(L), KHAT
   ENDIF

C FORMAT STATEMENTS
900 FORMAT(/ ///,'PARAMETER VALUES FOR MEDIAN S/N CURVE',///,2X,
& 'NUMBER OF REGIONS:',14.5X,'E(BETA0) =',F8.4,5X,'E(K) =',
& F8.4,///,'REGION','m','K','LIFE BOUND','STRESS BOUND',/)
910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E11.5)
920 FORMAT(/)
930 FORMAT(/ ///,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
& ///,14X,'m',14X,'K',13X,'E(k)',
& ///,7X,F8.5,5X,E12.5,6X,F7.4,/)

RETURN
END
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

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
MEAN DISTRIBUTION FOR EACH REGION
NPPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-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 POSTERIOR NORMAL
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
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
(X = Ln S)

C INITIALIZE ARRAYS

DO 50 L = 1, MAXREG
   MCHAT(1,L) = 0.0
   MCHAT(2,L) = 0.0
   MU(L) = 0.0
   SIG(L) = 0.0
50 CONTINUE

C BEGIN CALCULATION FOR EACH REGION

DO 100 L = 1, NUMREG
   MCHAT(1,L) = - DD(L)
   MCHAT(2,L) = SQRT (SUHAT2(L))
   SUMX2 = NPPR(L) * SX2(L)
   ARG = SUMX2 + DELTA(L)

   IF (DELTA(L) .EQ. 0.0) THEN
      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
   ENDF

7 - 303
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(*,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
  & WRITE(*,*) 'SUHAT2 = ', SUHAT2(L), ' MCHAT2 = ',
  & WRITE(*,*) 'NPFR = ', NPFR(L), ' SX2 = ', SX2(L),
  & WRITE(*,*) 'SUMX2 = ', SUMX2,
  & WRITE(*,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
  & WRITE(*,*) 'MO = ', MO(L), ' MU = ', MU(L)
  & WRITE(*,*) 'SUHAT2 = ', SUHAT2(L), ' SIGMA2 = ', SIGMA2(L),
  & SIG = ', SIG(L)
ENDIF

100 CONTINUE

RETURN
END

C**************************************************
SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND CO TO OBTAIN POSTERIOR RANGES ON M FOR EACH REGION
C SUBPROGRAMS: TRMNAT

C INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM
C OUTPUTS: RANGEM

C 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 CO FOR EACH REGION -- MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR CO
MCPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC() FOR EACH REGION

C********************************************************
MPNT()  1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO()  2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS 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. 1) THEN
      THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER INFORMATION: ASSUME POINT POSTERIOR ON M GIVEN BY THE PRIOR
      RANGEM(1,L) = MZERO(1,L)
      RANGEM(2,L) = 0.0
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L)
         WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L), ' RANGEM(2,L) = ', RANGEM(2,L)
      ENDIF
   ELSEIF ((MPNT(L) .EQ. 2) .AND. (MCPNT(L) .EQ. 0)) THEN
      THERE IS A PRIOR RANGE ON M, BUT NO CO CONSTRAINT USE MO
      RANGEM(1,L) = MZERO(1,L)
      RANGEM(2,L) = MZERO(2,L)
      IF (IOUT .EQ. 10) THEN
         WRITE(8,*), 'MZERO(1,L) = ', MZERO(1,L), ' 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. 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), MCP(1,L))
      UPPER = MZERO(2,L)
      IF (UPPER .LT. LOWER) THEN
         WRITE(8,*), 'ERROR: NO INTERSECTION BETWEEN MO AND MC CALL TRNMAT
      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)
      ENDIF
   ENDIF
100 CONTINUE
WRITE(8,*) 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
CALL TRNAT
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 SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
 DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
 PROGRAMMER: L. NEWLIN
 DATE: CODE: 10FEB88 FORMAT/COMMENTS: 12AUG91
 SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
INPUTS: RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT, MO, SIGMA2

OUTPUTS: RANGEM, MCHAT, MU, SIG, NUMREG

IMPLICIT NONE

INTEGER MAXREG

PARAMETER (MAXREG = 3)

COMMON IOUT

INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

REAL MCHAT(2, MAXREG), MO(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) = M(0,L), THE ESTIMATE FOR M AND MCHAT(2,L) = CHAT, 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 MO() 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

IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG = ', NUMREG

DO 100 L = 1, NNODAT
NUMREG = NUMREG + 1
LL = NUMREG

IF (IOUT .EQ. 10) WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG,
& LL = ', LL, ' MPNT(LL) = ', MPNT(LL)

IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN

RANGEM(1,LL) = MZERO(1,LL)
RANGEM(2,LL) = MZERO(2,LL)
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)
WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
& SIGMA2(LL)
ENDIF

SPECIFY E(M) OF POSTERIOR FOR SAKE OF CALCUATIONS IN SUBROUTINE EXPECTD

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 TRNSAT
ENDIF

100 CONTINUE

RETURN
END
VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE

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 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION

MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION

NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST

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

NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET

NPTS() 1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET

NUMREG NUMBER OF REGIONS OF INTEREST

RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND

RAND RANDOM NUMBER SEED

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

SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

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

SZ2 SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

VARY CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;

1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION;

3 - TRUNCATED NORMAL VARIATION

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

ZZ() 1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,

Z = F(STR, NF, 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
C WRITE(7,900) NUMREG, BZERO
C DO 200 L = ZROREG, NUMREG
   WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
C 200 CONTINUE
C WRITE (7,920)
C FORMAT STATEMENTS

900 FORMAT(3X, 'SELECTED VALUES OF S/M CURVE PARAMETERS', &
         'NUMBER OF REGIONS: ', I4, 5X, 'BETA0 = ', F8.4,
         'REGION', 'N', 'M', 'K', 'LIFE BOUND', 5X, &
         'STRESS BOUND')/
910 FORMAT(5X, I1, 5X, F9.5, 5X, E12.5, 5X, E9.3, 6X, E11.5)
920 FORMAT(///)
RETURN
END

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

C SUBROUTINE FINDM CALCULATES THE VALUE OF M FOR EACH REGION BY
C SAMPLING OFF THE APPROPRIATE M RANGE
C PROGRAMMER: L. HENLIN
C DATE: CODE: 7JUN88 COMMENTS: 13JUL89
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)
C INPUTS: RAND, NUMREG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: RANDOM, TRMNTAT
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
NUMREG NUMBER OF REGIONS OF INTEREST
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
PICK() 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RAND RANDOM NUMBER SEED
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
         FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
         RANGEM(2,L) IS THE UPPER BOUND
X UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
         OFF THE RANGE ON M

C INITIALIZE MM()
DO 50 L = 0, MAXREG
MM(MAXREG) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS

DO 100 L = 1, NUMREG

PICK(1) = 0.0
PICK(2) = 0.0

IF (RANGEM(2,L) .EQ. 0.0) THEN
C M IS SPECIFIED AS A POINT VALUE
MM(L) = RANGEM(1,L)
IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
& 'MM(L) =', MM(L)
ELSEIF (L .EQ. 1) THEN
C SAMPLE ON EXISTING RANGE
CALL 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
C ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
AND THEN SAMPLE
PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
PICK(2) = RANGEM(2,L)
IF (PICK(1) .GT. PICK(2)) THEN
C NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
STOP PROGRAM
WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
CALL TRNNT
ELSE
C SAMPLE ON ADJUSTED RANGE
CALL RANDOM(X, RAND)
MM(L) = (PICK(2) - PICK(1)) * X + PICK(1)
ENDIF
ENDIF
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
100 CONTINUE

RETURN
END

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

C*******************************************************************************
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. GRONDAISKI, L. NEWLIN
DATE: 1DEC87
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
V4.3, V4.4, V4.5
C*********************************************************************************************

7 - 311
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 = 671083.0
RANC = 7090885.0
RANM = 33554432.0

10 RAND = RANA * RAND + RANC
RANDIV = RAND / RANM
RANT = DINT(RANDIV)
RANSUB = RANT * RANM
RAND = RANX - RANSUB
FRAC = RANX / RANM

IF ((FRAC .EQ. 0.0) .OR. (FRAC .EQ. 1.0)) GOTO 10
IF (IOUT .EQ. 2) WRITE(8, *) 'RANX =', RANX, ' RAND =', RAND,
 & ' RANDIV =', RANDIV,
 & ' RANT =', RANT, ' RANSUB =', RANSUB, ' RAND =', RAND,
 & ' FRAC =', FRAC

RETURN
END

C NOTES: IOUT = 2 DUMPS TO SCREEN

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

C SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C PROGRAMMER: L. NEWMAN
C DATE: CODE: 7JUN89 COMMENTS: 13FEB89
C SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
C INPUTS: RAND, NUMREG, MU, SIG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: NORMGN, TRANSAT
C
C IMPLICIT NONE
C
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT

7-312
INTEGER IOUT, L, NUMREG

REAL MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG),
    & SIG(MAXREG), X

DOUBLE PRECISION RAND

C
C LIST OF VARIABLES
C
IOUT     OUTPUT DUMP CONTROLLER
L CONTROLLS 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
          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 NORMGN (RAND, MU(L), SIG(L), X)
          IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
          MM(L) = X
          IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
            & ' RANGEM(2,L) =', RANGEM(2,L)
            WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
          ENDIF
        ELSE
          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 TRANSAT
          ELSE
            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
            IF (IOUT .EQ. 10) THEN
              WRITE(8,*), 'L =', L, ' MM(L-1) =', MM(L-1),
          ENDIF
        ENDIF
      100 CONTINUE

7 - 313
SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

SUBPROGRAM: RANDOM

IMPLICIT NONE

COMMON IOUT

DOUBLE PRECISION RAND
REAL FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2
PARAMETER (PI = 3.1415926536)
INTEGER IOUT

LIST OF VARIABLES
FRAC UNIFORM(0,1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
MU MEAN OF NORMAL DISTRIBUTION
RAND RANDOM NUMBER SEED
SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION
X NORMAL RANDOM VARIATE
U1 UNIFORM RANDOM NUMBER U(0,1)
U2 UNIFORM RANDOM NUMBER U(0,1)
Z1 NORMAL RANDOM NUMBER ON N(0,1)
Z2 NORMAL RANDOM NUMBER ON N(0,1)

IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*), 'RANGEM(1.L) = ', RANGEM(1,L)
& WRITE(8,*), 'RANGEM(2.L) = ', RANGEM(2,L)
& WRITE(8,*), 'PICK(1) = ', PICK(1)
& WRITE(8,*), 'PICK(2) = ', PICK(2)
& WRITE(8,*), 'X = ', X
& 'MM(L) = ', MM(L)
ENDIF
ENDIF

100 CONTINUE
RETURN
END
X = SIGMA * Z1 + MU
IF ((IOUT .EQ. 10) .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 COMMENTS: 13JUL89
C VERSION: MATCHR VS, V8.3, V8.4, V8.5
C MATGRM V4.3, V4.4, V4.5
C
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)

LIST OF VARIABLES
I CONTROLS DO LOOP FOR EACH DATA POINT
IOUT OUTPUT DUMP CONTROLLER
K CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
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 RASTR() 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
C INITIALIZE VARIABLES
NP = 0
DO 50 I = 1, MAXDAT
50 CONTINUE

7 - 315
C BEGIN CALCULATIONS

DO 100 L = 1, NUMREG
   MM(L) = MM(L)
   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-1))
   IF (IOUT .EQ. 10) WRITE(8,*),'LL = ', LL, ' NBND(LL-1) = ', NBND(LL-1), ' MM(LL-1) = ', MM(LL-1), ' MM(LL) = ', MM(LL), ' ZZ = ', ZZ(NP)

300 CONTINUE

200 CONTINUE

100 CONTINUE

RETURN
END

C*******************************************************************************
C SUBROUTINE SMMVAR 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, V6.2, V7, V7.1, V8, V8.1, V8.2,
C V8.2, V8.4, V8.5
C*******************************************************************************

C INPUTS: NP, ZZ
C OUTPUTS: MEANZ, SZ2
C IMPLICIT NONE
C INTEGER MAXDAT
C PARAMETER (MAXDAT = 50)
COMMON IOUT, MAXDAT, MEANZ, NP, ZZ
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. 10) WRITE(8,'(A, I, F)', I = ' ', 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,'(A, I, F)', I = ' ', I, ' SZ2 = ', SZ2)
   200 CONTINUE
   SZ2 = SZ2 / FLOAT(NP - 1)
   IF (IOUT .EQ. 10) WRITE(8,*,'SZ2 = ', SZ2)
RETURN
END

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

C SUBROUTINE KBETA CALCULATES K AND BETAO FROM THE SAMPLE MEAN AND
C VARIANCE OF Z = F(STR, NF, NBND, MM)
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 6OCT87 COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C V8.4, V8.5
SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)

C INPUTS: MEANZ, SZ2
C OUTPUTS: K, BZERO
C IMPLICIT NONE
REAL PI
PARAMETER (PI = 3.1415926536)
COMMON IOUT
INTEGER IOUT
REAL BZERO, K, MEANZ, SZ, SZ2

LIST OF VARIABLES
BZERO VALUE OF WEIBULL PARAMETER, BETAO, CHARACTERIZING THE
SPECIFIC MATERIAL S/W DATA SET
IOUT OUTPUT DUMP CONTROLLER
K VALUE OF K -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL
DATA BASE
MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
PI SELF EXPLANATORY CONSTANT
SZ SZ2 ** 0.5
SZ2 SAMPLE VARIANCE OF THE TRANSFORMED DATA,
   Z = F(STR, NF, NBND, MM)
C PERFORM CALCULATIONS

SZ = SZ2 ** 0.5
BZERO = 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
C EACH REGION
C PROGRAMMER: L. NEWLIN
C DATE: 7JUN88
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
SUBROUTINE FINDK (BZERO, K, MM, NBND, NUMREG, BIGK)

C INPUTS: BZERO, K, MM, NBND, NUMREG
C OUTPUTS: BIGK
C IMPLICIT NONE
INTEGER MAXREG
REAL GAMMA
PARAMETER (GAMMA = 0.57721566490, MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL BIGK(0:MAXREG), BZERO, K, MM(0:MAXREG), NBND(0:MAXREG)

LIST OF VARIABLES

BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
FOR EACH REGION
BZERO VALUE OF WEIBULL PARAMETER, BETA0, 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

7 - 318
DO 50 L = 0, MAXREG
    BIGK(L) = 0.0
50 CONTINUE

C CALCULATE K FOR REGION ONE

    BIGK(1) = (ALOG(2.0) ** (I._0 / BZERO)) * EXP(K + GAMMA / BZERO)
    WRITE(7,*) 'REGION: 1, K =', BIGK(1)
    IF (IOUT .EQ. 10) WRITE(8,*) 'BZERO =', BZERO, ' K =', K,
    & ' GAMMA =', GAMMA, ' BIGK(1) =', BIGK(1)

C CALCULATE K FOR REMAINING REGIONS

DO 100 L = 2, NUMREG
    BIGK(L) = BIGK(L-1) * NBND(L-1) ** ((1.0 / MM(L)) - (1.0 / MM(L-1)))
    WRITE(7,*) 'REGION: ', 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***************************************************************************

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
C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG, ZROREG
REAL BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
    & SBND(0:MAXREG)

LIST OF VARIABLES
BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NUMREG NUMBER OF REGIONS OF INTEREST
SBND() 1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0) CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION CONTAINED IN NBND()
C 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
MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
TRANSFORM METHOD"
PROGRAMMER: L. NEWLIN
DATE: CODE: 18MAR87  COMMENTS: 15SEP89
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
         HEATGN V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
         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 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
C RAND        RANDOM NUMBER SEED
C WEIB       WEIBULL(BETA, ETA) GENERATED RANDOM VARIATE

C CALCULATE CONSTRAINED ETA
ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))

C GENERATE WEIBULL RANDOM VARIATE
CALL RANDOM(FRAC, RAND)
ARG = -ALOG(1.0 - FRAC)
WEIB = ETA * ARG**(1.0/BETA)
IF (IOUT .EQ. 10) WRITE(8,*('BETA = ', BETA, ' ETA = ', ETA,
& ' FRAC = ', FRAC, ' ARG = ', ARG, ' WEIB = ', WEIB)
RETURN
END

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

C SUBROUTINE KOMO calculates k0 and M0 for the zero region (no data region to the left). It accounts for tying up the tensile point at zero, and scaling down the curve if it went above zero.
C PROGRAMMER: L. NEWLIN
C DATE: 1AUG91
C VERSION: MATCHR V8.5, MATGRM V4.5
C Copyright (C) 1990, California Institute of Technology. U.S. Government sponsorship under NASA contract NAS-918 is acknowledged.

C SUBROUTINE KOMO (SZERO, BIGN, M, NBND, TRSBND, TRBIGK, FACTR, NUMREG)
C INPUTS: SZERO, BIGN, M, NBND, TRSBND, FACTR
C OUTPUTS: TRBIGK, M, TRSBND
C
C IMPLICIT NONE
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NUMREG
C REAL BIGN(0:MAXREG), FACTR, M(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
M() 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 BIGN IF TRSBND(0) > SZERO
SZERO STRESS TENSILE TEST POINT, SO
TRBIGK() 1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP SBND(0) < SZERO FOR EACH TRIAL
TRSBND() 1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE

7 - 321
FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS LEVEL BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION

PROGRAMMER:  L. NEWLIN

DATE:  10FEB89


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

REAL FUNCTION GTLIFE (S, MM, BIGK, PHI, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO)

INPUTS:  S, MM, BIGK, PHI, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO

OUTPUTS:  GTLIFE

IMPLICIT NONE

INTEGER IOUT, L, MAXREG, NUMREG, ZROREG

PARAMETER (MAXREG = 3)

COMMON IOUT

REAL BIGK(0:MAXREG), GETLIF, KRATIO, LNZ, MM(0:MAXREG), PHI, S, SBND(0:MAXREG), SZERO, TEMP

LIST OF VARIABLES

BIGK()  1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION

GETLIF  VALUE TO BE ASSIGNED TO GTLIFE -- CYCLES TO FAILURE FOR THE REQUIRED STRESS LEVEL

IOUT  OUTPUT DUMP CONTROLLER

KRATIO  RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS

L  CONTROLS DO LOOP FOR EACH REGION

LNZ  NORMAL(0,PVAR) 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
PHI

WEIBULL(ETA0, BETA0) GENERATED RANDOM VARIATE

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 TEST POINT, SO

TEMP TEMPORARY VARIABLE USED TO PREVENT ARITHMETIC UNDER AND OVER

FLOWS

ZZERO REG ZEROREG -- VALUES CHOSEN TO FACILITATE REGION DO LOOP

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

GETLIF = 0.0

C CALCULATE CYCLES TO FAILURE

IF ((S .GE. SZERO) .AND. (ZZERO .EQ. 0)) THEN

GETLIF = 1.0

ELSE

DO 100 I = ZZERO, NUMREG

IF (S .GT. SBND(I)) THEN

TEMP = MM(I) * (ALOG(BIGK(I)) - ALOG(S) + ALOG(PHI)

& + ALOG(KRATIO) + LNZ)

IF (TEMP .GT. 86.0) THEN

TEMP = 86.0

ENDIF

GETLIF = EXP (TEMP)

GOTO 150

ENDIF

100 CONTINUE

ENDIF

150 CONTINUE

GETLIFE = GETLIF

RETURN

END

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

C SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST

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 IMPLICIT NONE

COMMON IOUT

INTEGER I, IN, 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
MAXM MAXIMUM NUMBER OF M'S TO BE SORTED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
NUM NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
NUMREG NUMBER OF REGIONS OF INTEREST
TEMP TEMPORARY SORTING VARIABLE

DO 400 L = 1, NUMREG
5 INC = NUM
10 IF (INC .GT. 1) THEN
20 INC = INC / 2
20 INORDR = .TRUE.
DO 300 I = 1, (NUM - INC)
300 IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN
300 TEMP = ALLM(I,L)
300 ALLM(I,L) = ALLM(I + INC, L)
300 ALLM(I + INC, L) = TEMP
300 INORDR = .FALSE.
300 CONTINUE
310 IF (.NOT. INORDR) GOTO 20
400 CONTINUE
END
Section 7.3
Materials Characterization Program

The program tree structure, a list of subprograms, a description of the key variables, and the FORTRAN source listing for the materials characterization model code MATCHR are given here. The pertinent materials characterization methodology is given in Section 2.1.2. The overall description of the program and the flowcharts are given in Section 4.1. The user's guide for running MATCHR is given in Section 6.3.

7.3.1 MATCHR Program

7.3.1.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for the stress formulation of MATCHR using Uniform variation on the materials shape parameter \( m \) is given in Figure 7-7, and the tree structure for the truncated Normal case is given in Figure 7-8. The tree structure for the strain formulation of MATCHR using Uniform variation on the materials shape parameters \( m_p \) and \( m_E \) is given in Figure 7-9, and the tree structure for the truncated Normal case is given in Figure 7-10. In all four trees, those subprograms not "shadow-boxed" are random number generators and are described in Section 4.4. The program, subprogram, and file names are indicated by UPPERCASE letters.

7.3.1.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7-7. The section numbers where the subprograms are described by means of flowcharts are given next to the names.

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDREG</td>
<td>4.1.3.9</td>
<td>Adds the ( m ) ranges for the non-data life regions to the right of those with data, for the Uniform distribution case of the stress formulation of the S/N curve.</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>4.1.3.15</td>
<td>Adds the ( m ) ranges for the non-data life regions to the right of those with data, for the truncated Normal case of the stress formulation of the S/N curve.</td>
</tr>
<tr>
<td>ADJSTM</td>
<td>4.1.7</td>
<td>Adjusts the lower bound of the posterior ranges on ( m_E ) to be consistent with the randomly selected value of ( m_p ) of the strain formulation of the S/N curve.</td>
</tr>
</tbody>
</table>
Figure 7-7  Tree Structure for the Stress Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameter $m$
Figure 7-8  Tree Structure for the Stress Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameter $m$
Figure 7-9 Tree Structure for the Strain Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameters $m_p$ and $m_E$
Figure 7-9 Tree Structure for the Strain Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameters $m_p$ and $m_E$ (Cont'd)
Figure 7-10 Tree Structure for the Strain Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameters $m_P$ and $m_E$
Figure 7-10 Tree Structure for the Strain Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameters $m_p$ and $m_E$ (Cont'd)
Table 7-7  List of Subprograms for Program MATCHR (Cont’d)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONCAV&lt;sup&gt;2&lt;/sup&gt;</td>
<td>4.1.3.10</td>
<td>Adjusts the upper bound of the posterior ranges on ( m ) to be consistent with concavity constraints of the stress formulation of the S/N curve.</td>
</tr>
<tr>
<td>CONVRT&lt;sup&gt;3&lt;/sup&gt;</td>
<td>4.1.3.3</td>
<td>Transforms stress data to equivalent zero-mean stresses with stress ratio of (-1.0) for the stress formulation of the S/N curve.</td>
</tr>
<tr>
<td>DECOMP&lt;sup&gt;4&lt;/sup&gt;</td>
<td>4.1.4</td>
<td>Controls the logical flow for the plastic-elastic strain decomposition and the information aggregation portion of the strain formulation materials characterization model.</td>
</tr>
<tr>
<td>EXPCTD&lt;sup&gt;5&lt;/sup&gt;</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>FCT</td>
<td>4.1.9.2</td>
<td>Calculates the value of the function and its derivative at the current iteration value of life, in order to find the solution of the strain formulation S/N curve given by Equation 2-50.</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.</td>
</tr>
<tr>
<td>FINDM&lt;sup&gt;6&lt;/sup&gt;</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 for each life region implied by the constraint on the coefficient of variation of fatigue strength ( C ) by using Equations 2-28 through 2-32 for the stress formulation of the S/N curve.</td>
</tr>
<tr>
<td>FINDMN&lt;sup&gt;6&lt;/sup&gt;</td>
<td>4.1.5.2</td>
<td>Obtains the value of ( m ) for each life region by sampling from the appropriate truncated Normal distribution on ( m ).</td>
</tr>
<tr>
<td>FINDSB</td>
<td>4.1.5.7</td>
<td>Calculates the life region &quot;tie-points&quot; or stress values which correspond to the &quot;life boundaries&quot; conditional on the randomly selected ( m ) for each region. Also calculates ( K ), characterizing the specific material S/N data set, which is a function of ( \beta_o ) and ( k ).</td>
</tr>
<tr>
<td>FNDRNG&lt;sup&gt;7&lt;/sup&gt;</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 stress formulation S/N curve of Equation 2-48.</td>
</tr>
<tr>
<td>GTLIF2</td>
<td>4.1.9</td>
<td>Calculates the cycles to failure for a particular strain based upon the materials characterization model strain formulation S/N curve of Equation 2-50.</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>4.1.3.7</td>
<td>Calculates ( \sigma^2 ), Equation 2-49, the extent of departures from the multiple heat median S/N curve warranted by the information available.</td>
</tr>
</tbody>
</table>
Table 7-7  List of Subprograms for Program MATCHR (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>PURPOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>INFAGG</td>
<td>4.1.3</td>
<td>Controls the logical flow for the information aggregation portion of the stress formulation 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 stress formulation information aggregation subroutine, INFAGG, to zero.</td>
</tr>
<tr>
<td>INITD</td>
<td>4.1.4.1</td>
<td>Initializes the entries of the arrays used in the strain formulation information aggregation subroutine, DECOMP, to zero.</td>
</tr>
<tr>
<td>INTRVL</td>
<td>4.1.3.6</td>
<td>Calculates the 95% confidence Intervals for C and Jo for m, for each region by using Equations 2-24 and 2-26.</td>
</tr>
<tr>
<td>KBETA</td>
<td>4.1.5.5</td>
<td>Calculates k and β̂o 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.</td>
</tr>
<tr>
<td>KOMO</td>
<td>4.1.6</td>
<td>Calculates K̂o and m̂o for the zero region, the no data region to the left of the first data region. Extends the stress formulation S/N curve consistent with the tensile point at S_o.</td>
</tr>
<tr>
<td>MATCHR</td>
<td>4.1.2</td>
<td>The main routine that controls the logical flow of the calculations of the materials characterization model.</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 for the stress formulation S/N curve.</td>
</tr>
<tr>
<td>MUSIG</td>
<td>4.1.3.13</td>
<td>Calculates the posterior Normal distribution parameters: mean m̂ and standard deviation σ̂, for each life region of the S/N curve.</td>
</tr>
<tr>
<td>NEWTON</td>
<td>4.1.9.1</td>
<td>Solves the general nonlinear equations of the form f(x) = 0 by means of Newton's iteration method.</td>
</tr>
<tr>
<td>NORMGN</td>
<td>4.4.3</td>
<td>Generates Normal(μ, σ²) random variates.</td>
</tr>
<tr>
<td>NORTNG</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>PECOMP</td>
<td>4.1.4.4</td>
<td>Calculates the plastic-elastic strain decomposition by using Equations 2-45 through 2-47 for the information aggregation calculations of the strain formulation S/N curve.</td>
</tr>
<tr>
<td>PREP</td>
<td>4.1.4.3</td>
<td>Places the plastic-elastic decomposition strain data into arrays of the appropriate data structure to allow usage of routines of the stress formulation S/N curve calculations.</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>NAME</td>
<td>SECTION</td>
<td>PURPOSE</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>RCE</td>
<td>4.1.3.2</td>
<td>Reads the data from SPECFD and RELATD; calls CONVRT to transform the stress data to a stress ratio of $-1.0$; and echoes the data to SPECFO and RELATO for the stress formulation S/N curve. RCE also breaks S/N data sets into regions as specified by the user.</td>
</tr>
<tr>
<td>RDECHO</td>
<td>4.1.4.2</td>
<td>Reads the data from SPECFD and RELATD and then echoes the data to SPECFO and RELATO for the strain formulation S/N curve.</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.</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>4.1.3.4</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; 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 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>

---

1. No data regions to the right are discussed on Page 2-17.
2. Concavity constraints are discussed on Pages 2-13 through 2-14.
3. The stress transformation is discussed on Page 2-7.
4. Plastic-elastic strain decomposition is discussed on Page 2-20; the information aggregation calculations are discussed on Pages 2-6 through 2-14.
5. The median S/N curve parameter estimation calculations are described on Pages 2-15 through 2-18.
6. Selection of the $\{m_j\}$ parameters is discussed on Page 2-15.
7. Combining information to obtain the posterior credibility ranges on $m$ is discussed on Page 2-13.
8. The information aggregation calculations are discussed on Pages 2-6 through 2-14.
9. Extension of the S/N curve to the left is discussed on Page 2-17.
10. Calculation of the truncated Normal distribution parameters is discussed on Page 2-14.
11. The Normal distribution is discussed on Page 2-23.
12. The parameter estimation calculations are discussed on Pages 2-15 through 2-18.
13. The Uniform distribution is discussed on Page 2-23.
14. The need for saving $m$'s is discussed on Page 2-15.
15. The S/N data transformation is discussed on Page 2-16.
7.3.1.3 Description of Variables

A list of variables used in the materials characterization model code MATCHR is given in Table 7-8. The chart of Table 7-9 provides a cross-reference among routines, or subprograms, and their variable usage.\(^1\) The variable names are indicated by BOLD UPPERCASE letters; the variable "type" is specified as follows: CH40 is a standard character variable, forty characters long; INT is a standard integer variable; RE is a standard real variable; DRE is a double precision variable; and LOG is a standard logical variable. The various array dimensions are defined by using the following parameters: CHITAB, MAXDAT, MAXMM, MAXREG, MAXSET, MAXTNS, and TTAB. In the strain formulation of the materials characterization model, those variables with MAXREG as a dimension use only region 1.

Table 7-8 List of Variables for Program MATCHR
(Footnotes are at the end of the table)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALLM(MAXMM, MAXREG)</td>
<td>RE</td>
<td>2-D array containing the materials model shape parameters (m's) for each life region to be used in the truncated Normal median S/N curve calculation.(^1)</td>
</tr>
<tr>
<td>ALPHA(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the ratio of (S_p) to (S_E), Equation 2-47, for each total strain point.</td>
</tr>
<tr>
<td>ARG, ARG1, ARG2</td>
<td>RE</td>
<td>Intermediate calculation variables.</td>
</tr>
<tr>
<td>BB(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of (b = \frac{S_{XY}(L)}{SY^2(L)}), Equation 2-21, for each life region of the S/N curve.</td>
</tr>
<tr>
<td>BIGK(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the materials model location parameter (K), Equation 2-12, where (A = K^{m}).</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>BIGKE(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the value of the materials model location parameter (K_E), Equation 2-43, for the elastic strain components.</td>
</tr>
<tr>
<td>BIGKHT</td>
<td>RE</td>
<td>Value of the materials model location parameter (K) for life region 1 corresponding to the median value(s) for the materials model shape parameter(s).</td>
</tr>
<tr>
<td>BIGKP(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing the value of the materials model location parameter (K_p), Equation 2-43, for the plastic strain components.</td>
</tr>
</tbody>
</table>

\(^1\) A numeric entry in Table 7-9 indicates the appropriate definition of Table 7-8 to be used for the variable and routine of interest.
<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BZERO</td>
<td>RE</td>
<td>Estimate of Weibull distribution shape parameter $\beta_o$, Equation 2-11, which characterizes the intrinsic variation of the S/N data set.</td>
</tr>
<tr>
<td>BZEROE</td>
<td>RE</td>
<td>Estimate of Weibull distribution shape parameter, $\beta_{OE}$, that characterizes the intrinsic variation of the elastic strain components of the S/N data set, used to estimate $\beta_o$ in Equation 2-43.</td>
</tr>
<tr>
<td>BZEROP</td>
<td>RE</td>
<td>Estimate of Weibull distribution shape parameter, $\beta_{OP}$, that characterizes the intrinsic variation of the plastic strain components of the S/N data set, used to estimate $\beta_o$ in Equation 2-43.</td>
</tr>
<tr>
<td>BZHAT</td>
<td>RE</td>
<td>Value of $\beta_o$ corresponding to the median S/N curve.</td>
</tr>
<tr>
<td>CHI025(CHITAB)</td>
<td>RE</td>
<td>Table of 0.025 percentage points for the $\chi^2$ distribution used in Equation 2-24.</td>
</tr>
<tr>
<td>CHI975(CHITAB)</td>
<td>RE</td>
<td>Table of 0.975 percentage points for the $\chi^2$ distribution used in Equation 2-24.</td>
</tr>
<tr>
<td>CHITAB</td>
<td>INT</td>
<td>Maximum number of degrees of freedom in CHI025() and CHI975(). The maximum number is 150.</td>
</tr>
<tr>
<td>COUNT</td>
<td>INT</td>
<td>Array index variable used by RCE to facilitate data input, stress ratio transformation, life region partitioning, and data echo.</td>
</tr>
<tr>
<td>CZERO</td>
<td>RE</td>
<td>$C_o$ in Equations 2-28 through 2-32, the user-specified upper bound of the coefficient of variation of material fatigue strength.</td>
</tr>
<tr>
<td>CZERO2</td>
<td>RE</td>
<td>Value of CZERO ** 2.</td>
</tr>
<tr>
<td>DD(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of $d = \frac{S_{XY}(L)}{S_{X2}(L)}$, Equation 2-20, for each life region of the S/N curve.</td>
</tr>
<tr>
<td>DELTA(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the shape parameter $\delta$ of the Bayesian prior distribution for each life region.</td>
</tr>
<tr>
<td>DELTAE(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the value of the shape parameter $\delta_E$ of the Bayesian prior distribution for the elastic strain components.</td>
</tr>
<tr>
<td>DELTAP(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the value of the shape parameter $\delta_P$ of the Bayesian prior distribution for the plastic strain components.</td>
</tr>
<tr>
<td>DESCRP(0:MAXSET)</td>
<td>CH40</td>
<td>1-D array containing descriptions of each S/N data set.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DERF</td>
<td>RE</td>
<td>DERivative of F. The resultant value of the derivative of F at the root during the Newton’s method calculations.</td>
</tr>
<tr>
<td>DIFFX(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the difference between LNSTR(K,J,L) and MEANX(J) for each point in each S/N data set for region L, used in calculations for Equations 2-20 and 2-21.</td>
</tr>
<tr>
<td>DIFFY(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the difference between LNNF(K,J,L) and MEANY(J) for each point in each S/N data set for region L used in calculations for Equations 2-20 and 2-21.</td>
</tr>
<tr>
<td>ELAS(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the elastic strain components $S_E$ (%).</td>
</tr>
<tr>
<td>EPS</td>
<td>RE</td>
<td>Value of the maximum error allowed in the result of the Newton’s method calculations.</td>
</tr>
<tr>
<td>F</td>
<td>RE</td>
<td>Resultant value of the function F at the root during the Newton’s method calculations.</td>
</tr>
<tr>
<td>FACTR</td>
<td>RE</td>
<td>Value of $(\text{PHI} \times \text{KRATIO} \times Z)$ in Equation 2-48.</td>
</tr>
<tr>
<td>FTU</td>
<td>RE</td>
<td>Material ultimate strength (psi).</td>
</tr>
<tr>
<td>FTUZ</td>
<td>RE</td>
<td>Material ultimate strength (psi) for specific material.</td>
</tr>
<tr>
<td>FTY</td>
<td>RE</td>
<td>Material yield strength (psi).</td>
</tr>
<tr>
<td>FTYZ</td>
<td>RE</td>
<td>Material yield strength (psi) for specific material.</td>
</tr>
<tr>
<td>GAMMA</td>
<td>RE</td>
<td>Euler’s constant.</td>
</tr>
<tr>
<td>1) GETLIFE</td>
<td>RE</td>
<td>Value to be assigned to GTLIFE, the fatigue cycles to failure at the required stress level.</td>
</tr>
<tr>
<td>2) GETLIF</td>
<td>RE</td>
<td>Value to be assigned to GTLIF2, the fatigue cycles to failure at the required strain level.</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>RE</td>
<td>Function that calculates the fatigue cycles to failure at a given stress, given by Equation 2-48.</td>
</tr>
<tr>
<td>GTLIF2</td>
<td>RE</td>
<td>Function that calculates the fatigue cycles to failure at a given strain, given by Equation 2-50.</td>
</tr>
<tr>
<td>1) I</td>
<td>INT</td>
<td>Controls DO loop for each data point in an S/N data set.</td>
</tr>
<tr>
<td>2) I</td>
<td>INT</td>
<td>Controls the DATA statements for CHI025( ) and CHI975( ).</td>
</tr>
<tr>
<td>3) I</td>
<td>INT</td>
<td>Controls insertion sort DO loop.</td>
</tr>
<tr>
<td>IEND</td>
<td>INT</td>
<td>Maximum number of iteration steps allowed to find the solution during the Newton’s method calculations.</td>
</tr>
</tbody>
</table>
### Table 7-8 List of Variables for Program MATCHR (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>IER</td>
<td>INT</td>
<td>Provides information on the type of error occurring in the Newton’s method calculations. A value of 0 indicates that no error occurred; a value of 1 indicates no convergence after IEND iteration steps; and a value of 2 indicates that at some iteration step, the derivative DERF was equal to zero.</td>
</tr>
<tr>
<td>II</td>
<td>INT</td>
<td>Controls DO loop for each trial when the empirical median S/N curve for the truncated Normal case has been requested.</td>
</tr>
<tr>
<td>INC</td>
<td>INT</td>
<td>Increment variable used during insertion sort.</td>
</tr>
<tr>
<td>INORDR</td>
<td>LOG</td>
<td>Logical flag to indicate whether the insertion sort has been completed.</td>
</tr>
<tr>
<td>INVME</td>
<td>RE</td>
<td>INVerse of ME, the value of 1 / ME.</td>
</tr>
<tr>
<td>INVMP</td>
<td>RE</td>
<td>INVerse of MP, the value of 1 / MP.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>IZERO(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the 95% confidence interval for C, l_o, Equation 2-24, for each life region.</td>
</tr>
<tr>
<td>1) J</td>
<td>INT</td>
<td>Controls DO loop for each S/N data set.</td>
</tr>
<tr>
<td>2) J</td>
<td>INT</td>
<td>S/N data set of interest during stress ratio transformation calculations.</td>
</tr>
<tr>
<td>JZERO(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the 95% confidence interval for m, J_o, Equation 2-26, for each life region.</td>
</tr>
<tr>
<td>1) K</td>
<td>INT</td>
<td>Controls DO loop for each point in a life region.</td>
</tr>
<tr>
<td>2) K</td>
<td>RE</td>
<td>k in Equations 2-40 through 2-42, a parameter that characterizes the specific material S/N data base for a set of ( {m_j} ).</td>
</tr>
<tr>
<td>1) KE</td>
<td>RE</td>
<td>( K_E ) in Equation 2-47, the median ( K_E ) for the user-provided elastic strain components. (See KHATE in DECOMP)</td>
</tr>
<tr>
<td>2) KE</td>
<td>RE</td>
<td>Scalar variable used by function GTLIF2, equal to BIGKE(1).</td>
</tr>
<tr>
<td>KEPROD</td>
<td>RE</td>
<td>Value of KE * PHI * KRATIO * Z.</td>
</tr>
<tr>
<td>KHAT</td>
<td>RE</td>
<td>Value of ( k ) corresponding to the median S/N curve.</td>
</tr>
<tr>
<td>KHATE</td>
<td>RE</td>
<td>( K_E ) in Equation 2-47, the median ( K_E ) for the user-provided elastic strain components.</td>
</tr>
<tr>
<td>KHATP</td>
<td>RE</td>
<td>( K_P ) in Equation 2-47, the median ( K_P ) for the user-provided plastic strain components.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>KP</td>
<td>RE</td>
<td>$K_p$ in Equation 2-47, the median $K_p$ for the user-provided plastic strain components. (See KHATP in DECOMP)</td>
</tr>
<tr>
<td>KP</td>
<td>RE</td>
<td>Scalar variable used by function GTUF2, equal to BIGKP(1).</td>
</tr>
<tr>
<td>KPPROD</td>
<td>RE</td>
<td>Value of $KP \times PHI \times KRATIO \times Z^6$</td>
</tr>
<tr>
<td>KRATIO</td>
<td>RE</td>
<td>Ratio of MED $K^*/MED K$ in Equation 2-48. KRATIO is constant over life regions and strain components for the materials model.</td>
</tr>
<tr>
<td>KTERM</td>
<td>RE</td>
<td>Value of $KP / KE$.</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>LAMBDA(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of $\lambda_j$, Equations 2-46 and 2-47, the plastic fraction of the total strain.</td>
</tr>
<tr>
<td>LAMN</td>
<td>RE</td>
<td>$\lambda_N$, the ratio of $V(\ln N</td>
</tr>
<tr>
<td>LIFE</td>
<td>RE</td>
<td>The fatigue cycles to failure at a given stress or strain given by Equation 2-48 or 2-50.</td>
</tr>
<tr>
<td>LL</td>
<td>INT</td>
<td>Value of $N$ for an iteration during the Newton's method calculations.</td>
</tr>
<tr>
<td></td>
<td>INT</td>
<td>Value of NUMREG for a set of calculations.</td>
</tr>
<tr>
<td></td>
<td>INT</td>
<td>Controls inner DO loop for each life region of the S/N curve.</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>LNNF(MAXDAT, 0:MAXSET, MAXREG)</td>
<td>RE</td>
<td>3-D array containing values of $\ln(RAWNDF( ))$ indexed for life regions.</td>
</tr>
<tr>
<td>LNSTR(MAXDAT, 0:MAXSET, MAXREG)</td>
<td>RE</td>
<td>3-D array containing values of $\ln(RATSTR( ))$ indexed for life regions.</td>
</tr>
<tr>
<td>LNSTRE(MAXDAT, 0:MAXSET, MAXREG)</td>
<td>RE</td>
<td>3-D array containing values of $\ln(SE( ))$ or $\ln(STRE( ))$ for the elastic strain components.</td>
</tr>
<tr>
<td>LNSRP(MAXDAT, 0:MAXSET, MAXREG)</td>
<td>RE</td>
<td>3-D array containing values of $\ln(SP( ))$ or $\ln(STRP( ))$ for the plastic strain components.</td>
</tr>
<tr>
<td>LNZ</td>
<td>RE</td>
<td>$\ln(Z)$ in Equation 2-48, the Normal(0, PVAR) random variate for the materials process variation aspect of the materials model.</td>
</tr>
</tbody>
</table>
Table 7-8  List of Variables for Program MATCHR (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOWER</td>
<td>RE</td>
<td>The lower bound of the intersection to find ( x(m) ) in a particular region.(^9)</td>
</tr>
<tr>
<td>LOWERM</td>
<td>RE</td>
<td>( \max (m_i, L_{i+1}) ) in Equation 2-33, the lower bound of the ( m ) range, due to concavity considerations, to be used in the median calculation.(^10)</td>
</tr>
<tr>
<td>LPHIM(0:MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of ( \ln(\Phi_i) \times MM ) for each life region of the S/N curve.</td>
</tr>
<tr>
<td>1) M</td>
<td>INT</td>
<td>Controls DO loop for each of the NDIV data divisions during data input, stress formulation.</td>
</tr>
<tr>
<td>2) M</td>
<td>INT</td>
<td>Controls DO loop for each tensile test point in an S/N data set, strain formulation.</td>
</tr>
<tr>
<td>MAXDAT</td>
<td>INT</td>
<td>Maximum number of points per data set per region allowed for S/N curve. The maximum number of data points per set allowed 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>MAXSET</td>
<td>INT</td>
<td>Maximum number of related S/N data sets allowed. The maximum number of related data sets is 5.</td>
</tr>
<tr>
<td>MAXTNS</td>
<td>INT</td>
<td>Maximum number of tensile test points per data set allowed for the strain formulation. The maximum number of tensile points is 5.</td>
</tr>
<tr>
<td>MC(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the ranges on the materials shape parameter ( m ) for each life region consistent with ( C_o ), the user-specified upper bound for the coefficient of variation of fatigue strength ( C ) and the S/N data. The ranges for ( m )'s implied by a specified bound ( C_o ) are given in Equations 2-28 through 2-32. ( MC(1,L) ) is the lower bound and ( MC(2,L) ) is the upper bound in region ( L ).</td>
</tr>
<tr>
<td>MCE(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the range on the materials shape parameter ( m_C ) for the elastic strain components consistent with ( C_o ), the user-specified upper bound for the coefficient of variation of fatigue strength ( C ) and the S/N data. Note: the ( C ) constraint is not utilized at this time by the strain formulation, but is a necessary parameter to some of the subroutines shared with the stress formulation.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1) MCHAT(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the point estimates of m and C for each region, given in Equation 2-22. MCHAT(1,L) = - DD(L), the point estimate $\hat{m}$ and MCHAT(2,L) = SUHAT, the point estimate $\hat{C}$.</td>
</tr>
<tr>
<td>2) MCHAT(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the point estimates of $C_P$ and $C_E$ given in Equation 2-22. MCHAT(2,1) = MCHATP(2,1), and MCHAT(2,2) = MCHATE(2,1).</td>
</tr>
<tr>
<td>MCHATE(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the point estimates of $m_E$ and $C_E$ for the elastic strain components, given in Equation 2-22. MCHATE(1,1) = - DD(1), the point estimate $\hat{m}_E$ and MCHATE(2,1) = SUHAT, the point estimate $\hat{C}_E$.</td>
</tr>
<tr>
<td>MCHATP(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the point estimates of $m_P$ and $C_P$ for the plastic strain components, given in Equation 2-22. MCHATP(1,1) = - DD(1), the point estimate $\hat{m}_P$ and MCHATP(2,1) = SUHAT, the point estimate $\hat{C}_P$.</td>
</tr>
<tr>
<td>MCP(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the range on the materials shape parameter $m_P$ for the plastic strain components consistent with $C_P$, the user-specified upper bound for the coefficient of variation of fatigue strength $C$ and the S/N data. Note: the $C$ constraint is not utilized at this time by the strain formulation, but is a necessary parameter to some of the subroutines shared with the stress formulation.</td>
</tr>
<tr>
<td>MCPNT(MAXREG)</td>
<td>INT</td>
<td>1-D array containing the number of points, 0, 1, or 2, in MC() for each life region.</td>
</tr>
<tr>
<td>MCPNTE(MAXREG)</td>
<td>INT</td>
<td>1-D array containing the number of points, 0, 1, or 2, in MCE(). All entries are zero for the strain formulation.</td>
</tr>
<tr>
<td>MCPNTP(MAXREG)</td>
<td>INT</td>
<td>1-D array containing the number of points, 0, 1, or 2, in MCP(). All entries are zero for the strain formulation.</td>
</tr>
<tr>
<td>1) ME(0:MAXREG)</td>
<td>RE</td>
<td>$m_E$ in Equation 2-43, the 1-D array containing the randomly selected value of the materials model shape parameter for the elastic strain component of the S/N curve.</td>
</tr>
<tr>
<td>2) ME</td>
<td>RE</td>
<td>$m_E$ in Equation 2-47, the median $m_E$ for the user-provided elastic strain components.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ME</td>
<td>RE</td>
<td>Scalar variable used by function GTLIF2, equal to ME(1).</td>
</tr>
<tr>
<td>MEANX(0:MAXSET)</td>
<td>RE</td>
<td>1-D array containing values of $\bar{x}_j$, Equations 2-20 and 2-21, the sample X mean for points from region L and data set J, where $X = \ln S$.</td>
</tr>
<tr>
<td>MEANY(0:MAXSET)</td>
<td>RE</td>
<td>1-D array containing values of $\bar{y}_j$, Equations 2-20 and 2-21, the sample Y mean for points from region L and data set J, where $Y = \ln N$.</td>
</tr>
<tr>
<td>MEANZ</td>
<td>RE</td>
<td>$\bar{Z}$ in Equation 2-42, the sample mean of the transformed data of Equations 2-39 through 2-41.</td>
</tr>
<tr>
<td>MEBND(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing the allowable posterior credibility range on $m_E$ for the elastic strain components, after the random selection of $m_P$ for the plastic strain components.</td>
</tr>
<tr>
<td>MEDM(MAXMM)</td>
<td>RE</td>
<td>1-D array containing the median $m$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>MEDME(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the median $m_E$ for the elastic strain component of the S/N curve.</td>
</tr>
<tr>
<td>MEDMP(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the median $m_P$ for the plastic strain component of the S/N curve.</td>
</tr>
<tr>
<td>MID</td>
<td>INT</td>
<td>Pointer to the median $m$ values in array SORTM().</td>
</tr>
<tr>
<td>MM(0:MAXREG)</td>
<td>RE</td>
<td>$m_j$ in Equation 2-12, the 1-D array containing the randomly selected values of the materials model shape parameter $m$ for each life region of the S/N curve.</td>
</tr>
<tr>
<td>MML</td>
<td>RE</td>
<td>Value of MM(L) for a set of calculations.</td>
</tr>
<tr>
<td>MO(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the location parameter $m_o$ of the Bayesian prior distribution of the shape parameter $m$ for each life region.</td>
</tr>
<tr>
<td>MOE(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the value of the location parameter $m_{OE}$ of the Bayesian prior distribution of the shape parameter $m_E$ for the elastic strain components.</td>
</tr>
<tr>
<td>MOP(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the value of the location parameter $m_{OP}$ of the Bayesian prior distribution of the shape parameter $m_P$ for the plastic strain components.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>1) MP(0:MAXREG)</td>
<td>RE</td>
<td>$m_p$ in Equation 2-43, the 1-D array containing the randomly selected value of the materials model shape parameter for the plastic strain component of the S/N curve.</td>
</tr>
<tr>
<td>2) MP</td>
<td>RE</td>
<td>$m_p$ in Equation 2-47, the median $m_p$ for the user-provided plastic strain components.</td>
</tr>
<tr>
<td>3) MP</td>
<td>RE</td>
<td>Scalar variable used by function GTLIF2, equal to $MP(1)$.</td>
</tr>
<tr>
<td>MPNT(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the number of points, 0, 1, or 2, in MZERO( ) for each life region.</td>
</tr>
<tr>
<td>MPNTE(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the number of points, 0, 1, or 2, in MZEROE( ) for the elastic strain component.</td>
</tr>
<tr>
<td>MPNTP(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the number of points, 0, 1, or 2, in MZEROP( ) for the plastic strain component.</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>MTERM</td>
<td>RE</td>
<td>Value of $-[(1/MP) - (1/ME)]$.</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>MUE(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution mean of the materials shape parameter $m_E$ for the elastic strain components of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>MUP(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution mean of the materials shape parameter $m_p$ for the plastic strain components of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>MZERO(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the user-specified explicit constraints on the shape parameter $m$ for each life region, given in Equation 2-27. MZERO(1,L) is the lower bound and MZERO(2,L) is the upper bound.</td>
</tr>
<tr>
<td>MZEROE(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the user-specified explicit constraints on the shape parameter $m_E$ for the elastic strain components. MZEROE(1,1) is the lower bound and MZEROE(2,1) is the upper bound.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MZERO P(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the user-specified explicit constraints on the shape parameter ( m_p ) for the plastic strain components. MZERO P(1, 1) is the lower bound and MZERO P(2, 1) is the upper bound.</td>
</tr>
<tr>
<td>NBND(0:MAXREG)</td>
<td>RE</td>
<td>( N_{i,j}^{+} ) in Equation 2-35, 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>NCOMPS</td>
<td>INT</td>
<td>Number of ComponentS. Controls the formulation of the S/N curve to be used. A value of 1 indicates the stress formulation, while a value of 2 indicates the strain formulation.</td>
</tr>
<tr>
<td>NDC(0:MAXSET)</td>
<td>INT</td>
<td>Number of Decomposed points. 1-D array containing the number of strain data points with user-provided plastic and elastic component information for each S/N data set.</td>
</tr>
<tr>
<td>NDIV</td>
<td>INT</td>
<td>Number of input data divisions for an S/N data set. A data set is broken into divisions by (stress ratio, life region) pairs during input.</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>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>NNODAT</td>
<td>INT</td>
<td>Number of NO DATA regions, the number of life regions to the right with no S/N data.</td>
</tr>
<tr>
<td>1) NP(0:MAXSET, MAXREG)</td>
<td>INT</td>
<td>2-D array containing number of points of each S/N data set in each life region.</td>
</tr>
<tr>
<td>2) NP</td>
<td>INT</td>
<td>The total number of points in the specific material S/N data set.</td>
</tr>
<tr>
<td>NPPR(MAXREG)</td>
<td>INT</td>
<td>Number of Points Per Region. 1-D array containing values of ( N ) in Equation 2-20 for each life region.</td>
</tr>
<tr>
<td>1) 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>2) NPTS(0:MAXSET)</td>
<td>INT</td>
<td>1-D array containing the total number of points in each S/N data set.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NSETS</td>
<td>INT</td>
<td>( P ) in Equation 2-20, the number of related material S/N data sets.</td>
</tr>
<tr>
<td>NTENS(0:MAXSET)</td>
<td>INT</td>
<td>1-D array containing the number of tensile test data points for each S/N data set for the strain formulation.</td>
</tr>
<tr>
<td>NTRIAL</td>
<td>INT</td>
<td>Number of ( m )'s to be used to calculate the empirical median S/N curve for the truncated Normal case.</td>
</tr>
<tr>
<td>1) NUM</td>
<td>INT</td>
<td>Value of NPPR(( L )) for a set of calculations.</td>
</tr>
<tr>
<td>2) NUM</td>
<td>INT</td>
<td>Number of S/N data points in a particular data division during data input.</td>
</tr>
<tr>
<td>3) NUM(MAXREG)</td>
<td>INT</td>
<td>1-D array containing values of ( [ (Z \cdot N_j) - 1 ] ) for each life region.</td>
</tr>
<tr>
<td>4) NUM(0:MAXSET)</td>
<td>INT</td>
<td>1-D array containing the number of points in each S/N data set.</td>
</tr>
<tr>
<td>5) NUM</td>
<td>INT</td>
<td>Number of elements in array ALLM( ) to be sorted.</td>
</tr>
<tr>
<td>NUM1</td>
<td>INT</td>
<td>Index of first point in STR( ) to have the stress ratio transformation performed.</td>
</tr>
<tr>
<td>NUM2</td>
<td>INT</td>
<td>Index of last point in STR( ) to have the stress ratio transformation performed.</td>
</tr>
<tr>
<td>NUMREG</td>
<td>INT</td>
<td>( R ) in Equation 2-11, the number of life regions of interest in the S/N curve.</td>
</tr>
<tr>
<td>NZERO</td>
<td>RE</td>
<td>( N_o ), the life from the elastic portion of the strain formulation S/N curve used as the initial value in the Newton's method calculations.</td>
</tr>
<tr>
<td>PHI</td>
<td>RE</td>
<td>( \varphi ) in Equation 2-11, the materials intrinsic variation, or scatter, given by a Weibull(( \beta_o ), ( \eta_o ) (( \beta_o ))) random variate.</td>
</tr>
<tr>
<td>PI</td>
<td>RE</td>
<td>( \pi ), constant equal to 3.1415926536.</td>
</tr>
<tr>
<td>PICK(2)</td>
<td>RE</td>
<td>1-D array containing the bounds of the credibility range for ( m ) in a region adjusted for concavity constraints after the random selection of ( m ) in the life region to the left. PICK( ) is only used in multiple region cases.</td>
</tr>
<tr>
<td>PLAS(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the plastic strain components ( S_p )(%).</td>
</tr>
<tr>
<td>PSIG</td>
<td>RE</td>
<td>( \sigma ) in Equation 2-48, the value of SQRT(( PVAR )).</td>
</tr>
<tr>
<td>PSIG2(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the estimates of ( \sigma^2 ), Equation 2-49, for each life region.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------</td>
<td>------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PVAR</td>
<td>RE</td>
<td>$\sigma^2$ in Equation 2-48, 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>S/N data stress ratio, a stress ratio of $-1.0$ is desired.</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>RANGME(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the posterior credibility range on the materials model shape parameter $m_E$ for the elastic strain components. RANGME(1,1) is the lower bound and RANGME(2,1) is the upper bound.</td>
</tr>
<tr>
<td>RANGMP(2, MAXREG)</td>
<td>RE</td>
<td>2-D array containing values of the posterior credibility range on the materials model shape parameter $m_P$ for the plastic strain components. RANGMP(1,1) is the lower bound and RANGMP(2,1) is the upper bound.</td>
</tr>
<tr>
<td>RATIO</td>
<td>RE</td>
<td>S/N data stress ratio, a stress ratio of $-1.0$ is desired.</td>
</tr>
<tr>
<td>RATSTR(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing stress points (psi) with stress ratio = $-1.0$, or strain total points (%) for all S/N data sets.</td>
</tr>
<tr>
<td>RAWNF(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing the cycles to failure data for all S/N data sets.</td>
</tr>
<tr>
<td>RAWSTR(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing the user-provided stress data (psi) or total strain data (%) for all S/N data sets.</td>
</tr>
<tr>
<td>REFNP(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>REG</td>
<td>INT</td>
<td>Life region of interest for a particular data division during data input for the stress formulation.</td>
</tr>
<tr>
<td>RFNP(MAXREG)</td>
<td>INT</td>
<td>1-D array containing the number of points for the specific material S/N data set, RFNP(1) = NP(0,1). Note: for the strain formulation case only region 1 is used.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RFSTRE(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing elastic strain component points (%), SE( ) or STRE( ), for the specific material S/N data set, strain formulation.</td>
</tr>
<tr>
<td>RFSTRP(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing plastic strain component points (%), SP( ) or STRP( ), for the specific materials S/N data set, strain formulation.</td>
</tr>
<tr>
<td>RSTR(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing stress points (psi) with stress ratio = -1.0, or strain total points (%) for all S/N data sets.</td>
</tr>
<tr>
<td>S</td>
<td>RE</td>
<td>Value of stress S (psi) for which a value of fatigue cycles to failure is required.</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>SCLK</td>
<td>RE</td>
<td>Adjustment factor for BIGK( ) if TRSBN(0) &gt; SZERO.21</td>
</tr>
<tr>
<td>SE(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the user-provided elastic strain components SE( %).</td>
</tr>
<tr>
<td>SIG(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution standard deviation of the materials model shape parameter m for each life region of the truncated Normal S/N curve.</td>
</tr>
<tr>
<td>SIG2E(MAXREG)</td>
<td>RE</td>
<td>1-D array containing value of V (In N</td>
</tr>
<tr>
<td>SIG2P(MAXREG)</td>
<td>RE</td>
<td>1-D array containing value of V (In N</td>
</tr>
<tr>
<td>SIGE(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution standard deviation of the materials model shape parameter m_E for the elastic strain components.</td>
</tr>
<tr>
<td>SIGMA2(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of V (In N</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SIGP(MAXREG)</td>
<td>RE</td>
<td>1-D array containing the posterior Normal distribution standard deviation** of the materials model shape parameter ( m_p ) for the plastic strain components.</td>
</tr>
<tr>
<td>SP(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the user-provided plastic strain components ( S_p ) (%) .</td>
</tr>
<tr>
<td>1) STR(MAXDAT, MAXREG)</td>
<td>RE</td>
<td>2-D array containing stress points with stress ratio ( \equiv -1.0 ), for the specific material S/N data set partitioned into life regions, stress formulation.</td>
</tr>
<tr>
<td>2) STR(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing the user-provided stress data (psi) or total strain data (%) for all S/N data sets.</td>
</tr>
<tr>
<td>3) STR</td>
<td>RE</td>
<td>Value of total strain ( S ) (%) for which a value of fatigue cycles to failure is required.</td>
</tr>
<tr>
<td>STRAIN</td>
<td>RE</td>
<td>Value of total strain ( S ) (%) for which fatigue cycles to failure is desired.</td>
</tr>
<tr>
<td>STRE(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the user-provided and the calculated elastic strain components ( S_E ) (%).</td>
</tr>
<tr>
<td>STRESS</td>
<td>RE</td>
<td>Value of stress (psi) for which fatigue cycles to failure is desired.</td>
</tr>
<tr>
<td>STRP(MAXDAT, 0:MAXSET)</td>
<td>RE</td>
<td>2-D array containing values of the user-provided and the calculated plastic strain components ( S_p ) (%).</td>
</tr>
<tr>
<td>SUHAT</td>
<td>RE</td>
<td>Value of SUHAT2(L) ** 0.5 for a set of calculations.</td>
</tr>
<tr>
<td>SUHAT2(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the residual variances ( S_u^2 ), Equation 2-21, from the X on Y regression for each life region, where ( X = \ln S, Y = \ln N ).</td>
</tr>
<tr>
<td>SUM</td>
<td>RE</td>
<td>Value of ( \Sigma(PSIG2(L) \times NUM(L)) ), a weighted sum of the PSIG2's, used to calculate a weighted average to obtain ( \sigma^2 ) in Equation 2-49.</td>
</tr>
<tr>
<td>SUMX2</td>
<td>RE</td>
<td>Value of ( (NPPR(L) \times SX2(L))^{0.5} ) for a set of calculations.</td>
</tr>
<tr>
<td>SWHAT</td>
<td>RE</td>
<td>Value of SWHAT2(L) ** 0.5 for a set of calculations.</td>
</tr>
<tr>
<td>SWHAT2(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the residual variances ( S_w^2 ), Equation 2-20, from the Y on X regression for each life region, where ( X = \ln S, Y = \ln N ).</td>
</tr>
<tr>
<td>SX</td>
<td>RE</td>
<td>Value of ( (NPPR(L) \times SX2(L))^{0.5} ) for a set of calculations.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>SX2(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the sample variance $S_X^2$, Equation 2-20, for each life region, where $X = \ln S$.</td>
</tr>
<tr>
<td>SXY(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the sample covariance $S_{XY}$, Equation 2-20, for each life region, where $X = \ln S$, $Y = \ln N$.</td>
</tr>
<tr>
<td>SY2(MAXREG)</td>
<td>RE</td>
<td>1-D array containing values of the sample variance $S_Y^2$, Equation 2-21, for each life region, where $Y = \ln N$.</td>
</tr>
<tr>
<td>SZ</td>
<td>RE</td>
<td>Value of $SZ^2 \times 0.5$.</td>
</tr>
<tr>
<td>SZ2</td>
<td>RE</td>
<td>$S_Z^2$ in Equation 2-42, the sample variance of the transformed data of Equations 2-39 through 2-41.</td>
</tr>
<tr>
<td>SZZERO</td>
<td>RE</td>
<td>Stress tensile test point, $S_0$ (psi).</td>
</tr>
<tr>
<td>T</td>
<td>RE</td>
<td>Value of $T025(\ )$ to be used in calculations for Equation 2-26.</td>
</tr>
<tr>
<td>T025(TTAB)</td>
<td>RE</td>
<td>Table of 0.025 percentage points for the Student's $t$-distribution.</td>
</tr>
<tr>
<td>1) TEMP</td>
<td>RE</td>
<td>Temporary variable used to prevent arithmetic underflows and overflows during fatigue life calculation.</td>
</tr>
<tr>
<td>2) TEMP</td>
<td>RE</td>
<td>Temporary variable used during insertion sort.</td>
</tr>
<tr>
<td>TEST</td>
<td>RE</td>
<td>Value of $K_1 S_{max} (1 - R) / 2$, to be compared with $FTY$ during the stress ratio transformation calculations. See Section 4.1.3.3 for a discussion of the stress ratio transformation calculations.</td>
</tr>
<tr>
<td>TESTM</td>
<td>RE</td>
<td>Upper bound of the posterior credibility range on $m$ in region $(L - 1)$, used during concavity adjustment calculations.</td>
</tr>
<tr>
<td>TNSILE(0:MAXSET, MAXTNS)</td>
<td>RE</td>
<td>2-D array containing tensile test data points for each S/N data set, strain formulation.</td>
</tr>
<tr>
<td>TOTAL</td>
<td>INT</td>
<td>Sum of NUM(\ ) over all life regions.</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 $S_0$.</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 &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 $S_0$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>TTAB</td>
<td>INT</td>
<td>Maximum number of degrees of freedom in T025(). The maximum number is 31.</td>
</tr>
<tr>
<td>UPPER</td>
<td>RE</td>
<td>The upper bound of the intersection to find $\pi(m)$ in a particular region. 9</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; 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>1) X</td>
<td>RE</td>
<td>Uniform(0, 1) random variate used to obtain a randomly selected value for $m$.</td>
</tr>
<tr>
<td>2) X</td>
<td>RE</td>
<td>Normal($\mu$, $\sigma^2$) random variate used to obtain a randomly selected value for $m$.</td>
</tr>
<tr>
<td>3) X</td>
<td>RE</td>
<td>Value of the root of the equation $f(x) = 0$ for a particular iteration of the Newton's method calculation.</td>
</tr>
<tr>
<td>XST</td>
<td>RE</td>
<td>Value of the initial guess of the root $x$ for the Newton's method calculations.</td>
</tr>
<tr>
<td>Z</td>
<td>RE</td>
<td>$Z$ in Equation 2-48, 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. 21</td>
</tr>
<tr>
<td>ZZ(MAXDAT)</td>
<td>RE</td>
<td>$Z$ in the data transformation of Equations 2-39 through 2-41.</td>
</tr>
</tbody>
</table>

1. The need for saving $m$'s is discussed on Page 2-15.
2. See variable BZERO for a description of $\beta_o$. 
4. The median S/N curve for the truncated Normal case is discussed on Page 2-15.
5. Use definition 3 of variables ME and MP.
6. Use definition 2 of variables KE and KP.
7. See variable K, definition 2, for a description of $k$.
8. $\lambda_N$ is discussed on Pages 2-21 and 2-22.
9. The procedure for obtaining $\pi(m)$ is discussed on Page 2-13.
10 The median \( m \) calculation for the Uniform case is discussed on Pages 2-13 through 2-14.

11 The median S/N curve for the Uniform case is discussed on Pages 2-13 through 2-14 and for the truncated Normal case is discussed on Page 2-15.

12 See Section 2.1.2.3 for a discussion of process variation in materials.

13 Use definition 2 of variables \( ME \) and \( MP \).

14 \( m \) of the posterior density of \( m \) is discussed on Page 2-14.

15 \( m_E \) of the posterior density of \( m_E \) is discussed on Page 2-14.

16 \( m_p \) of the posterior density of \( m_p \) is discussed on Page 2-14.

17 Extension of the S/N curve to the right is discussed on Page 2-17.

18 The posterior credibility ranges \( \pi(m) \) are discussed on Page 2-13.

19 The posterior credibility range \( \pi(m_E) \) is discussed on Page 2-13.

20 The posterior credibility range \( \pi(m_p) \) is discussed on Page 2-13.

21 Extension of the S/N curve to the left using the tensile point is discussed on Page 2-17.

22 \( \sigma \) of the posterior density of \( m \) is discussed on Page 2-14.

23 \( \sigma_E \) of the posterior density of \( m_E \) is discussed on Page 2-14.

24 \( \sigma_p \) of the posterior density of \( m_p \) is discussed on Page 2-14.

25 Use definition 3 for variable \( NUM() \).

26 Concavity constraints are discussed on Pages 2-13 through 2-14.
<table>
<thead>
<tr>
<th>ALPHAG</th>
<th>ALPHB</th>
<th>ALPHC</th>
<th>ALPHD</th>
<th>ALPHE</th>
<th>ALPHF</th>
<th>ALPHG</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDREG</td>
<td>ADDRGN</td>
<td>ADJSTM</td>
<td>CONCAV</td>
<td>CONVRT</td>
<td>DECOMP</td>
<td>EXPCTD</td>
</tr>
<tr>
<td>FCT</td>
<td>FINDK</td>
<td>FINDM</td>
<td>FINDMC</td>
<td>FINDMN</td>
<td>FINDSB</td>
<td>FNDMSG</td>
</tr>
<tr>
<td>INIT</td>
<td>INITD</td>
<td>INTRVL</td>
<td>KBETA</td>
<td>KOMO</td>
<td>MATCHR</td>
<td>MEDIAN</td>
</tr>
<tr>
<td>MUSIG</td>
<td>NEWTON</td>
<td>NORRNG</td>
<td>PAREST</td>
<td>PECOMP</td>
<td>PREP</td>
<td>RCE</td>
</tr>
<tr>
<td>RDECHO</td>
<td>SMNVAR</td>
<td>SORTM</td>
<td>SW2SU2</td>
<td>TRNSFM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 7-9 Routine/Variable Chart**

| ALPHA  | ARG1  | ARG2  | B1    | B2    | B3    | B4    | B5    | B6    | B7    | B8    | C1    | C2    | C3    | C4    | C5    | C6    | D1    | D2    | D3    | D4    | D5    | D6    | EPS   |
|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| ADDREG | ARG1  | ARG2  | B1    | B2    | B3    | B4    | B5    | B6    | B7    | B8    | C1    | C2    | C3    | C4    | C5    | C6    | D1    | D2    | D3    | D4    | D5    | D6    | EPS   |

Example: For the routine ADDREG, the variables appear in the columns specified.
Table 7-9  Routine/Variable Chart (Cont’d)

|               | F FACTR | F FACT | F FCTR | F TRUZ | F FTYZ | F GAMMA | F GETLIFE | F GETLIFE2 | I END | I IER | I IER | I INC | I IORD | I INORD | I INVME | I IOUT | I IZERO | J JZERO | K KPROD | K KPROD | K KPROD |
|---------------|---------|--------|--------|--------|--------|---------|-----------|------------|-------|-------|-------|-------|--------|----------|----------|--------|---------|---------|--------|--------|
| ADDRREG       |   X     |   X    |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| ADDRGN        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| ADJSTM        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| CONCAV        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| CONVRT        |   X     |   X    |   1    |   X    |   2    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| DECOMP        |   X     |   X    |   1    |   X    |   X    |   X     |   X       |            |       |       |       |       |        |          |          |        |         |         |        |        |
| EXPCTD        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| FCT           |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| FINDK         |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| FINDM         |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| FINDMC        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| FINDMN        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| FINDSB        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| FNDRNG        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| GTLIFE        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| GTLIFEJ       |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| GTLIFE2       |   X     |   2    |   X    |   X    |   X    |   X     |   2       |   2         |       |       |       |       |        |          |          |        |         |         |        |        |
| GTPVAR        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| INFAGG        |   X     |   X    |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| INIT          |   1     |   1    |   1    |   1    |   1    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| INITD         |   1     |   1    |   1    |   1    |   1    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| INTRVL        |   2     |   2    |   2    |   2    |   2    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| KBETA         |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| KOMO          |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| MATCHR        |   X     |   X    |   X    |   X    |   X    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| MEDIAN        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| MUSIG         |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| NEWTON        |   X     |   X    |   X    |   X    |   X    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| NORRNG        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| PAREST        |   X     |   2    |   2    |   2    |   2    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| PECOMP        |   1     |   1    |   1    |   1    |   1    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| PREP          |   1     |   1    |   1    |   1    |   1    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| RCE           |   X     |   X    |   X    |   X    |   X    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| RDECHO        |   1     |   1    |   1    |   1    |   1    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| SMNVAR        |   1     |   X    |   X    |   X    |   X    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| SORTM         |   3     |   X    |   X    |   X    |   X    |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| SW2SU2        |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |
| TRANSFM       |         |         |        |        |        |         |           |            |       |       |       |       |        |          |          |        |         |         |        |        |

7 - 353
<table>
<thead>
<tr>
<th>Column1</th>
<th>Column2</th>
<th>Column3</th>
<th>Column4</th>
<th>Column5</th>
<th>Column6</th>
<th>Column7</th>
<th>Column8</th>
<th>Column9</th>
<th>Column10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRREG</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDRGN</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADJSTM</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONCAV</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVRT</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECOMP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>EXPECTD</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCT</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>FINDK</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDM</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDMC</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDMN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDSB</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FNDNRG</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTLIFEJ</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTPVAR</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>INFAGG</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>INIT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INITD</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>INTRVL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>KBETA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KOMO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCHR</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUSIG</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NEWTON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>NORRNG</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>PAREST</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PECOMP</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>PREP</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCE</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>RDECHO</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORTM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>X</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

7 - 354
|         | MEANX | MEANY | MEANZ | MEANZD | MEDME | MEDME2 | MEDME3 | MEDME4 | MIN | MME | MME2 | MME3 | MME4 | MINP | MOP | MP | MPNT | MPTTE | MPTEP2 | MPTEC | MTERM | MUEP | MEUP | MEUER | MZEREO | MZEREO2 | MZEREO3 | MZEREO4 | NBEND | NCGOPS | NDOIV | NFME | NMED |
|---------|-------|-------|-------|--------|-------|--------|--------|--------|-----|-----|------|------|------|------|-----|----|------|-------|--------|--------|--------|-------|------|-------|--------|--------|--------|--------|-------|-------|-------|
| ADDREG  |       |       |       |        |       |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| ADDRGN  |       |       |       |        |       |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| ADJSTM  | X     |       |       |        |       |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| CONCAV  |       |       |       |        |       |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| CONVRT  |       |       |       |        |       |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| DECOMP  |       | X     | X     |        |       |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| EXPCTD  | X     | X     | X     |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| FCT     |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| FINDK   |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| FINDM   |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| FINDMC  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| FINDMN  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| FINDSB  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| FNDRNG  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| GTLIFE  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| GTLIFEJ |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| GTLF2   |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| GTPVAR  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| INFA GG |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| INIT    |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| INITD   |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| INTRVL  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| KBETA   |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| KOMO    |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| MATCHR  | X     | X     | X     |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| MEDIAN  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| MUSIG   | X     |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| NEWTON  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| NORRNG  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| PAREST  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| PECOMP  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| PREP    |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| RCE     |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| RDECHO  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| SMNVAR  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| SORTM   |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| SW2SU2  | X     |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |
| TRNSFM  |       |       |       |        |        |        |        |        |     |     |      |      |      |      |     |    |      |        |         |         |         |       |      |       |        |        |        |        |       |      |       |

Table 7-9 Routine/Variable Chart (Cont'd)
<p>| Routine/Variable Chart (Cont’d) | ADDRREG | ADDRGN | ADJSTM | CONCAV | CONVRT | DECOMP | EXPCTD | FCT  | FINDK | FINDM | FINDMC | FINDMN | FINDSB | FINDRNG | GTUFE | GTUFEJ | GTUF2 | GTPVAR | INFAGG | INIT  | INITD | INTRVL | KBETA | KOMO | MATCHCHR | MEDIAN | MUSIG | NEWMON | NORRNG | PAREST | PECOMP | PREP  | RCE   | RDECHO | SMNVAR | SORTM | SW2SU2 | TRNSFMM |
|-------------------------------|---------|--------|--------|--------|--------|--------|--------|------|-------|-------|--------|--------|--------|---------|-------|-------|-------|--------|-------|------|-------|--------|-------|-----|----------|--------|-------|--------|-------|--------|-------|-------|------|-------|
| ADDRREG                       | X       |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| ADDRGN                        | X       |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| ADJSTM                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| CONCAV                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| CONVRT                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| DECOMP                        | 1 X 2 X X | X X X X |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| EXPCTD                        | 2 1 X  |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| FCT                           |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| FINDK                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| FINDM                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| FINDMC                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| FINDMN                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| FINDSB                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| FINDRNG                       |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| GTUFE                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| GTUFEJ                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| GTUF2                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| GTPVAR                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| INFAGG                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| INIT                          |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| INITD                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| INTRVL                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| KBETA                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| KOMO                          |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| MATCHCHR                      |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| MEDIAN                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| MUSIG                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| NEWMON                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| NORRNG                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| PAREST                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| PECOMP                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| PREP                          |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| RCE                           |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| RDECHO                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| SMNVAR                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| SORTM                         |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| SW2SU2                        |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |
| TRNSFMM                       |         |        |        |        |        |        |        |      |       |       |        |        |        |         |       |       |       |        |       |     |        |        |       |      |        |        |        |       |       |      |       |</p>
<table>
<thead>
<tr>
<th></th>
<th>RFSTRE</th>
<th>RFSHP</th>
<th>RSTR</th>
<th>RSND</th>
<th>SCLK</th>
<th>SIG1</th>
<th>SIG2P</th>
<th>SIGMA2</th>
<th>SIGP</th>
<th>STRN</th>
<th>STRS</th>
<th>STRP</th>
<th>SUMH</th>
<th>SUMX</th>
<th>SUMHAT</th>
<th>SXX</th>
<th>SXY2</th>
<th>SZZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDREG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDRGN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADJSTM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONCAV</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONVERT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DECOMP</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXPCTD</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FCT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDMC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDMN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FINDSB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FNDRNG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTLIFE</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTLIFEJ</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTLIFE2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GTPVAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INFAGG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INITD</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTRVL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KBETA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KOMO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATCHR</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MEDIAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUSIG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEWTON</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NORRNG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAREST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PECOMP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREP</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDECHO</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SMNVAR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SORTM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SW2SU2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRNSF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 7-9  Routine/Variable Chart (Cont'd)*
### 7.3.1.4 Program MATCHR Listing

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program MATCHR Listing Temporal Order, Stress Formulation, Uniform Distribution</td>
<td>7-361</td>
</tr>
<tr>
<td>Program MATCHR Listing Temporal Order, Stress Formulation, Truncated Normal Distribution</td>
<td>7-362</td>
</tr>
<tr>
<td>Program MATCHR Listing Temporal Order, Strain Formulation, Uniform Distribution</td>
<td>7-363</td>
</tr>
<tr>
<td>Program MATCHR Listing Temporal Order, Strain Formulation, Truncated Normal Distribution</td>
<td>7-365</td>
</tr>
<tr>
<td>MATCHR</td>
<td>7-367</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>7-372</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-372</td>
</tr>
<tr>
<td>INIT</td>
<td>7-377</td>
</tr>
<tr>
<td>RCE</td>
<td>7-379</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-386</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-389</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-391</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-394</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-396</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-399</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-401</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-402</td>
</tr>
<tr>
<td>EXPCRD</td>
<td>7-403</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-405</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-407</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-409</td>
</tr>
<tr>
<td>DECOMP</td>
<td>7-411</td>
</tr>
<tr>
<td>INITD</td>
<td>7-418</td>
</tr>
<tr>
<td>RDECHO</td>
<td>7-421</td>
</tr>
<tr>
<td>PREP</td>
<td>7-425</td>
</tr>
<tr>
<td>PECOMP</td>
<td>7-427</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-430</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-432</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>FINDMN</td>
<td>7-434</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-436</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>ADJSTM</td>
<td>7-442</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-443</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-444</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-445</td>
</tr>
<tr>
<td>GTLIF2</td>
<td>7-446</td>
</tr>
<tr>
<td>FCT</td>
<td>7-447</td>
</tr>
<tr>
<td>NEWTON</td>
<td>7-449</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-450</td>
</tr>
</tbody>
</table>

MATCHR Version 8.5
Program MATCHR Listing Temporal Order, Stress Formulation, Uniform Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCHR</td>
<td>7-367</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-372</td>
</tr>
<tr>
<td>INIT</td>
<td>7-377</td>
</tr>
<tr>
<td>RCE</td>
<td>7-379</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-386</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-389</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-391</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-394</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-396</td>
</tr>
<tr>
<td>ADDRGT</td>
<td>7-399</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-401</td>
</tr>
<tr>
<td>MEDIAN</td>
<td>7-402</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-407</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-408</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-410</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-411</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-414</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-430</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-432</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-433</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-437</td>
</tr>
<tr>
<td>TRANSFM</td>
<td>7-438</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-439</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-441</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-443</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-436</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-443</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-444</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-445</td>
</tr>
</tbody>
</table>
Program MATCHR Listing Temporal Order, Stress Formulation, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCHR</td>
<td>7-367</td>
</tr>
<tr>
<td>INFAGG</td>
<td>7-372</td>
</tr>
<tr>
<td>INIT</td>
<td>7-377</td>
</tr>
<tr>
<td>RCE</td>
<td>7-379</td>
</tr>
<tr>
<td>CONVRT</td>
<td>7-386</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>FINDMC</td>
<td>7-389</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-405</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-394</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-407</td>
</tr>
<tr>
<td>ADDRGN</td>
<td>7-409</td>
</tr>
<tr>
<td>CONCAV</td>
<td>7-401</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-430</td>
</tr>
<tr>
<td>FINDMNN</td>
<td>7-434</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-436</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-437</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-438</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-439</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-441</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-436</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-433</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-443</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-433</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-444</td>
</tr>
<tr>
<td>KOMO</td>
<td>7-445</td>
</tr>
<tr>
<td>GTLIFE</td>
<td>7-450</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-403</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-437</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-438</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-439</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-441</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-444</td>
</tr>
</tbody>
</table>
Program MATCHR Listing Temporal Order, Strain Formulation, Uniform Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCHR</td>
<td>7-367</td>
</tr>
<tr>
<td>DECOMP</td>
<td>7-411</td>
</tr>
<tr>
<td>INITD</td>
<td>7-418</td>
</tr>
<tr>
<td>REDCHO</td>
<td>7-421</td>
</tr>
<tr>
<td>PREP</td>
<td>7-425</td>
</tr>
</tbody>
</table>

(Find \( K_p, m_p \) for given plastic components)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-391</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
</tbody>
</table>

(Find \( K_E, m_E \) for given elastic components)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-391</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
</tbody>
</table>

(Begin calculations on estimated components)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PECOMP</td>
<td>7-427</td>
</tr>
<tr>
<td>PREP</td>
<td>7-425</td>
</tr>
</tbody>
</table>

(Find \( K_p, m_p \) for estimated plastic components)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-391</td>
</tr>
<tr>
<td>FNDRNG</td>
<td>7-396</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
</tbody>
</table>

(Find \( K_E, m_E \) for estimated elastic components)
<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-394</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>PAREST</td>
<td>7-430</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-432</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>ADJSTM</td>
<td>7-442</td>
</tr>
</tbody>
</table>

(Perform parameter estimation for plastic components)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>PAREST</td>
<td>7-430</td>
</tr>
<tr>
<td>FINDM</td>
<td>7-432</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
</tbody>
</table>

(Perform parameter estimation for elastic components)

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEIBGN</td>
<td>7-443</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-436</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
</tbody>
</table>

(Pick \( \phi, Z \))

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GTLIF2</td>
<td>7-446</td>
</tr>
<tr>
<td>NEWTON</td>
<td>7-449</td>
</tr>
<tr>
<td>FCT</td>
<td>7-447</td>
</tr>
</tbody>
</table>
Program MATCHR Listing Temporal Order, Strain Formulation, Truncated Normal Distribution

<table>
<thead>
<tr>
<th>Routine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATCHR</td>
<td>7-367</td>
</tr>
<tr>
<td>DECOMP</td>
<td>7-411</td>
</tr>
<tr>
<td>INITD</td>
<td>7-418</td>
</tr>
<tr>
<td>RDECHO</td>
<td>7-421</td>
</tr>
<tr>
<td>PREP</td>
<td>7-425</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-391</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>(Find $K_p$, $m_p$ for given plastic components)</td>
<td></td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>INTRVL</td>
<td>7-391</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>(Begin calculations on estimated components)</td>
<td></td>
</tr>
<tr>
<td>PECOMP</td>
<td>7-427</td>
</tr>
<tr>
<td>PREP</td>
<td>7-425</td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-405</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-407</td>
</tr>
<tr>
<td>(Find $K_E$, $m_E$ for estimated elastic components)</td>
<td></td>
</tr>
<tr>
<td>SW2SU2</td>
<td>7-387</td>
</tr>
<tr>
<td>MUSIG</td>
<td>7-405</td>
</tr>
<tr>
<td>NORRNG</td>
<td>7-407</td>
</tr>
<tr>
<td>GTPVAR</td>
<td>7-394</td>
</tr>
<tr>
<td>(Perform parameter estimation for plastic components)</td>
<td></td>
</tr>
<tr>
<td>PAREST</td>
<td>7-430</td>
</tr>
<tr>
<td>FINDDMN</td>
<td>7-434</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-436</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>Routine</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>------</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>ADJSTM</td>
<td>7-442</td>
</tr>
<tr>
<td>PAREST</td>
<td>7-430</td>
</tr>
<tr>
<td>FINDDM</td>
<td>7-434</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-436</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>WEIBGN</td>
<td>7-443</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>NORMGN</td>
<td>7-436</td>
</tr>
<tr>
<td>RANDOM</td>
<td>7-433</td>
</tr>
<tr>
<td>GTLIF2</td>
<td>7-446</td>
</tr>
<tr>
<td>NEWTON</td>
<td>7-449</td>
</tr>
<tr>
<td>FCT</td>
<td>7-447</td>
</tr>
<tr>
<td>SORTM</td>
<td>7-450</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
<tr>
<td>EXPCTD</td>
<td>7-403</td>
</tr>
<tr>
<td>TRNSFM</td>
<td>7-437</td>
</tr>
<tr>
<td>SMNVAR</td>
<td>7-438</td>
</tr>
<tr>
<td>KBETA</td>
<td>7-439</td>
</tr>
<tr>
<td>FINDK</td>
<td>7-440</td>
</tr>
<tr>
<td>FINDSB</td>
<td>7-441</td>
</tr>
</tbody>
</table>

(Perform parameter estimation for elastic components)

(Pick φ, Z)

(Calculate life)

(Sort \( m_p \)'s and \( m_e \)'s)

(Calculate median plastic S/N parameters)

(Calculate median elastic S/N parameters)
PROGRAM MATCHR PERFORMS THE OVERALL CONTROL OF THE STAND-ALONE MATERIALS CHARACTERIZATION CALCULATIONS
PROGRAMMER: L. NEWLIN
DATE: JAug91
VERSION: 8.5

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

PROGRAM MATCHR

SUBPROGRAMS: TRNAT, INFAGG, PAREST, NORMGN, WEIBGN, KOMO, GTLIFE,
DECMP, ADJSTM, GTLIFE2, SORTh, EXPCTD
7: DUMP-NEW; 8: IOUTPR-NEW
NOTE: 5 & 6 ARE OPENED IN INFAGG OR DECOMP

IMPLICIT NONE

INTEGER MAXDAT, MAXMM, MAXREG
PARAMETER (MAXDAT = 50, MAXMM = 20001, MAXREG = 3)

INTEGER I, IOUT, L, MID, MPROC, NCOMPS, NMED, NPTS(MAXREG),
& NTRIAL, NUMREG, VARY, ZROREG

REAL ALLM(MAXMM, MAXREG), BIG(0:MAXREG), BIGK1,
& BIGK2(0:MAXREG), BIGKP(0:MAXREG), BZERO, BZEROE, BZEROP,
& FACTR, FTUZ, FTUZ, GTLIFE, GTLIFE, KSTAT, LIFE, LNZ,
& ME(0:MAXREG), MEBND(2, MAXREG), MDHM(MAXREG),
& NH(0:MAXREG), MP(0:MAXREG), MU(MAXREG), NUE(MAXREG),
& MUP(MAXREG), NBDV(0:MAXREG), NF(MAXDAT, MAXREG), PHI,
& PSIG, PVAR, RANGE(2, MAXREG), RANGS(2, MAXREG),
& RANGP2(2, MAXREG), RFSTRE(MAXDAT, MAXREG),
& RFSTP(MAXDAT, MAXREG), SEND(0:MAXREG), SIG(MAXREG),
& SIG(0:MAXREG), SIG(0:MAXREG), STR(MAXDAT, MAXREG), STRAIN,
& STRESS, SZERO, TRBIGK(0:MAXREG), TRSBN(0:MAXREG), Z

DOUBLE PRECISION RAND

LIST OF VARIABLES

ALLM() 2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
BIG() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR EACH REGION
BIGK1 EQUAL TO BIG(1) -- DUMMY PARAMETER FOR CALLS TO SUBROUTINE EXPCTD
BIGKE() 1-D ARRAY CONTAINING VALUE OF KE FOR ELASTIC COMPONENTS
BZERO VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
BZEROE VALUE OF WEIBULL PARAMETER, BETAoe, CHARACTERIZING ELASTIC COMPONENTS OF S/N DATA SET
BZEROP VALUE OF WEIBULL PARAMETER, BETAop, CHARACTERIZING PLASTIC COMPONENTS OF S/N DATA SET
FACTR IT IS FACTOR = PHI * KSTAT * Z
FTUZ ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTUZ YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
GTLIFE FUNCTION WHICH CALCULATES THE CYCLES TO FAILURE AT A GIVEN STRESS OR STRAIN WHEN NCOMPS = 1
II CONTROLS DO LOOP FOR EACH TRIAL
IOUT OUTPUT DUMP CONTROLLER
KSTAT RATIO OF K/K, CONSTANT OVER REGIONS AND COMPONENTS
LIFE CYCLES TO FAILURE AT A GIVEN STRESS OR STRAIN
LNZ NORMAL(0, PVAR) GENERATED RANDOM VARIATE
MAXDAT MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED

7 - 367
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXNM</td>
<td>Maximum number of M's to be sorted</td>
</tr>
<tr>
<td>MAXREG</td>
<td>Maximum number of regions allowed</td>
</tr>
<tr>
<td>MEB()</td>
<td>1-D array containing selected value of Me for elastic components</td>
</tr>
<tr>
<td>MEDM()</td>
<td>2-D array containing allowable range on Me, for elastic components; after selection of Mp</td>
</tr>
<tr>
<td>MID()</td>
<td>1-D array containing the median M for each region</td>
</tr>
<tr>
<td>MM()</td>
<td>Pointer to the median M values -- equal to half of S/N trial values</td>
</tr>
<tr>
<td>MP()</td>
<td>1-D array containing selected values of M for each region</td>
</tr>
<tr>
<td>MF()</td>
<td>1-D array containing selected value of Mp for plastic components</td>
</tr>
<tr>
<td>MPROC</td>
<td>Materials process variation -- controls materials process variation -- 0 -- no variation; 1 -- variation</td>
</tr>
<tr>
<td>MU()</td>
<td>1-D array containing values of the posterior normal distribution mean for each region</td>
</tr>
<tr>
<td>MUE()</td>
<td>1-D array containing the posterior normal distribution mean for elastic components</td>
</tr>
<tr>
<td>MUP()</td>
<td>1-D array containing the posterior normal distribution mean for plastic components</td>
</tr>
<tr>
<td>NBD()</td>
<td>1-D array containing upper bounds (cycles) for the numreg regions of interest for the specific (reference) material S/N data set</td>
</tr>
<tr>
<td>NCOMPS</td>
<td>Number of components -- 1 for stress and strain when decomposed data unavailable -- 2 for decomposed strain data</td>
</tr>
<tr>
<td>NF()</td>
<td>2-D array containing randn() (cycles to failure) for the specific material S-N data set broken into regions</td>
</tr>
<tr>
<td>NMED</td>
<td>Controls median calculation for the normal distribution case -- 0 -- no median calculation; 1 -- median calculation desired</td>
</tr>
<tr>
<td>NPTS()</td>
<td>1-D array containing the number of points per region for the specific (reference) material S/N data set</td>
</tr>
<tr>
<td>NTRIAL</td>
<td>Number of M's to be used to calculate median</td>
</tr>
<tr>
<td>NUMREG</td>
<td>Number of regions of interest</td>
</tr>
<tr>
<td>PHI</td>
<td>Weibull(Betao, Etao) generated random variable</td>
</tr>
<tr>
<td>PSIG</td>
<td>Equal to sqrt(PVAR)</td>
</tr>
<tr>
<td>PVAR</td>
<td>Characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information</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 for each region -- rangem(1,L) is the lower bound and rangem(2,L) is the upper bound</td>
</tr>
<tr>
<td>RANGME()</td>
<td>2-D array containing the posterior range on M for elastic components -- rangem(1,1) is the lower bound and rangem(2,1) is the upper bound</td>
</tr>
<tr>
<td>RANGMP()</td>
<td>2-D array containing the posterior range on Mp for plastic components -- rangem(1,1) is the lower bound and rangem(2,1) is the upper bound</td>
</tr>
<tr>
<td>RFSTRE()</td>
<td>2-D array containing elastic strain points (%) or STRE() for the specific material S/N data set</td>
</tr>
<tr>
<td>RFSTRP()</td>
<td>2-D array containing plastic strain points (%) or SP() for the specific material S/N data set</td>
</tr>
<tr>
<td>SBND()</td>
<td>1-D array containing the stress values (psi, r = -1.0) corresponding to the &quot;life boundary&quot; values for each region contained in nbnd()</td>
</tr>
<tr>
<td>SIG()</td>
<td>1-D array containing values of the posterior normal distribution standard deviation for each region</td>
</tr>
<tr>
<td>SIGE()</td>
<td>1-D array containing the posterior normal distribution standard deviation for elastic components</td>
</tr>
<tr>
<td>SIGP()</td>
<td>1-D array containing the posterior normal distribution standard deviation for plastic components</td>
</tr>
<tr>
<td>STR()</td>
<td>2-D array containing stress or strain points for the specific material S/N data set (stress ratio = -1.0)</td>
</tr>
<tr>
<td>STRAIN</td>
<td>Value of total strain (%) for which cycles to failure is desired</td>
</tr>
<tr>
<td>STRESS</td>
<td>Value of stress (psi) for which cycles to failure is desired</td>
</tr>
<tr>
<td>STZERO</td>
<td>Stress tensile test point, so</td>
</tr>
<tr>
<td>STBNG()</td>
<td>1-D array containing values of K, adjusted to bring curve down if sbnd(0) &gt; so for each trial</td>
</tr>
<tr>
<td>TRSBND()</td>
<td>1-D array containing the stress values (psi, r = -1.0) corresponding to the &quot;life boundary&quot; values for each region contained in nbnd() adjusted by variation parameters for each trial</td>
</tr>
<tr>
<td>VARY</td>
<td>Controls type of curve variation desired -- 0 -- no</td>
</tr>
</tbody>
</table>

7 - 368
VARIATION: 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
LOG-NORMAL(0, PVAR) RANDOM VARIATE
ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

OPEN(1, FILE = 'SPECFD', STATUS = 'OLD')
OPEN(3, FILE = 'SPECFO', STATUS = 'NEW')
OPEN(7, FILE = 'DUMP', STATUS = 'NEW')
OPEN(8, FILE = 'IOUTPR', STATUS = 'NEW')
WRITE(3, *) ' Copyright (C) 1990, California Institute of Technology. U.S. Government'
& WRITE(3, *) ' Sponsorship under NASA Contract ',
& 'NAS7-918 is acknowledged.'

READ(1, *) RAND
WRITE(8, *) ' RANDOM NUMBER SEED: ', RAND
READ(1, *) IOUT
WRITE(8, *) ' OUTPUT DUMP CONTROLLER: ', IOUT
READ(1, *) NCOMPS
WRITE(8, *) ' NUMBER OF COMPONENTS: ', NCOMPS
READ(1, *) VARY
WRITE(8, *) ' TYPE OF S/N VARIATION DESIRED '
WRITE(8, *) ' (0-NONE; 1-INTRINSIC; 2-UNIFORM; 3-NORMAL): ', VARY
READ(1, *) NMED
WRITE(8, *) ' MEDIAN CURVE FOR NORMAL TYPE '
WRITE(8, *) ' VARIATION DESIRED (0 - NO, 1 - YES): ', NMED
READ(1, *) MPROC
WRITE(8, *) ' MATERIALS PROCESS VARIATION DESIRED '
WRITE(8, *) ' (0 - NO, 1 - YES): ', MPROC

IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
WRITE(8, *) 'ERROR: INVALID TYPE OF S/N VARIATION DESIRED'
CALL TRMNAT
ENDIF
IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
WRITE(8, *) 'ERROR: INVALID RESPONSE TO NORMAL MEDIAN ',
& '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
NTRIAL = 1
IF ((VARY .EQ. 0) .AND. (NMED .EQ. 0)) NTRIAL = 2000

CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZROREG, NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC, KRATIO, PVAR)

7 - 369
IF (MPROC .EQ. 1) PSIG = SQRT (PVAR)

DO 100 II = 1, NTRIAL

CALL PAREST (VARY, RAMGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND, NBND, STR, BIGK, BZERO, MM, SBND)

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

CALL WEIBGN (BZERO, RAND, PHI)

IF (VARY .EQ. 0) PHI = 1.0
WRITE(7,*) 'PHI =', PHI
FACTR = PHI * KRATIO * Z

DO 25 L = ZROREG, NUMREG
TRSBND(L) = FACTR * SBND(L)
TRBIGK(L) = BIGK(L)
25 CONTINUE
TRSBND(0) = SBND(0)

IF (ZROREG .EQ. 0) THEN
CALL KOMO(SZERO, BIGK, 1414, NBND, TRSBND, TRBIGK, FACTR, NUMREG)
ENDIF

LIFE = GTLIFE (STRESS, MM, TRBIGK, PHI, KRATIO, LNZ, TRSBND, ZROREG, NUMREG, SZERO)

WRITE(8,*), 'NUMBER OF REGIONS=', NUMREG, 'PHI=', PHI
WRITE(8,*), 'REGION=', L, 'STRESS BOUND=', TRSBND(L)
WRITE(8,*), 'BIGK(L)', BIGK(L), 'MM(L)', MM(L)

50 CONTINUE
WRITE(8,*), 'STRESS=', STRESS, 'LIFE=', LIFE

DO 150 L = 1, NUMREG
ALLM(I,L) = MM(L)
150 CONTINUE

IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) THEN
CALL SORTM (ALLM, NUMREG, NTRIAL)
MID = NTRIAL / 2
DO 175 L = 1, NUMREG
MEDM(L) = ALLM(MID,L)
175 CONTINUE

CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND, BIGK1, BZERO)
ENDIF

ELSE IF (NCOMPS .EQ. 2) THEN
READ(1,*), STRAIN
WRITE(8,*), 'OF TOTAL STRAIN=', STRAIN

PERFORM CALCULATIONS ON DECOMPOSED STRAIN DATA

CALL DECOMP (RANGMP, RANGME, MUP, SIGP, MUE, SIGE, NF, NPTS, NBND, RFSTRP, RFSTRE, VARY, MPROC, KRATIO, PVAR)

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

DO 200 II = 1, NTRIAL

CALL PAREST (VARY, RANGMP, MUP, SIGP, NF, NPTS, 1, 1, RAND, NBND, RFSTRP, BIGKP, BZERO, MP, SBND)

CALL ADJSTM (MP, RANGME, MEBND)

CALL PAREST (VARY, MEBND, MUE, SIGE, NF, NPTS, 1, 1, RAND, NBND, RFSTRE, BIGKE, BZEROE, ME, SBND)

BZERO = 1.0 / SQRT (0.5 * ((1.0 / BZERO) ** 2 + (1.0 / BZEROE) ** 2))

CALL WEIBGN (BZERO, RAND, PHI)

IF (VARY .EQ. 0) PHI = 1.0

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

WRITE(7,*) 'PHI = ', PHI, ' Z = ', Z
LIFE = GTLIF2 (STRAIN, BIGKP(1), BIGKE(1), MP(1), ME(1), PHI, KRATIO, Z)

WRITE(8,*) 'KP: ', BIGKP(1), ' MP: ', MP(1)
WRITE(8,*) 'KE: ', BIGKE(1), ' ME: ', ME(1)
WRITE(8,*) 'PHI = ', PHI, ' Z: ', Z
WRITE(8,*) 'STRAIN: ', STRAIN, ' LIFE: ', LIFE

ALLM(II,1) = MP(1)
ALLM(II,2) = ME(1)

CONTINUE

200 CONTINUE

IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) THEN

CALL SORTM (ALLM, 2, NTRIAL)

MID = NTRIAL / 2

WRITE(7,*) 'MEDIAN PLASTIC S/N CURVE'
MEDM(1) = ALLM(MID,1)
CALL EXPCTD (2, MEDM, NPTS, RFSTRP, NF, 0.0, 1, 1, NBND, BIGKI, BZERO)

WRITE(7,*) 'MEDIAN ELASTIC S/N CURVE'
MEDM(1) = ALLM(MID,2)
CALL EXPCTD (2, MEDM, NPTS, RFSTRE, NF, 0.0, 1, 1, NBND, BIGKE, BZEROE)

BZERO = 1.0 / SQRT (0.5 * ((1.0 / BZERO) ** 2 + (1.0 / BZEROE) ** 2))

WRITE(7,*) 'TOTAL STRAIN E(BETAo) = ', BZERO

ENDIF

ELSE
WRITE(8,*) 'NCOMPS INCORRECTLY SPECIFIED'
CALL TRMNAT
ENDIF
STOP
END

C***************************************************************
SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
PROGRAMMER: L. NEWLIN
DATE: 5OC787
VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
V8.4, V8.5
SUBROUTINE TRMNAT
WRITE(8,*) 'PROGRAM EXECUTION TERMINATED'
STOP
END

C***************************************************************
SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
FOR THE STRESS FORMULATION
PROGRAMMER: L. NEWLIN
FORMAT/COMMENTS: 12AUG91
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, REFPN, SZERO, ZROREG,
& NUMREG, NBND, STR, FTUZ, FTYZ, VARY, MPROC,
& KRATIO, PVAR)
C INPUTS: READS DATA FROM SPECIFIED AND RELATED; VARY, MPROC
C OUTPUTS: RANGEM, MU, SIG, NF, REFPN, SZERO, ZROREG, NUMREG,
NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C SUBPROGRAMS: INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDRGN,
CONCAV, MEDIAN, EXPCTD, MUSIG, NORMNG, ADDRGN, GTPVAR
C FILES: 5:RELATED-OLD; 6:RELATED-NEW
C IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
& NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
& NSETS, NUMREG, REFPN(MAXREG), VARY, ZROREG
REAL BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
LIST OF VARIABLES

BIGNHT
BZERO
CZERO
DD()
DELTA()
FTUZ
FTY2
IOUT
IZERO()
JZERO()
KRATIO
LAMN
LNNF()
LNSTR()
MAXDAT
MAXREG
MAXSET
MC()
MCPNT()
MCHRAT()
MEDM()
MO()
MPART()
MPROC
MZERO()
NBND()
NF()
NNODAT
NP()
NPART()
NPTS()
NSETS
NUMREG
PVAR
RANGEM()
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

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)

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)

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

ZROREG ZER0 REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION

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

C RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET INFORMATION
C PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION
C INITIALIZE PRIMARY ARRAYS

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

C READ, CONVERT, ECHO INFORMATION

CALL RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP, LNSTR, & LNHF, REFNP, STR, NF, SZERO, ZROREG, NUMREG, MNODAT, & NSETS, NEND, CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, & SIGMA2, KRAI0, LAMN)

C CALCULATE M CONTRAINT BASED ON CO

CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNHF, SX2, SXY, SY2, DD, & SWHAT2, SUHAT2, NPPR)

C CALCULATE RESIDUAL VARIANCES

CALL SWM2U2 (NUMREG, NSETS, NP, LNSTR, LNHF, SX2, SXY, SY2, DD, & SWHAT2, SUHAT2, NPPR)

C CALCULATE M CONTRAINT BASED ON CO

CALL FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN

C CALCULATIONS FOR ALL TYPES OF VARIATION SAVE NORMAL

C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS

CALL INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, & JZERO, MCRES)
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, NNDAT, 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 (IZERO (L), L), IZERO (JZERO (L), L),
      &
      25    CONTINUE
   WRITE (7, 910)
   DO 50 L = 1, NUMREG
      WRITE (7, 915) L, MCHAT (2, L), MCHAT (1, L)
      50    CONTINUE
   IF (CZERO .GT. 0.0) THEN
      WRITE (7, 960)
      DO 150 L = 1, NUMREG
         IF (MCPNT (L) .EQ. 1) THEN
            WRITE (7, 965) L, MC (1, L)
         ELSEIF (MCPNT (L) .EQ. 2) THEN
            WRITE (7, 970) L, MC (1, L), MC (2, L)
         ENDIF
      150    CONTINUE
   ENDIF
   WRITE (7, 920)
   WRITE (7, 930)
   DO 100 L = 1, NUMREG
      WRITE (7, 940) L, RANGEM (1, L), RANGEM (2, L)
  100    CONTINUE
   WRITE (7, 950)
C CALCULATE MEDIAN M VALUES BASED ON DATA, MZERO, AND CZERO
   CALL MEDIAN (NUMREG, RANGEM, MEDM)
C CALCULATE ESTIMATED VALUES FOR S/N CURVE PARAMETERS
   CALL EXPCTD (1, MEDM, REFNP, STR, NF, SZERO, NUMREG, ZROREG, &
              BBND, BIKHIT, 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)
      200    CONTINUE
   ENDIF
ELSE
C NORMAL VARIATION IS DESIRED
C CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION
   CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPFR, 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)
   CONTINUE
   IF (CZERO .GT. 0.0) THEN
      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
      ENDIF
   ENDIF
   WRITE(7,990) L, MU(L), SIG(L)
   CONTINUE
   WRITE(7,995) PVAR
   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',
   376
& \text{\textcopyright (c) 1990, California Institute of Technology. U.S. Government sponsors under NASA Contract NAS7-918 as acknowledged. }
& \text{THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT}
& \text{MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION. }

\text{return}
\text{end}

\text{SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS }
\text{USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG }
\text{PROGRAMMER: L. NEWLIN }
\text{DATE: CODE: 21JUN88 COMMENTS: 13JUL89 }
\text{VERSION: MATCHR V8.1, V8.2, V8.3, V8.4, V8.5 }
\text{MATGRM V4.1, V4.2, V4.3, V4.4, V4.5 }

\text{SUBROUTINE INIT (NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR,}
& \text{REFNP, NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2) }
\text{INPUTS: --- }
\text{OUTPUTS: NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,}
\text{NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2 }
\text{IMPLICIT NONE }
\text{INTEGER MAXDAT, MAXREG, MAXSET }

7 - 377
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)

COMMON IOUT
INTEGER I, IOUT, J, K, L, NPNT(MAXREG), NP(0:MAXSET, MAXREG), & NPTS(0:MAXSET, MZERO(2, MAXREG)), NF(MAXDAT, MAXREG), & MZERO(2, MAXREG), NF(MAXDAT, MAXREG), & RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET), & SIGMA2(MAXREG), & STR(MAXDAT, MAXREG)

REAL DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG), & LNSTR(MAXDAT, 0:MAXSET, MAXREG), NO(MAXREG), & MZERO(2, MAXREG, NF(MAXDAT, MAXREG)) & RATSTR(MAXDAT, 0:MAXSET, RAWNF(MAXDAT, 0:MAXSET), SIGMA2(MAXREG), & STR(MAXDAT, MAXREG)

LIST OF VARIABLES

1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
I CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT OUTPUT DUMP CONTROLLER
K CONTROLS DO LOOP FOR EACH REGION
L CONTROLS DO LOOP FOR EACH POINT IN A REGION
LNNF() 3-D ARRAY CONTAINING LN(RAWNF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING LN(RATSTR()), ALSO INDEXED FOR REGION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
NP() 2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET IN EACH REGION
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
RATSTR() 2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS RATIO OR TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
RAWNF() 2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS
RAWSTR() 2-D ARRAY CONTAINING RAW STRESS DATA (PSI) OF TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
REFNF() 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
    RATSTR(I, J) = 0.0
350 CONTINUE
300 CONTINUE

DO 400 L = 1, MAXREG
  DO 425 K = 1, MAXDAT
    LOFN(K, J, L) = 0.0
425 CONTINUE
400 CONTINUE
```

 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 relatd. RCE also breaks s/n data sets into regions as specified by user.


implicit none

 integer maxdat, maxreg, maxset
 parameter (maxdat = 50, maxreg = 3, maxset = 5)

 common iout
 integer count, i, iout, j, k, l, m, mpt(maxreg), mproc, ndiv,
 & nnodat, np(0:maxset, maxreg), npfz(0:maxset, maxreg), nsets,
 & num, numreg, refnp(maxreg), rel, vary, nzreg

 real czero, deltamaxreg, ftu, ftuz, ftv, ftvz,
 & kratio, lamn, lnfn(maxdat, 0:maxset), lnstr(maxdat, maxreg), 
 & lnstr(maxdat, maxreg, 0:maxset, maxreg), mzero(maxreg), 
 & mzero(maxreg, 0:maxreg), nbd(0:maxreg), npfz(maxdat, maxreg), 
 & npfz(maxdat, 0:maxset), rawnf(maxdat, 0:maxset), 
 & rawstr(maxdat, 0:maxset), sigma2(maxreg), 
 & str(maxdat, maxreg), szero

 character*40 descrp(0:maxset)
```

7 - 379
LIST OF VARIABLES

COUNT INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO, CONVERSION, AND BREAK UP
CZERO EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE COEFFICIENT OF VARIATION, CO
DELTA() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND SIG() CALCULATION
DESCRP() 1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
FTU ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
FTUZ ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTY YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
FTYZ YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
IOUT CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
JOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
LAMN LAMBDA-N -- RATIO OF Var (ln N given S) / (m^2 c^2), CONSTANT OVER ALL REGIONS AND COMPONENTS
LNNF() 3-D ARRAY CONTAINING LN(RAWNFF()), ALSO INDEXED FOR REGION
LNSTR() 3-D ARRAY CONTAINING RAWFSTR(), ALSO INDEXED FOR REGION
M CONTROLS DO LOOP FOR EACH DATA DIVISION
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MO() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO() FOR EACH REGION
MPROC Materials PROCESS variation -- CONTROLS MATERIALS PROCESS VARIATION -- 0 - NO VARIATION; 1 - VARIATION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NDIV NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO, REGION PAIRS DURING INPUT
NF() 2-D ARRAY CONTAINING RAWF() (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)
NF() 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 STRESSES 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
RAINF() 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
REFP() 1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
REG REGION OF INTEREST IN A PARTICULAR DIVISION
SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR EACH REGION
STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
SZERO STRESS TENSILE TEST POINT, SO
VARY CONTROLS TYPE OF CURVE VARIATION DESIGNED -- 0 - NO VARIATION; 1 - S/N RANDOMNESS ONLY; 2 - UNIFORM VARIATION; 3 - TRUNCATED NORMAL VARIATION
ZROREG ZeRO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP BEGGINING 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(I,*) DESCRP(0), FTU, 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. I0) WRITE(8,900) DESCRP(0), FTU, FTU, NPTS(0)
WRITE(3,905)
IF (IOUT .EQ. I0) WRITE(8,905)
C STORE VALUES OF SPECIFIC MATERIAL FTU AND FTU INTO FTUZ AND FTYZ
FTUZ = FTU
FTYZ = FTU
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)
      RAWSTR(I,0) = RAWSTR(I,0)
   110  CONTINUE
C CHECK TO SEE IF STRESS RATIO IS -1.0 AND CONVERT STRESSES IF NOT
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, FTU)
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
IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RANNF(I,0), RATIO, REG, RATSTR(I,0), RANNF(I,0)

130 CONTINUE
C
BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2, EXPTD, AND PAREST
C
K = NP(0,REG)
DO 140 I = (COUNT + i), (COUNT + NUM)
   K = K + 1
   LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
   LNNF(K,0,REG) = ALOG(RANNF(I,0))
   STR(K,REG) = RATSTR(I,0)
   NF(K,REG) = RANNF(I,0)
140 CONTINUE
IF (K .GT. MAXDAT) THEN
   WRITE(8,*), 'ERROR: OVER NUMBER OF POINTS LIMIT IN ', 'SPECIFIC MATERIAL'
   CALL TRNNAT
   ENDIF
   NP(0,REG) = K
   REFPN(REG) = K
   COUNT = COUNT + NUM
100 CONTINUE
IF (NPTS(0) .NE. COUNT) THEN
   WRITE(8,*), 'ERROR: NUMBER OF POINTS PER DIVISION ', 'INCORRECTLY SPECIFIED'
   WRITE(8,*), 'IN SPECIFIC DATA SET'
   CALL TRNNAT
   ENDIF
READ(1,*), SZERO
IF (NINT(SZERO) .GT. 0) THEN
   ZROREG = 0
ELSE
   ZROREG = 1
ENDIF
IF (IOUT .EQ. 10) THEN
   WRITE(8,*), 'SZERO = ', 'ZZERO = ', 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 TRNNAT
ENDIF
DO 150 L = ZROREG, (NUMREG + NNODAT)
READ(1,*), NBND(L)
150 CONTINUE
READ(1,*), CZERO
DO 160 L = 1, (NUMREG + NNODAT)
   READ(1,*), MPNT(L), MZERO(1,L), MZERO(2,L)
160 CONTINUE
WRITE(3,913)
IF (ZROREG .EQ. 0) WRITE(3,914) 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 = 1, (NUMREG + NNODAT)  
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 ',  
& 'RANGE ON M'  
& CALL TRMNAT  
180 CONTINUE  
IF (VARY .EQ. 3) THEN  
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,*), '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)  
7 - 383
WRITE(6,905)
IF (IOUT .EQ. 10) WRITE(8,905)
DO 300 M = 1, NDIV
   READ(5,*) NUM, RATIO, REG
   IF (ABS(RATIO) .GT. 1.0) THEN
      WRITE(8,*) 'ERROR: INVALID VALUE OF RATIO: ', RATIO
      CALL TRMNAT
   ENDIF
   IF (REG .GT. MAXREG) THEN
      WRITE(8,*) 'ERROR: OVER REGION LIMIT IN RELATED MATERIAL ', J
      CALL TRMNAT
   ENDIF
   IF (IOUT .EQ. 10) THEN
      WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
      WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
   ENDIF
   DO 310 I = (COUNT + 1), (COUNT + NUM)
      READ(5,*), RAWSTR(I,J), RAWNF(I,J)
   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)
      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)
   CONTINUE
330 CONTINUE
   K = NP(J,REG)
   DO 340 I = (COUNT + 1), (COUNT + NUM)
      K = K + 1
      LNSTR(K,J,REG) = ALOG(RATSTR(I,J))
      LNHF(K,J,REG) = ALOG(RAWNF(I,J))
   CONTINUE
340 CONTINUE
   IF (K .GT. MAXDAT) THEN
      WRITE(8,*), 'ERROR: OVER LIMIT ON NUMBER OF POINTS ', 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',18X,E11.5,///,2X,'ULTIMATE STRENGTH',
            & 15X,E11.5,///,2X,'NUMBER OF POINTS',16X,I2)

905  FORMAT(///,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
            & 5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
            & 'STRESS',7X,'LIFE'/)

910  FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

913  FORMAT(///)

914  FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN SO OF',
            & 5X,E11.5)

915  FORMAT(2X,'THERE Is ',I2,' REGION(S) WITH DATA' ,
            & 2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
            & 2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
            & '(CYCLES): ','/)

920  FORMAT(10X,E10.3)

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,I1,11X,I1,12X,F7.4,9X,F7.4)

935  FORMAT(2X,'NUMBER OF DATA SETS:',2X,I2,///,17X,
            & 'NOTE: ALL K ASSUMED TO BE 1.0',///,23X,
            & 'TRANSFORMED DATA')

940  FORMAT(///,2X,'DESCRIPTION:',2X,A40,  
            & ///,2X,'YIELD STRENGTH',18X,F7.0,  
            & ///,2X,'ULTIMATE STRENGTH',15X,F7.0,  
            & ///,2X,'NUMBER OF POINTS',16X,I2)

945  FORMAT(///,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS:',
            & ///,2X,'REGION',5X,'DELTA',8X,'MO',10X,'SIGMA',/)

950  FORMAT(5X,I1,5X,F7.2,5X,F7.4,5X,E11.5)

955  FORMAT(///,2X,'MATERIALS PROCESS VARIATION INFORMATION',
            & ///,2X,'MEDK*/MEDK:',5X,E11.5,///,5X,'LAMBDAN:',5X,E11.5)

RETURN
END

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

C THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C STRESS RATIO, R, IS NOT -1.0
SUBROUTINE CONVRT (J, NUM1, NUM2, STR, RSTR, R, FTU, FTY)

INPUTS: J, NUM1, NUM2, STR, R, FTU, FTY

OUTPUTS: RSTR

IMPLICIT NONE

INTEGER MAXDAT, MAXSET
PARAMETER (MAXDAT = 50, MAXSET = 5)

COMMON IOUT
INTEGER I, IOUT, J, NUM1, NUM2

REAL FTU, FTY, R, RSTR(MAXDAT, 0:MAXSET), &
     STR(MAXDAT, 0:MAXSET), TEST

LIST OF VARIABLES

FTU  ULTIMATE STRENGTH OF MATERIAL (PSI)
FTY  YIELD STRENGTH OF MATERIAL (PSI)
IOUT OUTPUT DUMP CONTROLLER
J  DATA SET OF INTEREST
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
NUM1 FIRST INDEX TO BE TRANSFORMED
NUM2 LAST INDEX TO BE TRANSFORMED
R  STRESS RATIO (R = -1.0 IS DESIRED)
RSTR() STR() VALUES TRANSFORMED TO R = -1.0 (PSI)
STR() ARRAY CONTAINING STRESS VALUES (PSI) FOR S/N CURVE
TEST  Kt * Smax * (1 - R)/2 , TO BE COMPARED WITH FTY

C Kt IS ASSUMED TO BE ONE

DO 100 I = NUM1, NUM2
    TEST = STR(I,J) * (1.0 - R)/2.0
    IF (IOUT.EQ.10) WRITE(8,*)'I = ',I,' J = ',J,' TEST = ',TEST
    IF (TEST .GE. FTY) THEN
        RSTR(I,J) = TEST
        IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() = ',RSTR(I,J)
    ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN
        RSTR(I,J) = TEST/((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
SUBROUTINE SW2SU2 CALCULATES, SWHAT2, 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: 6OCT87

COMMENTS: 13JUL89


SUBROUTINE SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, SX2, SKY, SY2, DD, SWHAT2, SUHAT2, NPPR)

C INPUTS: NUMREG, NSETS, NP, LNSTR, LNNF
C OUTPUTS: SX2, SKY, SY2, DD, SWHAT2, SUHAT2, NPPR
C
C IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5)
COMMON 1OUT
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), SKY(MAXREG), SY2(MAXREG)

LIST OF VARIABLES
I-DARRAY 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(RAWNF(}), 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
2-D ARRAY CONTAINING VALUES OF ((SUM OF (NP())-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
1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y, COVARIANCE FOR REGIONS
C SY2() 1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
(Y = Ln N)

C INITIALIZE ARRAYS

DO 50 L = 1, MAXREG
SY2(L) = 0.0
SX2(L) = 0.0
SXY(L) = 0.0
SWHAT2(L) = 0.0
SUHAT2(L) = 0.0
BB(L) = 0.0
DD(L) = 0.0
NPPR(L) = 0
50 CONTINUE

DO 60 J = 0, MAXSET
DO 70 K = 1, MAXDAT
DIFFY(K,J) = 0.0
DIFFX(K,J) = 0.0
70 CONTINUE
MEANY(J) = 0.0
MEANX(J) = 0.0
60 CONTINUE

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

DO 100 L = 1, NUMREG

C 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,
' N = ', NP(J,L)

DO 250 K = 1, NP(J,L)
MEANY(J) = MEANY(J) + LNLF(K,J,L)
MEANX(J) = MEANX(J) + LNSTR(K,J,L)
250 CONTINUE
MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
IF (IOUT .EQ. 10) WRITE(8,*)'MEAN(Y(J) = ', MEANY(J),
' MEAN(X(J) = ', MEANX(J)

C NOW CALCULATE SAMPLE VARIANCES, SY2, SX2 AND SXY,
OF Y AND Y FOR EACH REGION BY SUMMING OVER EACH
DATA SET IN REGION L

DO 300 K = 1, 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)
300 CONTINUE
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

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
C 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))
SX(L) = SX(L) / FLOAT(NPPR(L))

C NOW CALCULATE THE RESIDUAL VARIANCES, SWHAT2, SUHAT2, FOR EACH
C 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,*) 'NP(J,L) =', NP(J,L)
ENDIF
DO 400 J = 0, NSETS
IF (IOUT .EQ. 10) WRITE(8,*) 'J =', J, ' NP(J,L) =', NP(J,L)
DO 500 K = 1, NP(J,L)
SWHAT2(L) = SWHAT2(L)
& + (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. 10) WRITE(8,*) 'K =', K, ' 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. 10) WRITE(8,*) 'NPPR(L) =', NPPR(L),
& ' SWHAT2(L) =', SWHAT2(L), ' SUHAT2(L) =', SUHAT2(L)
100 CONTINUE
RETURN
END

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

C SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C THE CO GIVEN BY THE USER
C PROGRAMMER: L. NEWLIN
C DATE: 8OCT87   CODE: 13JUL89    COMMENTS: 13JUL89
C VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
V8.4, V8.5
C SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)
C INPUTS: NUMREG, CZERO, SX2, SXY, SY2
C OUTPUTS: MCPNT, MC
C IMPLICIT NONE

7 - 389
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
       (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
         C THEN NO M CONSTRAINT IS REQUIRED
         MCPNT(L) = 0
     ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN
         C THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M
         MCPNT(L) = 1
         MC(1,L) = - SY2(L) / (2.0 * SXY(L))
     ELSE
         C THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
         COMMON CALCULATIONS
C

7 - 390

C INPUTS: NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C OUTPUTS: IZERO, JZERO, MCHAT
C SUBPROGRAMS: TMRNAT

C IMPLICIT NONE
INTEGER CHITAB, MAXREG, TTAB
PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)

ARG2 = (SXY(L) ** 2 - SY2(L) * ARG1)
IF (ARG2 .LT. 0.0) THEN
C ARG2 IS NEGATIVE -- IMPLIES M IS COMPLEX
WRITE(8,*), 'ERROR: Co TOO LOW'
CALL TMRNAT
ELSE
ARG2 = ARG2 ** 0.5
ENDIF
IF (SX2(L) .LT. CZERO2) THEN
C AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M
MC = (1.
MC(1,L) = (- SXY(L) - ARG2) / ARG1
ELSE
C SX2(L) .GT. CZERO2 -- THIS TIME THE M CONSTRAINT IS A RANGE
MC = (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, ' MC = ', MC(L)
WRITE(8,*), 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
WRITE(8,*), 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
200 CONTINUE
ENDIF
RETURN
END
<table>
<thead>
<tr>
<th>I</th>
<th>OUT</th>
<th>INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG</th>
</tr>
</thead>
<tbody>
<tr>
<td>REAL</td>
<td>ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IZERO(2, MAXREG), MCHAT(2, MAXREG), SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SX2(MAXREG), T, TO25(TTAB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DATA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CHI025(I), I = 1, 75)</td>
<td>/</td>
</tr>
<tr>
<td>0.000982069, 0.506356, 0.215795, 0.484419, 0.831211, 1.237347, 1.68987, 2.17973, 2.70039, 3.24697,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.81575, 4.40379, 5.00874, 5.62872, 6.26214, 6.90766, 7.56418, 8.23075, 8.90655, 9.59083, 10.2829, 10.923, 11.568, 12.401, 13.415, 14.573, 15.307, 16.547, 17.708, 19.04, 19.80, 20.56, 21.33, 22.10, 22.87, 23.65, 24.31, 25.21, 26.99, 28.78, 29.52, 28.36, 29.18, 29.95, 30.75, 31.56, 32.37, 33.16, 33.96, 34.77, 35.58, 36.39, 37.21, 38.02, 38.84, 39.66, 40.48, 41.30, 42.12, 42.95, 43.77, 44.60, 45.43, 46.26, 47.09, 47.92, 48.75, 49.59, 50.42, 51.26, 52.10, 52.94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.78, 54.62, 55.46, 56.50, 57.15, 57.80, 58.84, 59.69, 60.54, 61.39, 62.24, 63.09, 63.94, 64.79, 65.64, 66.50, 67.35, 68.21, 69.07, 69.92, 70.78, 71.64, 72.50, 73.36, 74.22, 75.08, 75.94, 76.80, 77.66, 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, 89.83, 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, 115.33, 116.21, 117.10, 117.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.02389, 7.37776, 9.34840, 11.1433, 12.8325, 14.4494, 16.0128, 17.5346, 19.0228, 20.4831, 21.9200, 23.3367, 24.7596, 26.1190, 27.4884, 28.8545, 30.1910, 31.5264, 32.8723, 34.1686, 35.4789, 36.8708, 38.0757, 39.3641, 40.6465, 41.9232, 43.1944, 44.4607, 45.7222, 46.9792, 48.23, 49.48, 50.72, 51.96, 53.20, 54.45, 55.67, 56.89, 58.12, 59.34, 60.56, 61.77, 62.99, 64.20, 65.41, 66.62, 67.82, 69.02, 70.22, 71.42, 72.61, 73.81, 75.00, 76.19, 77.38, 78.57, 79.75, 80.93, 82.12, 83.29, 84.48, 85.65, 86.83, 88.00, 89.18, 90.35, 91.52, 92.69, 93.86, 95.02, 96.19, 97.35, 98.52, 99.68, 100.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>102.00, 103.16, 104.31, 105.47, 106.62, 107.78, 108.94, 110.09, 111.24, 112.39, 113.54, 114.69, 115.84, 116.99, 118.13, 119.28, 120.43, 121.59, 122.74, 123.89, 125.00, 126.14, 127.28, 128.42, 129.56, 130.70, 131.84, 132.98, 134.11, 135.25, 136.38, 137.52, 138.65, 139.79, 140.92, 142.05, 143.18, 144.31, 145.44, 146.56, 147.70, 148.83, 149.96, 151.09, 152.21, 153.34, 154.47, 155.59, 156.72, 157.84, 158.97, 160.09, 161.21, 162.33, 163.46, 164.59, 165.70, 166.82, 167.94, 169.07, 170.18, 171.30, 172.41, 173.53, 174.65, 175.77, 176.88, 178.00, 179.12, 180.23, 181.35, 182.46, 183.58, 184.69, 185.80</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7 - 392
VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:

1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719


DATA TO25 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447, 2.365, 2.306, 2.228, 2.201, 2.179, 2.101, 2.056, 2.024, 2.000, 2.074, 2.069, 2.064, 2.017, 2.001, 1.960 /

LIST OF VARIABLES

ARG INTERMEDIATE CALCULATION VARIABLE
CHI025() TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHI975() TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
CHITAB MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
DD() 1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
I CONTROLS LOOP FOR CHI025() AND CHI975()
IOUT OUTPUT DUMP CONTROLLER
IZER0(2) 2-D ARRAY CONTAINING IO, THE 95% CONFIDENCE INTERVALS ON C FOR EACH REGION
JZERO(2) 2-D ARRAY CONTAINING JO, THE 95% CONFIDENCE INTERVALS ON M FOR EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION, BASED ON MATERIALS DATA ONLY -- MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
NPFR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1)-1) OVER ALL DATA SETS IN A REGION (Number of Points Per Region)
NUM EQUAL TO NPFR(L) FOR A SET OF CALCULATIONS
NUMREG NUMBER OF REGIONS OF INTEREST
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)
SX = EQUAL TO (NPFR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION (X = Ln S)
T VALUE OF TO25() USED IN CALCULATIONS
TTAB MAXIMUM NUMBER OF DEGREES OF FREEDOM IN TO25

INITIALIZE IZERO, JZERO AND MCHAT

DO 50 L = 1, MAXREG
IZER0(1,L) = 0.0
IZER0(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

CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

DO 75 L = 1, NUMREG
IF (NPFR(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 = 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,*),'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,*),'IZER0(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*********************************************************
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
SUBROUTINE GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
C INPUTS: NSETS, NP, NUMREG, LAMN, MCHAT
C OUTPUTS: PVAR
C IMPLICIT NONE
C INTEGER MAXREG, MAXSET
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

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(L), ' PSIG2 = ', PSIG2(L)
300    CONTINUE
    WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
ENDIF

PVAR = SUM / FLOAT(TOTAL)

RETURN
END
SUBROUTINE FNDRNG combines the prior engineering knowledge on both M and C0 with the 95% confidence intervals (JZERO from INTRVL) to obtain posterior credibility ranges on M for each region

PROGRAMMER: L. NEWLIN
DATE: CODE: ZPEB88 FORMAT/COMMENTS: 12AUG91

SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT, RANGEM)

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
IOUT OUTPUT DUMP CONTROLLER
JZERO( ) 2-D ARRAY CONTAINING JS, 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 -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER BOUND
MCHAT( ) 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C FOR EACH REGION -- MCHAT(1,L) = DD(L), THE ESTIMATE FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MCPNT( ) 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MC( ) FOR EACH REGION
MPNT( ) 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZERO( ) FOR EACH REGION
MZERO( ) 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L) IS THE UPPER BOUND
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM( ) 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
UPPER UPPER BOUND OF INTERSECTION

INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
RANGEM(1,L) = 0.0
RANGEM(2,L) = 0.0
50 CONTINUE

PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
DO 100 L = 1, NUMREG
IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG
  WRITE(8,*) 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
ENDIF

IF ((MPNT(L) .EQ. 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),
    & ' RANGEM(2,L) = ', RANGEM(2,L)
  ENDIF
ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) 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. 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)
ENDIF

7 - 397
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.1)) THEN
  THERE IS A PRIOR RANGE ON M AND A RANGE DUE TO CO CONSTRAINT, INTERSECT THESE TWO RANGES WITH Jo
  LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
  UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
  IF (UPPER.LT. LOWER) THEN
    WRITE(8,*),'ERROR: NO INTERSECTION BETWEEN Jo, Mo, ' ,
    'AND MC'
    CALL TRMNAT
  ELSE
    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))
  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

ENDIF
ENDIF
IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
  WRITE(8,*) 'JZERO(2,L) = ', JZERO(2,L),
  &
  WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
  &
  WRITE(8,*) 'MZERO(2,L) = ', MZERO(2,L),
  &
  WRITE(8,*) 'MC(1,L) = ', MC(1,L),
  &
  WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
ENDIF
ELSE
  WRITE(8,*) 'ERROR: PRIOR ON M INCORRECTLY SPECIFIED IN ', L
  CALL ERRNT
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 = I, NUMREG
  IF ((MC(1,L) .LT. RANGEM(1,L))
    & .OR. (MC(1,L) .GT. RANGEM(2,L)))
    &
    WRITE(8,*) 'NOTE: E(m) IS NOT IN THE POSTERIOR RANGE ',
    &
    'ON m IN REGION ', L
300 CONTINUE
RETURN
END

C***************************************************************
C SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C WITHOUT DATA
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 2FEB88   FORMAT/COMMENTS: 12AUG91
C           V8.4, V8.5
C SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
C INPUTS: RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C OUTPUTS: RANGEM, MCHAT, NUMREG
C IMPLICIT NONE
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG
C REAL MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)

7 - 399
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
         MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
         MPNT() FOR EACH REGION
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
         EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
         IS UPPER BOUND
NNODAT NUMBER OF NO DATA regions (REGIONS WITHOUT ANY S/N DATA)
NUMREG NUMBER OF REGIONS OF INTEREST
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
         FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND
         RANGEM(2,L) IS THE UPPER BOUND

IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG
DO 100 L = 1, NNODAT
   NUMREG = NUMREG + 1
   LL = NUMREG
   IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)
   IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
      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),
         ' RANGEM(2,LL) = ', RANGEM(2,LL),
         ' MZERO(2,LL) = ', MZERO(2,LL)
      ENDIF
      SPECIFY E(M) OF POSTERIOR FOR SAKE OF
      CALCULATIONS IN SUBROUTINE EXPCTD
      IF (RANGEM(2,LL) .EQ. 0.0) THEN
         MCHAT(1,LL) = RANGEM(1,LL)
      ELSE
         MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
      ENDIF
   ELSE
      IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT = ', MCHAT(1,LL)
      ELSE
         WRITE(8,*) 'ERROR: OVERALL PRIOR RANGE INCORRECTLY ',
         ' SPECIFIED IN REGION WITHOUT DATA'
      CALL TRMNAT
   ENDIF
100 CONTINUE
RETURN
END

-------------------------------------------------------------

SUBROUTINE CONCAVADJUSTS THE UPPER BOUNDS OF THE POSTERIOR CREDIBILITY
RANGES ON M TO BE CONSISTENT WITH CONCAVITY CONSTRAINTS
PROGRAMMER: L. NEWLIN
DATE: 2FEB88 FORMAT/COMMENTS: 15SEP89
VERSION: MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
         V8.4, V8.5

7 - 400
SUBROUTINE CONCAV (NUMREG, RANGEM)

INPUTS: NUMREG, RANGEM
OUTPUTS: RANGEM
SUBPROGRAMS: TRMNAT

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

ADJUST RANGE TO INSURE CONCAVITY
DO 100 L = NUMREG, 2, -1
  IF (RANGEM(2,L-1) .EQ. 0.0) THEN
    RANGE IS A POINT IN REGION L-1
    IF (RANGEM(1,L-1) .GT. AMAX1(RANGEM(1,L),RANGEM(2,L))) THEN
      WRITE(8,*) 'ERROR: POSTERIOR INTERVAL IN REGION ', L-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
      ENDIF
      ELSE
        RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
  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: 1DEC87

VERSION: MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
V8.4, V8.5


SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

INPUTS: NUMREG, RANGEM

IOUTPUT: MEDM

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

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
RANGE IS A POINT
MEDM(L) = RANGEM(1,L)
ELSEIF (L .EQ. 1) THEN
WE ARE IN REGION ONE — NOT AFFECTED BY OTHER REGIONS
MEDM(L) = (RANGEM(1,L) + RANGEM(2,L)) / 2.0
ELSE
MUST TAKE MEDIAN OF REGION L-1 INTO ACCOUNT
LOWER M = AMAX1(RANGEM(1,L), MEDM(L-1))
MEDM(L) = (LOWER M + RANGEM(2,L)) / 2.0
ENDIF

7 - 402
IF (IOUT .EQ. 10) THEN
    WRITE(8,*), 'L = ', L, ' NUMREG = ', NUMREG
    WRITE(8,*), 'RANGEM(1,L) = ', RANGEM(1,L),
    & WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,L),
    & WRITE(8,*), 'LOWERM = ', LOWERM, ' MEDM(L) = ', MEDM(L)
ENDIF
100 CONTINUE
RETURN
END

C**********************************************************
C SUBROUTINE EXPCTD CALCULATES THE EXPECTED OR MEDIAN VALUES
C OF THE S/N CURVE PARAMETERS
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 13FEB89 FORMAT/COMMENTS: 15SEP89
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

SUBROUTINE EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
         & ZROREG, NBND, BIGK1, BZHAT)

C INPUTS: NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG
C OUTPUTS: BIGK1, BZHAT
C SUBPROGRAMS: TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO
C
C IMPLICIT NONE
INTEGER MAXDAT, MAXREG
PARAMETER (MAXDAT = 50, MAXREG = 3)
COMMON IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG
REAL BIGK(0:MAXREG), BIG1, 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
BIG1   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

7-403
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 %)
S22 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
ZROREG ZER0 REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
ZZ() 1-D ARRAY CONTAINING TRANSFORMED S-N DATA, Z = F(STR, NF, NBND, MM)

C INITIALIZE VARIABLES

DO 50 L = 0, MAXREG
  MM(L) = 0.0
50 CONTINUE

C CREATE MM() ARRAY FROM MEDM() ARRAY

DO 100 L = 1, NUMREG
  MM(L) = MEDM(L)
100 CONTINUE

C TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C CALCULATE BETAo AND K

CALL KBETA (MEANZ, SZ2, KHAT, BZEAT)

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. 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), SEND(L)
  200 CONTINUE
  IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
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
C SUBROUTINE MUSIG (NUMREG, SX2, DD, SUHAT2, SX2, SSX2, NUMPR, DELTA,
& MO, SIGMA2, MCHAT, MU, SIG)
C INPUTS: NUMREG, SX2, DD, SUHAT2, SX2, SSX2, NUMPR, DELTA, MO, SIGMA2
C OUTPUTS: MCHAT, MU, SIG
C IMPLICIT NONE
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NUMREG, NUMPR(MAXREG)
C REAL ARG, DD(MAXREG), DELTA(MAXREG), MCHAT(2, MAXREG),
& MO(MAXREG), MU(MAXREG), SIG(MAXREG), SIGMA2(MAXREG),
& SUHAT2(MAXREG), SSX2, SUHAT2, SX2(MAXREG)
C
C LIST OF VARIABLES

ARG INTERMEDIATE CALCULATION VARIABLE
DD() 1-D ARRAY CONTAINING SKY(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

C
C 200 CONTINUE
WRITE(7,920)
ELSE
WRITE(7,930) MM(1), BIGK(1), KHAT
ENDIF
C FORMAT STATEMENTS
900 FORMAT(/,'PARAMETER VALUES FOR MEDIAN S/N CURVE',//,2X,
& 'NUMBER OF REGIONS:', I4, 5X, 'E(BETA) =', F8.4, 5X, 'E(k) =',
& 'STRESS BOUND',/) 910 FORMAT(5X, I1, 5X, F9.5, 5X, E12.5, 5X, E9.3, 9X, E11.5)
920 FORMAT(/)
930 FORMAT(/,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
& ', 11X, 'm', 14X, 'K', 13X, 'E(k)',
& 'REGION', 7X, 'm', 15X, 'K', 9X, 'LIFE BOUND', 7X,)
RETURN
END
INTEGER

CHAT(1, L) = - DD(L), THE ESTIMATE FOR M AND CHAT(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

NPR() 1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER ALL
DATA SETS IN A REGION (Number of Points Per Region)

SUG() NUMBER OF REGIONS OF INTEREST

SUG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION STANDARD DEVIATION FOR EACH REGION

SUHAT2() 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 NPR() * SX2() FOR A PARTICULAR REGION

SUMX2() 1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)

SX2() 1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION

(X = Ln S)

C INITIALIZE ARRAYS

DO 50 L = 1, MAXREG

CHAT(1, L) = 0.0

CHAT(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

CHAT(1, L) = - DD(L)

CHAT(2, L) = SQRT (SUHAT2(L))

SUMX2 = NPR(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) = CHAT(1, L)

ELSE

UPDATE THE ESTIMATE OF M WITH MO USING DELTA

MU(L) = (CHAT(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 (TOUT .EQ. 10) THEN

WRITE (8, *) 'L = ', L, ' DD = ', DD(L), ' CHAT1 = ',

& WRITE (8, *) 'SUHAT2 = ', SUHAT2(L), ' CHAT2 = ',

& WRITE (8, *) 'NPFR = ', NPR(L), ' SX2 = ', SX2(L),

& WRITE (8, *) 'SUMX2 = ', SUMX2,

& WRITE (8, *) 'DELTA = ', DELTA(L), ' ARG = ', ARG

WRITE (8, *) 'MO = ', MO(L), ' MU = ', MU(L),

& WRITE (8, *) 'WHAT2 = ', WHAT2(L), ' SIGMA2 = ',

& SIG = ', SIG(L)

ENDIF

100 CONTINUE

RETURN
SUBROUTINE NORRNG COMBINES THE PRIOR INFORMATION ON BOTH M AND CO TO
OBTAIN POSTERIOR RANGES ON M FOR EACH REGION

PROGRAMMER: L. NEWLIN
DATE: CODE: 10FEB88 FORMAT/COMMENTS: 12AUG91

SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

INPUTS: NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
OUTPUTS: RANGEM
SUBPROGRAMS: TRMNAT

IMPLICIT NONE
INTEGER MAXR
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG
REAL LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
& RANGEM (2, MAXREG), UPPER

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

INITIALIZE VARIABLES
DO 50 L = 1, MAXREG
 RANGEM(1,L) = 0.0
 RANGEM(2,L) = 0.0
50 CONTINUE

PERFORM CALCULATIONS FOR EACH REGION OF INTEREST
DO 100 L = 1, NUMREG
IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'L = ', L, ' NUMREG = ', NUMREG
WRITE(8,*), 'MPNT = ', MPNT(L), ' MCPNT = ', MCPNT(L)
100 CONTINUE

END
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),
&
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 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),
&
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 AND A LOWER BOUND DUE TO CO
CONSTRAINT ADJUST THE LOWER BOUND OF M0 BY M2

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 M2'
CALL TRAMPAT
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,*), '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 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 M2'
CALL TRAMPAT
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,*), 'RANGEM(1,L) = ', RANGEM(1,L),
&
WRITE(8,*), 'RANGEM(2,L) = ', RANGEM(2,L)
ENDIF

7 - 408
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 TRNAT
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 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,
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 LIST OF VARIABLES
C IOUT OUTPUT DUMP CONTROLLER

7 - 409
CONTROLS DO LOOP FOR EACH REGION

MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED

MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND
C FOR EACH REGION, BASED ON MATERIALS DATA ONLY --

MCHAT(1,L) = - DD(L), THE ESTIMATE FOR M AND

MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C

M() 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() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION MEAN FOR EACH REGION

MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
EACH REGION — MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
IS UPPER BOUND

NNODAT NUMBER OF NO DATA regions (REGIONS WITHOUT ANY S/N DATA)

NUMREG NUMBER OF REGIONS OF INTEREST

RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION — RANGEM(1,L) IS THE LOWER BOUND AND
RANGEM(2,L) IS THE UPPER BOUND

SIG() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
DISTRIBUTION Standard Deviation FOR EACH REGION

SIGMA2() 1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
VARIANCE FOR EACH REGION

IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG = ', NUMREG

DO 100 L = 1, NNODAT
    NUMREG = NUMREG + 1
    LL = NUMREG
    & IF (IOUT .EQ. 10) WRITE(8,*) 'L = ', L, ' NUMREG = ', NUMREG,
    & 'LL = ', LL, ' MPNT(LL) = ', MPNT(LL)
    IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
        POSTERIOR ON M IS SAME AS PRIOR ON M
        RANGEM(1,LL) = MZERO(1,LL)
        RANGEM(2,LL) = MZERO(2,LL)
        MU(LL) = MO(LL)
        SIG(LL) = SQRT(SIGMA2(LL))
        IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,LL) = ', RANGEM(1,LL),
            'MZERO(1,LL) = ', MZERO(1,LL),
            'RANGEM(2,LL) = ', RANGEM(2,LL),
            'MZERO(2,LL) = ', MZERO(2,LL),
            'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
            WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
            SIGMA2(LL)
        ENDIF
    ELSE
        SPECIFY E(M) OF POSTERIOR FOR SAKE OF
CALCULATIONS IN SUBROUTINE EXCTD
        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'
        ENDIF
    CALL TRMNAT
100 CONTINUE
RETURN
END
SUBROUTINE DECOMP CONTROLS THE CALCULATIONS FOR THE INFORMATION
AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
FOR THE STRAIN FORMULATION
PROGRAMMER: L. NEWLIN
DATE: CODE: 21JUN88 FORMAT/COMMENTS: 15SEP89
VERSION: MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
Copyright (C) 1990, California Institute of Technology.
U.S. Government sponsorship under NASA contract NAS7-918
is acknowledged.

SUBROUTINE DECOMP (RANGMP, RANGME, MUP, SIGP, MUE, SIGE, NF,
RFNP, NBND, RFSTRP, RFSTRE, VARY, MPROC,
& KRATIO, PVAR)

INPUTS: READS DATA FROM SPECIFIED AND RELATED; VARY, MPROC
OUTPUTS: RANGMP, RANGME, MUP, SIGP, MUE, SIGE, NF, RFNP,
NBND, RFSTRP, RFSTRE, KRATIO, PVAR
SUBPROGRAMS: INITD, RDECHO, PREF, SW2SU2, INTRVL, FNDRNG,
EXPCTD, PECOMP, MUSIG, NORRNG, GTPVAR
FILES: 5: RELATD-OLD; 6: RELATD-NEW

IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET, MAXTNS
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5, MAXTNS = 5)
COMMON IOUT
INTEGER IOUT, J, MCPNTE(MAXREG), MCPNTP(MAXREG), MPNTE(MAXREG),
& MPNTP(MAXREG), NDC(0:MAXSET),
& NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
& NSETS, NTENS(0:MAXSET), RFNP(MAXREG), VARY
REAL BZERO, BZEROE, BZEROP, DD(MAXREG), DELTAE(MAXREG),
& DELTAP(MAXREG), IZERO(2, MAXREG), JZERO(2, MAXREG),
& KHATE, KHAIP, KRATIO, LANN,
& LINF(MAXDAT, 0:MAXSET, MAXREG),
& LINSTR(MAXDAT, 0:MAXSET, MAXREG),
& LINSTRP(MAXDAT, 0:MAXSET, MAXREG), MCE(2, MAXREG),
& MCHAT(2, MAXREG), MCHATP(2, MAXREG), MCHAT(2, MAXREG),
& MCP(2, MAXREG), MEDME(MAXREG), MEDMP(MAXREG),
& MEO(MAXREG), MOP(MAXREG), MUP(MAXREG),
& MZEROE(2, MAXREG), MZEROP(2, MAXREG), NBND(0:MAXREG),
& NF(MAXDAT, MAXREG), PVAR, RANGME(2, MAXREG),
& RANGMP(2, MAXREG), RAWNF(MAXDAT, 0:MAXSET),
& RAMSTR(MAXDAT, 0:MAXSET), RFSTR(MAXDAT, MAXREG),
& RFSTRP(MAXDAT, MAXREG), SE(MAXDAT, 0:MAXSET),
& SIG2E(MAXREG), SIG2P(MAXREG), SIGE(MAXREG), SIGE(MAXREG),
& SP(MAXDAT, 0:MAXSET), STRE(MAXDAT, 0:MAXSET),
& STRP(MAXDAT, 0:MAXSET), SUHAT2(MAXREG), SWHAT2(MAXREG),
& SX2(MAXREG), SKY(MAXREG), SY2(MAXREG),
& TNSILE(0:MAXSET, MAXTNS)
CHARACTER*40 DESCRP(0:MAXSET)

LIST OF VARIABLES

BZERO VALUE OF WEIBULL PARAMETER, BETA0, CHARACTERIZING S/N DATA SET
BZEROE VALUE OF WEIBULL PARAMETER, BETA0E, CHARACTERIZING ELASTIC COMPONENTS OF S/N DATA SET
BZEROP VALUE OF WEIBULL PARAMETER, BETA0P, CHARACTERIZING PLASTIC COMPONENTS OF S/N DATA SET
DD() 1-D ARRAY CONTAINING SKY(L)/SX2(L) FOR EACH REGION
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
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C

DELTAE

( )

DELTAP

()

DESCRP( )
IOUT
XZERO()
J
.J_ERO( )
i_'rP
KRATIO

M/_DAT
_,_ET
_S
MCE( )

MCHAT ( )
MCHATE ( )

MCHATP ( )

MCP( )

MCPNTE

( )

HCPNTP

( )

NEDHE ( )
MEDHP ( )

HOE( )
MOP()
_trrE

()

),mi'rrP ( )
MPROC

_re: ( )
t,J_P( )
MZEROE ( )

_oP
z,rmCD( )
NOC( )

()

I-D ARRAY CONTAINING
BAYESIAN
_/LTIPLIER
USED
IN MUE()
AND
SIGE ( ) CALCULATION
I-D ARRAY
CONTAINING
BAYESIAN
MULTIPLIER
USED
IN MUP()
AND
SIGP( ) CALCULATION
I-D ARRAY
CONTAINING
DESCRIPTIONS
OF EACH
DATA
SET
OUTPUT
DUMP
CONTROLLER
2-D ARRAY
CONTAINING
Io, THE 95% CONFIDENCE
INTERVALS
ON C
FOR EACH
REGION
CONTROLS
DO LOOP
FOR EACH
DATA
SET
2-D ARRAY
CONTAINING
Jo, THE 95% CONFIDENCE
INTERVALS
ON M
FOR EACH
REGION
E(Ke),
THE MEDIAN
Ke FOR ELASTIC
COMPONENTS
EIKp),
THE MEDIAN
Kp FOR PLASTIC
COMPONENTS
RATIO
OF K*/K,
CONSTANT
OVER
REGIONS
AND COMPONENTS
LAMBDA-N
-- RATIO
OF Var
(Ln N given_S_)
/ (m**2
C*'2),
CONSTANT
OVER
REGIONS
AND COMPONENTS
3-D ARRAY
CONTAINING
LN(RAWNF())
3-D ARRAY CONTAINING
LN(SE())
OR LN(STRE())
3-D ARRAY
CONTAINING
LN(SP())
OR LN(STRP(
MAXIMUM
NUMBER
OF POINTS
II_ _/N DATA
SET
REGION)
ALLOWED
MAXIMUM
NUMBER
OF REGIONS
ALLOWED
MAXIMUM
NUMBER
OF S/N DATA
SETS ALLOWED
MAXIMUM
NUMBER
OF TENSILE
TESTS
PER DATA
SET
2-D ARRAY
CONTAINING
VALUES
OF THE RANGE
ON Me CONSISTENT
WITH
GIVEN
VALUE
OF CO AND THE DATA
-- NOTE:
THE CO
CONSTRAINT
IS NOT APPLICABLE
TO THE STRAIN
FOR_3LATION,
BUT IS A NECESSARY
PARAMETER
TO SOME
OF THE SUBROUTINES
SHARED
WITH
THE STRESS
CASE
2-D ARRAY
CONTAINING
VALUES
OF THE ESTIMATES
OF Cp AND Ce,
BASED
ON MATERIALS
DATA
ONLY
-- MCHAT(2,1)
- MC_ATP(2,1),
MCHAT(2,2)
= MCMATE(2,1)
2-D ARRAY
CONTAINING
VALUES
OF THE ESTIMATES
OF Me AND
Ce FOR ELASTIC
COMPONENTS,
BASED
ON MATERIALS
DATA
ONLY
-- MCHATE(1,
i) m - DD(L),
THE ESTIMATE
FOR Me AND
MCHATE(2,1)
s SUHAT,
THE ESTIMATE
FOR Ce

2-D
Cp FOR

coNTAI.ING
vALUEs
oFTHE

O,DAAO T

PLASTIC
COMPONENTS,
BASED
ON MATERIALS
Y
MCHATP
( i, i) s - DD (L), THE ESTIMATE
FOR Mp AND
MCHATP(2,
i) m SUHAT,
THE ESTIMATE
FOR Cp
2-D ARRAY
CONTAINING
VALUES
OF THE RANGE
ON _
CONSISTENT
WITH
GIVEN
VALUE
OF CO AND THE DATA
-- NOTE.
THE CO
CONSTRAINT
IS NOT APPLICABLE
TO THE STRAIN
FORMULATION,
BUT IS A NECESSARY
PARAMETER
TO SOME
OF THE SUBROUTINES
SHARED
WITH
THE STRESS
CASE
I-D ARRAY
CONTAINING
THE NUMBER
OF POINTS,
0, i, OR 2, IN
MCE()
(ALWAYS
0 FOR STRAIN
CASE)
I-D ARRAY
CONTAINING
THE NUMBER
OF POINTS,
0, i, OR 2, IN
MCP()
(ALWAYS
0 FOR STRAIN
CASE)
I-D ARRAY
CONTAINING
THE MEDIAN
Me VALUE
I-D ARRAY
CONTAINING
THE MEDIAN
Mp VALUE
I-D ARRAY
CONTAINING
THE PRIOR
NORMAL
DISTRIBUTION
MEAN
FOR
ELASTIC
COMPONENTS
I-D ARRAY
CONTAINING
THE PRIOR
NORMAL
DISTRIBUTION
MEAN
FOR
PLASTIC
COMPONENTS
I-D ARRAY
CONTAINING
THE NUMBER
OF POINTS,
0, I, OR 2, IN
MZEROE()
I-D ARRAY CONTAINING
THE NUMBER
OF POINTS,
0, I, OR 2, IN
t'_ER. OP ( )
Materials
PROCess
variation
-- CONTROLS
MATERIALS
PROCESS
VARIATION
-- 0 - NO VARIATION;
I - VARIATION
I-D ARRAY
CONTAINING
THE POSTERIOR
NORMAL
DISTRIBUTION
MEAN
ELASTIC
COMPONENTS
THE POSTERIOR
NORMAL
DISTRIBUTION
MEAN
I-D ARRAY
CONTAINING
PLASTIC
COMPONENTS
2-D
ARRAY
CONTAINING
ELASTIC
COMPONENTS
MZEROE(I
i) IS THE LOWER
BOUND
AND
MZEROE(2,1)
IS THE _p ER som_
2-D ARRAY
CONTAINING
VALUES
OF THE PRIOR
RANGE
ON Mp FOR
PLASTIC
COMPONENTS
-- MZEROP(I,1)
IS THE IXMER
BOUND
AND
MZEROP
(2,1)
IS THE UPPER
BOUND
I-D ARRAY
CONTAINING
UPPER
BOUNDS
(CYCLES)
FOR THE NUMREG
REGIONS
OF INTEREST

vALUEs
orTHEPRIOR

I-D ARRAY
POINTS

CONTAINING
Number
FOR EACH
DATA
SET

7 - 412

of

given

DeComposed

ONMe

STRAIN

FOR
FOR


OPEN(5, FILE = 'RELATD', STATUS = 'OLD')
OPEN(6, FILE = 'RELATO', STATUS = 'NEW')
C RELATD CONTAINS THE RELATED MATERIAL S/N DATA SET INFORMATION
C RELATO CONTAINS THE PROCESSED RELATED MATERIAL S/N DATA SET
C INFORMATION
C INITIALIZE PRIMARY ARRAYS

CALL INITD (NPTS, RAWNF, RAWSTR, NDC, SP, SE, NTENS, TNSILE, & RFNP, RFSTRE, RFSTRP, NBND, DELTAP, DELTAE, & MOP, MOE, SIG2P, SIG2E, MCHAT)
C READ AND ECHO INFORMATION
CALL RDECHO (VARY, MPROC, NPTS, RAWNF, RAWSTR, NSETS, DESCRP, & NDC, SP, SE, NTENS, TNSILE, MEZERO, MPNTR, MZEROE, & MPNTE, DELTAZ, DELTAE, MOE, MOE, SIG2P, SIG2E, & KRAPO, LAMN)

C PREPARE GIVEN DECOMPOSED STRAIN DATA FOR INFORMATION AGGREGATION PROCESSING
C CALL PREP (0, NDC, SP, SE, RAWNF, NP, LNSTRP, LNSTRE, LNNF, & RFNP, RFSTRP, RFSTRE, NF)

C BEGIN INFORMATION AGGREGATION PROCESSING FOR THE GIVEN PLASTIC STRAIN DATA
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (1, 0, NP, LNSTRP, LNNF, SX2, SXY, SY2, DD, & SWHAT2, SUHAT2, NPPR)

C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS
CALL INTRVL (1, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO, & MCHATP)

C WRITE RESULTS TO FILE
WRITE (7,900)
WRITE(7,905) JZERO(1,1), JZERO(2,1), MCHATP(1,1)
WRITE (7,910)

C CALCULATE MEDIAN VALUES FOR PLASTIC S/N CURVE PARAMETERS
C BASED ON GIVEN DECOMPOSITION DATA
MEDMP(1) = MCHATP(1,1)
CALL EXPCTD (2, MEDMP, RFNP, RFSTRP, NF, 0.0, 1, 1, NBND, KHAPE, & BZEROP)

C BEGIN INFORMATION AGGREGATION PROCESSING FOR THE GIVEN ELASTIC STRAIN DATA
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (1, 0, NP, LNSTRE, LNNF, SX2, SXY, SY2, DD, & SWHAT2, SUHAT2, NPPR)

C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS
CALL INTRVL (1, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO, & MCHATE)

C WRITE RESULTS TO FILE
WRITE (7,915)
WRITE(7,920) JZERO(1,1), JZERO(2,1), MCHAT(1,1)
WRITE (7,925)

C CALCULATE MEDIAN VALUES FOR ELASTIC S/N CURVE PARAMETERS
C BASED ON GIVEN DECOMPOSITION DATA
MEDME(1) = MCHAT(1,1)
CALL EXPCTD (2, MEDME, RFNP, RFSTRE, NF, 0.0, 1, 1, NBND, KHATE, & BZEREO)

C DECOMPOSE TOTAL STRAIN DATA BASED ON MEDIAN S/N CURVE PARAMETERS FOR THE GIVEN PLASTIC AND ELASTIC STRAIN COMPONENT DATA TO OBTAIN ESTIMATED PLASTIC AND ELASTIC STRAIN COMPONENTS
CALL PECOMP (NSETS, DESCRP, NDC, SP, SE, NPTS, RAWSTR, & RAWNF, NTENS, TNSILE, MEZEP, KHAPE, KHATE, MCHATP(1,1), & MCHAT(1,1), STRP, STRE)

C PREPARE ESTIMATED DECOMPOSED STRAIN DATA FOR INFORMATION AGGREGATION PROCESSING

7-414
CALL PREP (NSETS, NPTS, STRP, STRE, RAWNF, RP, LNSTRP, LNSTRE,
& LNNF, RFNP, RFSTRP, RFSTRE, RP)

IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1) .OR. (VARY .EQ. 2)) THEN
C BEGIN INFORMATION AGGREGATION PROCESSING FOR THE ESTIMATED PLASTIC
C STRAIN DATA -- FOR ALL TYPES OF VARIATION SAVE NORMAL
C CALCULATE RESIDUAL VARIANCES
& CALL SW2SU2 (1, NSETS, NP, LNSTRP, LNNF, SX2, SX2, SY2, DD,
& SWHAT2, SUHAT2, NPPR)
C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS
& CALL INTRVL (1, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO,
& MCHATP)
C COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION
C TO OBTAIN POSTERIOR RANGE ON Mp
C WRITE RESULTS TO FILE
WRITE (7, 930)
WRITE (7, 905) JZERO(1,1), JZERO(2,1), MCHATP(1,1)
WRITE (7, 935)
WRITE (7, 940) 'mp', RANGMP(1,1), RANGMP(2,1)
WRITE (7, 945)

C CALCULATE MEDIAN VALUES FOR PLASTIC S/N CURVE PARAMETERS
C BASED ON ESTIMATED DECOMPOSITION DATA
IF (RANGMP(2,1) .EQ. 0.0) THEN
RANGE ON Mp IS A POINT
MEDMP(1) = RANGMP(1,1)
ELSE
RANGE ON Mp IS AN INTERVAL
MEDMP(1) = (RANGMP(1,1) + RANGMP(2,1)) / 2.0
ENDIF
CALL EXPCTD (2, MEDMP, RFNP, RFSTRP, NP, 0.0, 1, 1, NBND,
& KHATP, BZEROP)
C BEGIN INFORMATION AGGREGATION PROCESSING FOR THE ESTIMATED ELASTIC
C STRAIN DATA
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (1, NSETS, NP, LNSTRE, LNNF, SX2, SX2, SY2, DD,
& SWHAT2, SUHAT2, NPPR)
C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS
CALL INTRVL (1, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO,
& MCHATP)
C COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION
C TO OBTAIN POSTERIOR RANGE ON Me
CALL FNDRNG (1, MPNTE, MZEROE, MCPNTE, MCP, JZERO, MCHATE,
& RANGME)
C WRITE RESULTS TO FILE
WRITE (7, 950)
WRITE (7, 920) JZERO(1,1), JZERO(2,1), MCHATE(1,1)
WRITE (7, 955)
WRITE (7, 940) 'me', RANGE(1,1), RANGME(2,1)
WRITE (7, 960)
CALCULATE MEDIAN VALUES FOR ELASTIC S/N CURVE PARAMETERS

BASED ON ESTIMATED DECOMPOSITION DATA

IF (RANGME(2,1) .EQ. 0.0) THEN
  RANGE ON Me IS A POINT
  MEDME(1) = RANGME(1,1)
ELSE
  RANGE ON Me IS AN INTERVAL
  MEDME(1) = (RANGME(1,1) + RANGME(2,1)) / 2.0
ENDIF

CALL EXPCTD (2, MEDME, RFNP, RFSTRE, NF, 0.0, 1, 1, NBND, KHADE, BZEROE)
&
BZERO = 1.0 / SQRT (0.5 * ((1.0 / BZEROP) ** 2 + (1.0 / BZEROP) ** 2))
&WRI 7,985) BZERO

CHECK TYPE OF VARIATION DESIRED AND FIX M RANGES AT MEDIAN
IF NECESSARY

IF (((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
  RANGME(1,1) = MEDME(1)
  RANGME(2,1) = MEDME(1)
  RANGMP(1,1) = MEDMP(1)
  RANGMP(2,1) = MEDMP(1)
ENDIF
ELSE
BEGIN INFORMATION AGGREGATION PROCESSING FOR THE ESTIMATED PLASTIC
C STRAIN DATA -- FOR NORMAL VARIATION
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (1, NSETS, NP, LNSTRP, LNNF, SX2, SY2, SY2, DD, 
&
CALL THE POSTERIOR MEAN AND STANDARD DEVIATION
CALL MUSIG (1, SX2, DD, WHAT2, WHAT2, NPFR, DELTAP, MOP, 
&
USE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGE ON ME
CALL NORRNG (1, MPNTE, MZEROP, MCPNTE, MCP, MCHATP, RANGMP)
C WRITE RESULTS TO FILE
WRI 7,975) MCHATP(1,1)
WRI 7,935)
WRI 7,940) 'MP', RANGMP(1,1), RANGMP(2,1)
WRI 7,965) MUP(1), SIGP(1)
BEGIN INFORMATION AGGREGATION PROCESSING FOR THE ESTIMATED ELASTIC
C STRAIN DATA
C CALCULATE RESIDUAL VARIANCES
CALL SW2SU2 (1, NSETS, NP, LNSTR, LNNF, SX2, SY2, SY2, DD, 
&
CALL THE POSTERIOR MEAN AND STANDARD DEVIATION
CALL MUSIG (1, SX2, DD, WHAT2, WHAT2, NPFR, DELTAE, MOE, 
&
USE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGE ON ME
CALL NORRNG (1, MPNTE, MZEROE, MCPNTE, MCE, MCHAT, RANGME)
C WRITE RESULTS TO FILE

7 - 416
WRITE(7,990) MCHATE(1,1)
WRITE(7,955)
WRITE(7,940) 'me', RANGME(1,1), RANGME(2,1)
WRITE(7,970) MUE(1), SIGE(1)

ENDIF

C CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

IF (MPROC .EQ. 1)
  THEN
    DO 100 J = 0, NSETS
    CONTINUE
    MCHAT(2,1) = MCHATP(2,1)
    MCHAT(2,2) = MCHAT(2,1)
    CALL GTPVAR (NSETS, NP, 2, LAMN, MCHAT, 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.',/7X,'RESULTS OF STRAIN DECOMPOSITION AND INFORMATION ',
& 'AGGREGATION CALCULATIONS',/7X,'95% CONFIDENCE INTERVAL AND POINT ESTIMATE OF me',
& '/2X,'FOR GIVEN PLASTIC COMPONENTS'/)
905 FORMAT(7X,'Jop = ( ',F12.9,', ',F12.9,')',5X,'mp = ',F9.6)
910 FORMAT(///,2X,'RESULTS FOR GIVEN PLASTIC COMPONENT DATA'/)
915 FORMAT(///,2X,'95% CONFIDENCE INTERVAL AND POINT ESTIMATE OF me',
& '/2X,'FOR GIVEN ELASTIC COMPONENTS'/)
920 FORMAT(7X,'Joe = ( ',F12.9,', ',F12.9,')',5X,'me = ',F9.6)
925 FORMAT(///,2X,'RESULTS FOR GIVEN ELASTIC COMPONENT DATA'/)
930 FORMAT(///,2X,'RESULTS OF INFORMATION AGGREGATION'/,
& '2X,95% CONFIDENCE INTERVAL AND POINT ESTIMATE OF mp',
& '/2X,'FOR ESTIMATED PLASTIC COMPONENTS'/)
935 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON mp FOR ESTIMATED ',
& 'PLASTIC COMPONENTS'/)
940 FORMAT(7X,A2, ' = ( ',F7.4,', ',F7.4,')')
945 FORMAT(///,2X,'RESULTS FOR ESTIMATED PLASTIC COMPONENT DATA'/)
950 FORMAT(///,2X,'95% CONFIDENCE INTERVAL AND POINT ESTIMATE OF me',
& '/2X,'FOR ESTIMATED ELASTIC COMPONENTS'/)
955 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON me FOR ESTIMATED ',
& 'ELASTIC COMPONENTS'/)
960 FORMAT(///,2X,'RESULTS FOR ESTIMATED ELASTIC COMPONENT DATA'/)
965 FORMAT(///,2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS FOR ',
& 'ESTIMATED PLASTIC COMPONENTS',/12X,'MEAN',8X,'STD DEV',
& '/10X,F7.4,5X,E11.5)
970 FORMAT(///,2X,'POSTERIOR NORMAL DISTRIBUTION PARAMETERS FOR ',
& 'ESTIMATED ELASTIC COMPONENTS',/12X,'MEAN',8X,'STD DEV',
& '/10X,F7.4,5X,E11.5)
975 FORMAT(///,2X,'POINT ESTIMATE OF mp FOR ESTIMATED PLASTIC ',
& 'COMPONENTS',/7X,'mp = ',F9.6)
SUBROUTINE INITD PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS USED IN THE STRAIN FORMULATION OF THE INFORMATION AGGREGATION MODEL CALCULATIONS

PROGRAMMER: L. NEWLIN
DATE: CODE: 21JUN88 COMMENTS: 13JUL89
VERSION: MATCHR V8.1, V8.2, V8.3, V8.4, V8.5

SUBROUTINE INITD (NPTS, RAWNF, RAWSTR, NDC, SP, SE, NTENS, & TNSILE, NF, LNRFN, LNSTRP, LNSTRE, STRP, STRE, & MZEROP, MZEROE, MPNTE, MPNTE, MCP, MCE, MCPNTE, & MCPNTE, RFNP, NF, RFSTRE, RFSTRP, NBND, DELTAP, & DELTAE, MOP, MOE, SIG2P, SIG2E, MCHAT)

INPUTS: NPTS, RAWNF, RAWSTR, NDC, SP, SE, NTENS, TNSILE, NF, LNRFN, LNSTRP, LNSTRE, STRP, STRE, MZEROP, MZEROE, MPNTE, MPNTE, MCP, MCE, MCPNTE, MCPNTE, RFNP, NF, RFSTRE, RFSTRP, NBND, DELTAP, DELTAE, MOP, MOE, SIG2P, SIG2E, MCHAT

OUTPUTS: NPTS, RAWNF, RAWSTR, NDC, SP, SE, NTENS, TNSILE, NF, LNRFN, LNSTRP, LNSTRE, STRP, STRE, MZEROP, MZEROE, MPNTE, MPNTE, MCP, MCE, MCPNTE, MCPNTE, RFNP, NF, RFSTRE, RFSTRP, NBND, DELTAP, DELTAE, MOP, MOE, SIG2P, SIG2E, MCHAT

IMPLICIT NONE

INTEGER MAXDAT, MAXREG, MAXSET, MAXTNS
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5, MAXTNS = 5)

COMMON IOUT
INTEGER I, IOUT, J, K, L, M, MCPNTE(MAXREG), MCPNTP(MAXREG), & MPNTE(MAXREG), MPNTP(MAXREG), NDC(0:MAXSET), & MP(0:MAXSET, MAXREG), NPTS(0:MAXSET), NTENS(0:MAXSET), & RFNP(MAXREG)
REAL DELTAE(MAXREG), DELTAP(MAXREG), & LNSTRE(MAXDAT, 0:MAXSET, MAXREG), & LNSTRP(MAXDAT, 0:MAXSET, MAXREG), & MCHAT(2, MAXREG), MCP(2, MAXREG), MOE(MAXREG), & MOP(MAXREG), MZEROE(2, MAXREG), MZEROP(2, MAXREG), & NBND(0:MAXREG), NF(MAXDAT, MAXREG), & RANF(MAXDAT, 0:MAXSET), RANSTR(MAXDAT, 0:MAXSET), & RFSTR(MAXDAT, MAXREG), RFSTRP(MAXDAT, MAXREG), & SE(0:MAXSET), SIG2E(MAXREG), SIG2P(MAXREG), & SP(MAXDAT, 0:MAXSET), STRE(MAXDAT, 0:MAXSET), & STRP(MAXDAT, 0:MAXSET), TNSILE(0:MAXSET, MAXTNS)

LIST OF VARIABLES
DELTAE() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MUE() AND SIGE() CALCULATION
DELTAP() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MPU() AND SIGP() CALCULATION
I CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
IOUT
OUTPUT DUMP CONTROLLER
J
CONTROLS DO LOOP FOR EACH DATA SET
K
CONTROLS DO LOOP FOR EACH POINT IN A REGION
L
CONTROLS DO LOOP FOR EACH REGION
LNRFN() 3-D ARRAY CONTAINING LN(RANUF())
LNSTR() 3-D ARRAY CONTAINING LN(SE()) OR LN(STR())
LNSTRP() 3-D ARRAY CONTAINING LN(SP()) OR LN(STRP())
M
CONTROLS DO LOOP FOR EACH TENSILE TEST POINT IN A DATA SET
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
MAXTNS MAXIMUM NUMBER OF TENSILE TESTS PER DATA SET ALLOWED
MCE() 2-D ARRAY CONTAINING VALUES OF THE RANGE ON Me CONSISTENT WITH GIVEN VALUE OF CO AND THE DATA -- NOTE: THE CO CONSTRAINT IS NOT APPLICABLE TO THE STRAIN FORMULATION, BUT IS A NECESSARY PARAMETER TO SOME OF THE SUBROUTINES SHARED WITH THE STRESS CASE
MCHAT() 2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF CO AND Ce, BASED ON MATERIALS DATA ONLY -- MCHAT(2,1) = MCHATP(2,1) AND MCHAT(2,2) = MCHAT(2,1)
MCP() 2-D ARRAY CONTAINING VALUES OF THE RANGE ON Md CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA -- NOTE: THE CO CONSTRAINT IS NOT APPLICABLE TO THE STRAIN FORMULATION, BUT IS A NECESSARY PARAMETER TO SOME OF THE SUBROUTINES SHARED WITH THE STRESS CASE
MCPNTE() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MCE() (ALWAYS 0 FOR STRAIN CASE)
MCPNTP() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MCP() (ALWAYS 0 FOR STRAIN CASE)
MOE() 1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR ELASTIC COMPONENTS
MOP() 1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR PLASTIC COMPONENTS
MPNT() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZEROE()
MPNTP() 1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN MZEROE()
MZEROE() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Me FOR ELASTIC COMPONENTS -- MZEROE(1,1) IS THE LOWER BOUND AND MZEROE(2,1) IS THE UPPER BOUND
MZERO() 2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Md FOR PLASTIC COMPONENTS -- MZERO(1,1) IS THE LOWER BOUND AND MZERO(2,1) IS THE UPPER BOUND
NBND() 1-D ARRAY CONTAINING UPPPER BOUND (CYCLES) FOR THE NUMREG REGIONS OF INTEREST
NDC() 1-D ARRAY CONTAINING Number of given Decomposed STRAIN POINTS FOR EACH DATA SET
NF() 2-D ARRAY CONTAINING RAWF() 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
NTSNS() 1-D ARRAY CONTAINING NUMBER OF TENSILE TEST DATA POINTS FOR EACH DATA SET
RAWF() 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 RAW TOTAL STRAIN DATA (%) FOR ALL S/N DATA SETS
RFN() 1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC MATERIAL S/N DATA SET
RFST() 2-D ARRAY CONTAINING ELASTIC STRAIN POINTS (%), SE() OR STR() FOR THE SPECIFIC MATERIAL S/N DATA SET
RFSTRP() 2-D ARRAY CONTAINING PLASTIC STRAIN POINTS (%), SP() OR STRP() FOR THE SPECIFIC MATERIAL S/N DATA SET
SEP() 2-D ARRAY CONTAINING GIVEN ELASTIC STRAIN COMPONENTS (%)
SIG2E() 1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR ELASTIC COMPONENTS
SIG2P() 1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE FOR PLASTIC COMPONENTS
SP() 2-D ARRAY CONTAINING GIVEN PLASTIC STRAIN COMPONENTS (%)
ST() 2-D ARRAY CONTAINING CALCULATED ELASTIC STRAIN COMPONENTS (%)
STRP() 2-D ARRAY CONTAINING CALCULATED PLASTIC STRAIN COMPONENTS (%)
TNSILE() 2-D ARRAY CONTAINING TENSILE TEST DATA -- INDEXED FOR EACH DATA SET

7-419
DO 100 J = 0, MAXSET
   NDC(J) = 0
   NPT(J) = 0
   NTENS(J) = 0
100 CONTINUE

DO 200 J = 0, MAXSET
   DO 250 I = 1, MAXDAT
      SE(I, J) = 0.0
      SP(I, J) = 0.0
      RNWF(I, J) = 0.0
      RNSR(I, J) = 0.0
      STR(I, J) = 0.0
      STRP(I, J) = 0.0
250 CONTINUE
200 CONTINUE

DO 300 L = 1, MAXREG
   DO 325 J = 0, MAXSET
      DO 350 K = I, MAXDAT
         LNNF(K, J, L) = 0.0
         LNSTRE(K, J, L) = 0.0
         LNSRP(K, J, L) = 0.0
350 CONTINUE
325 CONTINUE
300 CONTINUE

DO 400 L = 1, MAXREG
   DO 450 J = 0, MAXSET
      CONTROL = 0
450 CONTINUE
400 CONTINUE

DO 500 L = 1, MAXREG
   MZEROP(1, L) = 0.0
   MZEROP(2, L) = 0.0
   MZEROE(1, L) = 0.0
   MZEROE(2, L) = 0.0
   MFNTP(L) = 0
   MPNTE(L) = 0
   MCP(1, L) = 0.0
   MCP(2, L) = 0.0
   MCE(1, L) = 0.0
   MCE(2, L) = 0.0
   MCPNTP(L) = 0
   MCPNTE(L) = 0
   RFNP(L) = 0
   DELTAP(L) = 0.0
   DELTAE(L) = 0.0
   MOP(L) = 0.0
   MOE(L) = 0.0
   SIG2P(L) = 0.0
   SIG2E(L) = 0.0
   MCHAT(1, L) = 0.0
   MCHAT(2, L) = 0.0
500 CONTINUE

DO 525 L = 0, MAXREG
   NBND(L) = 0.0
525 CONTINUE
NBND(1) = 1.0E+36

DO 600 L = 1, MAXREG
   DO 650 K = 1, MAXDAT
      NF(K, L) = 0.0
      RFSRE(K, L) = 0.0
      RFSTRP(K, L) = 0.0
650 CONTINUE
600 CONTINUE

DO 700 M = 1, MAXTNS
   DO 750 J = 0, MAXSET
      TNSILE(J, M) = 0.0
750 CONTINUE
C SUBROUTINE RDECHO READS THE DATA FROM SPECIFIED AND RELATED AND ECHOES
C THE DATA TO SPECIFIED AND RELATED
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88 FORMAT/COMMENTS: 15SEP89

SUBROUTINE RDECHO (VARY, MPROC, NPTS, RAWNF, RAWSTR, NSETS, &
DESCRP, NDC, SP, SE, NTENS, TNSILE, MZEROP, &
MPNTP, MZEROE, MPNTE, DELTAP, DELTAE, MOP, &
MOE, SIG2P, SIG2E, KRATIO, LAMN)
C INPUTS: VARY, MPROC
C OUTPUTS: NPTS, RAWNF, RAWSTR, NSETS, DESCRP, NDC, SP, SE, &
NTENS, TNSILE, MZEROP, MPNTP, MZEROE, MPNTE, DELTAP, &
DELTAE, MOP, MOE, SIG2P, SIG2E, KRATIO, LAMN
C SUBPROGRAMS: TRMNAT
C
C IMPLICIT NONE
INTEGER MAXDAT, MAXREG, MAXSET, MAXTNS
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5, MAXTNS = 5)
COMMON IOUT, IN, IOUT, J, M, MPNTE(MAXREG), MPNTP(MAXREG), MPROC, &
NDC(0:MAXSET), NPTS(0:MAXSET), NSETS, NTENS(0:MAXSET), &
VARY
REAL DELTAE(MAXREG), DELTAP(MAXREG), KRATIO, LAMN, &
MOE(MAXREG), MOP(MAXREG), MZEROP(2, MAXREG), &
MPNTP(2, MAXREG), RAWNF(MAXDAT, 0:MAXSET), &
RAWSTR(MAXDAT, 0:MAXSET), SE(MAXDAT, 0:MAXSET), &
SIG2E(MAXREG), SIG2P(MAXREG), SP(MAXDAT, 0:MAXSET), &
TNSILE(0:MAXSET, MAXTNS)
CHARACTER*40 DESCRP(0:MAXSET)

LIST OF VARIABLES
DELTAE() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED MUE() AND
SIGE() CALCULATION
DELTAP() 1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED MUP() AND
SIGP() CALCULATION
DESCRP() 1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
I CONTROLS DO LOOP FOR EACH POINT IN A DATA SET
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH DATA SET
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
LAMN LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
           CONSTANT OVER REGIONS AND COMPONENTS
M CONTROLS DO LOOP FOR EACH TENSILE POINT IN A DATA SET
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 SET ALLOWED
MAXTNS MAXIMUM NUMBER OF TENSILE TESTS PER DATA SET ALLOWED
MOE() 1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
       ELASTIC COMPONENTS
MOP() 1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
       PLASTIC COMPONENTS
READ DECOMPOSED STRAIN DATA FOR THE SPECIFIC MATERIAL DATA
SET FROM SPECFD AND ECHO TO SPECFO

READ(1,*) DESCRP(0)
READ(1,*) NDC(0), NPTS(0), NTENS(0)

IF (NDC(0) .GT. MAXDAT) THEN
  WRITE(8,*) 'ERROR: EXCEEDED MAXIMUM NUMBER OF POINTS IN ', &
  'SPECIFIC MATERIAL DATA SET'
  CALL TRMNAT
ENDIF

IF (NPTS(0) .GT. MAXDAT) THEN
  WRITE(8,*) 'ERROR: EXCEEDED MAXIMUM NUMBER OF POINTS IN ', &
  'SPECIFIC MATERIAL DATA SET'
  CALL TRMNAT
ENDIF

IF (NTENS(0) .GT. MAXTNS) THEN
  WRITE(8,*) 'ERROR: EXCEEDED MAXIMUM NUMBER OF TENSILE ', &
  'POINTS IN SPECIFIC MATERIAL DATA SET'
  CALL TRMNAT
ENDIF

WRITE(3,900) DESCRP(0), NDC(0), NPTS(0), NTENS(0)
WRITE(3,910)

IF (IOUT .EQ. I0) THEN
  WRITE(8,900) DESCRP(0), NDC(0), NPTS(0), NTENS(0)
  WRITE(8,910)
ENDIF

DO 50 I = 1, NDC(0)
  READ(1,*) RAWSTR(I,0), RAWNF(I,0), SP(I,0), SE(I,0)
  WRITE(3,920) RAWSTR(I,0), RAWNF(I,0), SP(I,0), SE(I,0)
  IF (IOUT .EQ. I0) WRITE(8,920) RAWSTR(I,0), RAWNF(I,0), SP(I,0), SE(I,0)
  CONTINUE

READ REMAINING TOTAL STRAIN DATA FOR THE SPECIFIC MATERIAL
DATA SET FROM SPECFD AND ECHO TO SPECFO
DO 60 I = (NDC(0) + 1), NPTS(0)
READ(1,*) RAWSTR(I,0), RAWNF(I,0)
WRITE(3,930) RAWSTR(I,0), RAWNF(I,0)
IF (IOUT .EQ. 10) WRITE(8,930) RAWSTR(I,0), RAWNF(I,0)
60 CONTINUE
WRITE(3,980)
IF (IOUT .EQ. 10) WRITE(8,980)
DO 70 M = 1, NTENS(0)
READ(1,*) TNSILE(0,M)
WRITE(3,990) TNSILE(0,M)
IF (IOUT .EQ. 10) WRITE(8,990) TNSILE(0,M)
70 CONTINUE
C
READ PRIOR RANGE INFORMATION FROM SPECFD
READ(1,*) MPNTP(1), MZEROP(1,1), MZEROE(2,1)
WRITE(3,960) MPNTP(1), MZEROP(1,1), MZEROE(2,1)
READ(1,*) MPNTE(1), MZEROE(1,1), MZEROP(2,1)
WRITE(3,965) MPNTE(1), MZEROE(1,1), MZEROP(2,1)
IF (IOUT .EQ. 10) THEN
WRITE(8,960) MPNTP(1), MZEROP(1,1), MZEROE(2,1)
WRITE(8,965) MPNTE(1), MZEROE(1,1), MZEROP(2,1)
ENDIF
IF ((VARY .EQ. 3) .AND. 
& ((MPNTP(1) .EQ. 0.0) .OR. (MPNTE(1) .EQ. 0.0))) THEN
WRITE(8,*) 'ERROR: NORMAL VARIATION REQUIRES PRIOR RANGES ',
& 'ON MP AND Me'
CALL TRMNAT
ENDIF
C
IF (VARY .EQ. 3) THEN
READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
READ(1,*) DELTAP(1), MOP(1), SIG2P(1)
WRITE(3,970) DELTAP(1), MOP(1), SIG2P(1)
IF ((DELTAP(1) .LT. 0.0) .OR. 
& (DELTAP(1) .GT. 0.0) .AND. (MOP(1) .LE. 0.0))) THEN
WRITE(8,*) 'ERROR: BAD VALUE FOR DELTAP OR VALUE OF ',
& 'MOP INCONSISTENT WITH DELTAP'
CALL TRMNAT
ENDIF
READ(1,*) DELTAE(1), MOE(1), SIG2E(1)
WRITE(3,975) DELTAE(1), MOE(1), SIG2E(1)
IF ((DELTAE(1) .LT. 0.0) .OR. 
& (DELTAE(1) .GT. 0.0) .AND. (MOE(1) .LE. 0.0))) THEN
WRITE(8,*) 'ERROR: BAD VALUE FOR DELTAE OR VALUE OF ',
& 'MOE INCONSISTENT WITH DELTAE'
CALL TRMNAT
ENDIF
IF (IOUT .EQ. 10) THEN
WRITE(8,970) DELTAP(1), MOP(1), SIG2P(1)
WRITE(8,975) DELTAE(1), MOE(1), SIG2E(1)
ENDIF
ENDIF
IF (MPROC .EQ. 1) THEN
READ(1,*) KRATIO, LAMN
WRITE(3,995) KRATIO, LAMN
IF (IOUT .EQ. 10) WRITE(8,995) KRATIO, LAMN
ENDIF
C
READ STRAIN DATA FOR THE RELATED MATERIAL DATA SETS
FROM RELATD AND ECHO TO RELATO
READ(5,*) NSETS
IF (NSETS .GT. MAXSET) THEN
WRITE(8,*) 'ERROR: EXCEEDED MAXIMUM NUMBER OF RELATED ',
& 'DATA SETS'
CALL TRMNAT
ENDIF
WRITE(6,940) NSETS
IF (IOUT .EQ. 10) WRITE(8,940) NSETS
DO 100 J = 1, NSETS
  READ(5,*) DESCRP(J), NDC(J), NPTS(J), NTENS(J)
  IF (NDC(J) .GT. MAXDAT) THEN
    WRITE(8,*) 'ERROR: EXCEEDED MAXIMUM NUMBER OF POINTS IN ', &
    'RELATED DATA SET', J
    CALL TRMNAT
  ENDIF
  IF (NPTS(J) .GT. MAXDAT) THEN
    WRITE(8,*) 'ERROR: EXCEEDED MAXIMUM NUMBER OF POINTS IN ', &
    'RELATED DATA SET', J
    CALL TRMNAT
  ENDIF
  IF (NTENS(J) .GT. MAXTNS) THEN
    WRITE(8,*} 'ERROR: EXCEEDED MAXIMUM NUMBER OF TENSILE ', &
    'POINTS IN RELATED DATA SET', J
    TRMNAT
  ENDIF
WRITE(6,950) DESCRP(J), NDC(J), NPTS(J), NTENS(J)
WRITE(6,910)
IF (IOUT .EQ. 10) THEN
  WRITE(8,*) DESCR_C(J), SDC(J), NPTS(J), TESS(J)
  WRITE (8, 910) &
ENDIF
DO 125 I = 1, NDC(J)
  READ(5.*) RAWSTR(I,J), RAWNF(I,J), SP(I,J), SE(I,J)
  IF (IOUT .EQ. 10) WRITE(8,920) RAWSTR(I,J), RAWNF(I,J) &
  ENDIF
DO 150 I = (NDC(J) + 1), NPTS(J)
  READ(5.*) RAWSTR(I,J), RAWNF(I,J)
  WRITE(6,930) RAWSTR(I,J), RAWNF(I,J)
  IF (IOUT .EQ. 10) WRITE(8,930) RAWSTR(I,J), RAWNF(I,J) &
  ENDIF
WRITE(6,980)
IF (IOUT .EQ. 10) WRITE(8,980)
DO 175 M = 1, NTENS(J)
  READ(5.*) TNSILE(J,M)
  WRITE(6,990) TNSILE(J,M)
  IF (IOUT .EQ. 10) WRITE(8,990) TNSILE(J,M) &
  ENDIF
CONTINUE
10 CONTINUE
900 FORMAT///,13X,'MATERIAL INPUT',///,2X,'DESCRIPTION:',2X,A40, &
  /,5X,'NUMBER OF DECOMPOSED STRAIN POINTS:',2X,I2, &
  /,2X,'NUMBER OF POINTS IN SPECIFIC DATA SET:',2X,I2, &
  /,10X,'NUMBER OF TENSILE TEST POINTS:',2X,I2)
910 FORMAT///,7X,'TOTAL STRAIN',5X,'LIFE',5X,'PLASTIC STRAIN', &
  /,5X,'ELASTIC STRAIN',/)
920 FORMAT(10X,F7.4,6X,F6.0,8X,F8.5,10X,F8.5)
930 FORMAT(10X,F7.4,6X,F6.0)
940 FORMAT(20X,'NUMBER OF DATA SETS:',2X,I2)
950 FORMAT///,2X,'DESCRIPTION:',2X,A40, &
  /,2X,'NUMBER OF DECOMPOSED STRAIN POINTS:',2X,I2, &
  /,8X,'NUMBER OF POINTS IN DATA SET:',2X,I2, &
  /,7X,'NUMBER OF TENSILE TEST POINTS:',2X,I2)
C SUBROUTINE PREP PREPARES THE DECOMPOSED STRAIN COMPONENT DATA FOR 
C PROCESSING THROUGH THE INFORMATION AGGREGATION MODEL -- STRAIN DATA 
C MUST CONFORM TO DATA STRUCTURE REQUIRED BY THE STRESS FORMULATION 
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, 
V8.4, V8.5 
C SUBROUTINE PREP (NSETS, NUM, PLAS, ELAS, RAWNF, NP, LNSTP, 
& LNSTRE, LNNF, RFNP, RFSTRP, RFSTRE, NF) 
C INPUTS: NSETS, NUM, PLAS, ELAS, RAWNF 
C OUTPUTS: NP, LNSTP, LNSTRE, LNNF, RFNP, RFSTRP, RFSTRE, NF 
C IMPLICIT NONE 
INTEGER MAXDAT, MAXREG, MAXSET 
PARAMETER (MAXDAT = 50, MAXREG = 3, MAXSET = 5) 
COMMON IOUT, ELAS(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET), 
& LNSTRE(MAXDAT, 0:MAXSET), RFNP(MAXREG), 
& RFSTRP(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET), 
& RFNP(MAXREG), RFSTRP(MAXDAT, MAXREG) 
C LIST OF VARIABLES 
C ELAS() 2-D ARRAY CONTAINING ELASTIC STRAIN COMPONENTS (%) 
C IOUT OUTPUT DUMP CONTROLLER
C CONTROLS DO LOOP FOR EACH DATA SET
C
K CONTROLS DO LOOP FOR EACH POINT IN A REGION
LNF() 3-D ARRAY CONTAINING LN(RAWN())
LNSTRE() 3-D ARRAY CONTAINING LN(SE()) OR LN(STRE())
LNSTRP() 3-D ARRAY CONTAINING LN(SP()) OR LN(STRP())
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
NF() 2-D ARRAY CONTAINING RAWN() (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
NSETS NUMBER OF RELATED MATERIAL S/N DATA SETS
NUM() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
PLAS() 2-D ARRAY CONTAINING PLASTIC STRAIN COMPONENTS (%)
RAWNF() 2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N DATA SETS
RFNP() 1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
RFSTRE() 2-D ARRAY CONTAINING ELASTIC STRAIN POINTS (%), SE() OR STRE() FOR THE SPECIFIC MATERIAL S/N DATA SET
RFSTRP() 2-D ARRAY CONTAINING PLASTIC STRAIN POINTS (%), SP() OR STRP(), FOR THE SPECIFIC MATERIAL S/N DATA SET

C INITIALIZE VARIABLES

RFNP(1) = 0
DO 50 J = 0, NSETS
NP(J,1) = 0
50 CONTINUE
DO 75 K = 1, MAXDAT
RFSTRP(K,1) = 0.0
RFSTRE(K,1) = 0.0
NF(K,1) = 0.0
75 CONTINUE
DO 100 J = 0, NSETS
DO 150 K = 1, MAXDAT
LNSTRP(K,J,1) = 0.0
LNSTRE(K,J,1) = 0.0
LNNF(K,J,1) = 0.0
150 CONTINUE
100 CONTINUE
C BEGIN CALCULATIONS
C PREPARE NP, LNSTRP, LNSTRE, AND LNNF FOR USE BY SUBROUTINE SW2SU2
IF (IOUT .EQ. 10) WRITE(8,*) 'NSETS = ', NSETS
DO 200 J = 0, NSETS
NP(J,1) = NUM(J)
IF (IOUT .EQ. 10)
& WRITE(8,*) 'J = ', J, ' NUM = ', NUM(J), ' NP = ', NP(J,1)
DO 250 K = 1, NUM(J)
LNSTRP(K,J,1) = ALOG(PLAS(K,J))
LNSTRE(K,J,1) = ALOG(ELAS(K,J))
LNNF(K,J,1) = ALOG(RAWN(K,J))
IF (IOUT .EQ. 10) THEN
WRITE(8,*) 'K = ', K
WRITE(8,*) 'PLAS = ', PLAS(K,J), ' LNSTRP = ', LNSTRP(K,J,1)
WRITE(8,*) 'ELAS = ', ELAS(K,J), ' LNSTRE = ', LNSTRE(K,J,1)
WRITE(8,*) 'RAWN = ', RAWN(K,J), ' LNNF = ', LNNF(K,J,1)
ENDIF
250 CONTINUE
200 CONTINUE
C PREPARE RFNP, RFSTRP, RFSTRE, AND NF FOR USE BY SUBROUTINE EXPCTD
RFNP(1) = NUM(0)

7 - 426
IF (IOUT .EQ. 10) WRITE(8,*), 'NUM = ', NUM(0), ' RFNP = ', RFNP(1)

DO 300 K = 1, NUM(0)
RFSTRP(K,1) = PLAS(K,0)
RFSTRE(K,1) = ELAS(K,0)
NF(K,1) = RAWNF(K,0)
IF (IOUT .EQ. 10) THEN
WRITE(8,*), 'K = ', K
WRITE(8,*), 'PLAS = ', PLAS(K,0), ' RFSTRP = ', RFSTRP(K,1)
WRITE(8,*), 'RFSTRP = ', RFSTRP(K,1)
ELSE
ENDIF

300 CONTINUE
RETURN
END

C SUBROUTINE PECOMP CONTROLS THE CALCULATIONS FOR THE STRAIN DECOMPOSITION
C PORTION OF THE INFORMATION AGGREGATION MODEL FOR THE STRAIN FORMULATION
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 PECOMP (NSETS, DESCRP, NDC, SP, SE, NPTS, RAWSTR,
& RAWNF, NTENS, TNSILE, KP, KE, MP, ME,
& STRP, STRE)
C INPUTS: NSETS, DESCRP, NDC, SP, SE, NPTS, RAWSTR, RAWNF, NTENS,
C OUTPUTS: STRP, STRE, KP, KE, MP, ME
C SUBPROGRAMS: TRMNAT
C IMPLICIT NONE
INTEGER MAXDAT, MAXSET, MAXTNS
PARAMETER (MAXDAT = 50, MAXSET = 5, MAXTNS = 5)
COMMON /OUT,
& I, IOUT, J, M, NDC(0:MAXSET), NPTS(0:MAXSET), NSETS,
& NTENS(0:MAXTNS)
REAL ALPHA(MAXDAT, 0:MAXSET), KE, KP, KTERM,
& LAMBDA(MAXDAT, 0:MAXSET), ME, MP, MTERM,
& RAWN(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
& SE(MAXDAT, 0:MAXSET), SP(MAXDAT, 0:MAXSET),
& STRE(MAXDAT, 0:MAXSET), STRP(MAXDAT, 0:MAXSET),
& TNSILE(0:MAXSET, MAXTNS)
CHARACTER*40 DESCRP(0:MAXSET)

LIST OF VARIABLES
ALPHA() 2-D ARRAY CONTAINING THE RATIOS OF SP TO SE FOR EACH
TOTAL STRAIN POINT
DESCRP() 1-D ARRAY CONTAINING DESCRIPTIONS OF EACH S/N DATA SET
I OUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH POINT IN A DATA SET
KE S-N CURVE LOCATION PARAMETER, KE, FOR THE ELASTIC STRAIN
S-N CURVE
KP S-N CURVE LOCATION PARAMETER, KP, FOR THE PLASTIC STRAIN
S-N CURVE
KTERM EQUAL TO KP / KE

7 - 427
LAMBDA 2-D ARRAY CONTAINING VALUES FOR THE PLASTIC FRACTION OF
THE TOTAL STRAIN
M CONTROLS DO LOOP FOR EACH TENSILE TEST DATA POINT
MAXDAT MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
MAXSET MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
MAXTNS MAXIMUM NUMBER OF TENSILE TESTS PER DATA SET ALLOWED
ME S/N CURVE SHAPE PARAMETER, Me, FOR THE ELASTIC STRAIN S/N CURVE
MP S/N CURVE SHAPE PARAMETER, Mp, FOR THE PLASTIC STRAIN S/N CURVE
MTERM EQUAL TO -((1 / MP) - (1 / ME))
NDC 1-D ARRAY CONTAINING Number of given Decomposed STRAIN
POINTS FOR EACH DATA SET
NPTS() 1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
NTENS() 1-D ARRAY CONTAINING NUMBER OF TENSILE TEST DATA POINTS
FOR EACH DATA SET
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 RAW TOTAL
STRAIN DATA (%) FOR ALL S/N DATA SET
SE() 2-D ARRAY CONTAINING GIVEN ELASTIC STRAIN COMPONENTS (%)
SP() 2-D ARRAY CONTAINING GIVEN PLASTIC STRAIN COMPONENTS (%)
STRP() 2-D ARRAY CONTAINING CALCULATED ELASTIC STRAIN COMPONENTS (%)
STRF() 2-D ARRAY CONTAINING CALCULATED PLASTIC STRAIN COMPONENTS (%)
TNSILE() 2-D ARRAY CONTAINING TENSILE TEST DATA -- INDEXED FOR EACH
DATA SET

C INITIALIZE NECESSARY ARRAYS
DO 25 J = 0, MAXSET
DO 50 I = 1, MAXDAT
   ALPHA(I,J) = 0.0
LAMBDA(I,J) = 0.0
50 CONTINUE
25 CONTINUE

C BEGIN CALCULATIONS
KTERM = KP / KE
MTERM = -((1.0 / MP) - (1.0 / ME))
IF (IOUT .EQ. 10) THEN
   WRITE(8,*) 'KP = ', KP, ' KE = ', KE, ' KTERM = ', KTERM
   WRITE(8,*) 'MP = ', MP, ' ME = ', ME, ' MTERM = ', MTERM
   WRITE(8,*) 'NSETS = ', NSETS
ENDIF
DO 100 J = 0, NSETS
IF (IOUT .EQ. 10) THEN
   WRITE(8,*) 'J = ', J,
   WRITE(8,*) 'NDC = ', NDC(J),
   WRITE(8,*) 'NPTS = ', NPTS(J), ' NTENS = ', NTENS(J)
ENDIF

C IF GIVEN COMPONENT DATA EXITS, NO DECOMPOSITION IS REQUIRED
DO 150 I = 1, NDC(J)
   STRP(I,J) = SP(I,J)
   STRF(I,J) = SE(I,J)
IF (IOUT .EQ. 10) THEN
   WRITE(8,*) 'I = ', I,
   WRITE(8,*) 'SP = ', SP(I,J), ' STRP = ', STRP(I,J)
   WRITE(8,*) 'SE = ', SE(I,J), ' STRE = ', STRF(I,J)
ENDIF
150 CONTINUE

C DECOMPOSE REMAINING TOTAL STRAIN DATA
DO 200 I = (NDC(J) + 1), NPTS(J)
   ALPHA(I,J) = KTERM * RAWNF(I,J) ** MTERM
   LAMBDA(I,J) = ALPHA(I,J) / (1.0 + ALPHA(I,J))
   STRP(I,J) = LAMBDA(I,J) * RAWSTR(I,J)
200 CONTINUE
STRE(I,J) = RAWSTR(I,J) - STRP(I,J)
IF (IOUT .EQ. 10) THEN
  WRITE(8,*) 'I = ', I, ' RAWSTR = ', RAWSTR(I,J),
&
  WRITE(8,*) 'ALPHA = ', ALPHA(I,J),
&
  WRITE(8,*) 'LAMBDA = ', LAMBDA(I,J)
ENDIF

WRITE(8,*) 'STRP = ', STRP(I,J), ' STRE = ', STRE(I,J)

100 CONTINUE

C GENERATE ELASTIC COMPONENTS FOR TENSILE TEST DATA
C AND CALCULATE TOTAL STRAIN VALUE

IF ((NPTS(J) + NTENS(J)) .GT. MAXDAT) THEN
  WRITE(8,*) 'ERROR: EXCEEDED MAXIMUM NUMBER OF POINTS DUE ',
  'TO ADDITION OF TENSILE DATA IN DATA SET ', J
  CALL TRMNAT
ENDIF

I = NPTS(J)
DO 250 M = 1, NTENS(J)
  I = I + 1
  ALPHA(I,J) = KTERM
  STRP(I,J) = TNSILE(J,M)
  STRE(I,J) = TNSILE(J,M) / ALPHA(I,J)
  RAWNF(I,J) = 1.0
  RAWSTR(I,J) = STRP(I,J) + STRE(I,J)
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*) 'M = ', M, ' I = ', I
    WRITE(8,*) 'TNSILE = ', TNSILE(J,M),
    ' ALPHA = ', ALPHA(I,J),
    ' LAMBDA = ', LAMBDA(I,J)
    WRITE(8,*) 'STRP = ', STRP(I,J),
    ' RAWSTR = ', RAWSTR(I,J),
    ' RAWNF = ', RAWNF(I,J)
  ENDIF
  250 CONTINUE

NPTS(J) = NPTS(J) + NTENS(J)
100 CONTINUE

C WRITE RESULTS TO DUMP

WRITE(7,900)
WRITE(7,905) DESCRP(J), NPTS(J)
DO 300 I = 1, NPTS(0)
  WRITE(7,910) RAWNF(I,0), RAWSTR(I,0), STRP(I,0), STRE(I,0)
300 CONTINUE

WRITE(7,915) NSETS
DO 400 J = 1, NSETS
  WRITE(7,905) DESCRP(J), NPTS(J)
  DO 500 I = 1, NPTS(J)
    WRITE(7,910) RAWNF(I,J), RAWSTR(I,J), STRP(I,J), STRE(I,J)
  500 CONTINUE
400 CONTINUE

900 FORMAT(///.2X,'ESTIMATED STRAIN DECOMPOSITION',
&    ///.2X,'SPECIFIC MATERIAL')
905 FORMAT(///.2X,'DESCRIPTION:',.2X,A40,
&    ///.2X,'NUMBER OF DATA POINTS:',.2X,I2,
&    ///.7X,'LIFE',5X,'TOTAL STRAIN',5X,'PLASTIC STRAIN',
&    ///.5X,'ELASTIC STRAIN/')
910 FORMAT(6X,F6.0,6X,F7.4,11X,F8.5,10X,F8.5)
915 FORMAT(///.2X,'RELATED MATERIALS',
&    ///.2X,'NUMBER OF DATA SETS:',2X,I2,/)
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

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, 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, TRANSFM, SMANVAR, 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, M_4(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

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 OUTF OUTPUT DUMP CONTROLLER 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) MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION MU() 1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR EACH REGION NBND() 1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG REGIONS OF INTEREST NF() 2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS NP TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET NPTS() 1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE SPECIFIC MATERIAL S/N DATA SET NUMREG NUMBER OF REGIONS OF INTEREST 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 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

7-430
C STR() 2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
DATA SET BROKEN INTO REGIONS (PSI OR %)
C SZ2  SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C VARY  CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
1 - S-N RANDOMNESS ONLY; 2 - UNIFORM VARIATION;
3 - TRUNCATED NORMAL VARIATION
C ZROREG  ZERO REGION -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C ZZ()  1-D ARRAY CONTAINING TRANSFORMED S/N DATA, Z = F(STR,NF,NBND,MM)

C OBTAIN THE VALUES OF M FOR EACH REGION
IF (VARY .LE. 2) THEN
UNIFORM OR NO VARIATION IN M IS DESIRED
CALL FINDM (RAND, NUMREG, RANGEM, MM)
ELSE
NORMAL VARIATION IN M IS DESIRED
CALL FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
ENDIF

C TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)
CALL TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)

C CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)
CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C CALCULATE THE VALUES FOR k AND BETA0 FROM THE SAMPLE MEAN
C AND VARIANCE
CALL KBETA (MEANZ, SZ2, K, BZERO)

C CALCULATE THE VALUE OF K FOR EACH REGION WHERE A = K ** 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
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: ',14,5X,'BETA0 = ',F8.4,
& ///,2X,'REGION ',7X,'m ',15X,'K ',9X,'LIFE BOUND ',5X,
& 'STRESS BOUND',/)
910 FORMAT(5X,I1,5X,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

C IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES
OUTPUT DUMP CONTROLLER
MAXREG CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
PICK() 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND RANGEM(2,L) IS THE UPPER BOUND
X UNIFORM(0, 1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED OFF THE RANGE ON M

C INITIALIZE MM()
DO 50 L = 0, MAXREG
     MM(MAXREG) = 0.0
50 CONTINUE

C BEGIN CALCULATIONS
DO 100 L = 1, NUMREG

   PICK(1) = 0.0
   PICK(2) = 0.0

   IF (RANGEM(2,L) .EQ. 0.0) THEN
      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)
      END IF
   END IF
100 CONTINUE
SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE UNIFORMLY DISTRIBUTED RANDOM NUMBERS


PROGRAMMER: L. GRONDAHL, L. NEWLIN

DATE: DEC87

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

MATCHR 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

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

RETURN

END
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, ' RAXN = ', RANDIV, 
& ' RANT = ', RANT, ' RANSUB = ', RANSUB, ' RAND = ', RAND, 
& ' FRAC = ', FRAC))
RETURN
END

C NOTES: IOUT=2 DUMPS TO SCREEN

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

C SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 7JUN88 COMMENTS: 13FEB89
C VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
SUBROUTINE FINDMN (RAND, NUMREG, MU, SIG, RANGEM, MM)
C INPUTS: RAND, NUMREG, MU, SIG, RANGEM
C OUTPUTS: MM
C SUBPROGRAMS: NORMG, TRMNAT
C
IMPLICIT NONE
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL MM(0:MAXREG), MU(MAXREG), PICK(2), RANGEM(2, MAXREG), 
& SIG(MAXREG), X
DOUBLE PRECISION RAND

LIST OF VARIABLES

IOUT OUTPUT DUMP CONTROLLER
L CONTROLS DO LOOP FOR EACH REGION
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
MU() 1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
NUMREG NUMBER OF REGIONS OF INTEREST
PICK() 1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
RAND RANDOM NUMBER SEED
RANGEM() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
FOR EACH REGION -- RANGEM(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

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)
    ENDIF

    IF (IOUT .EQ. 10) THEN
        WRITE(8,*), 'RANGEM(1,L) =', RANGEM(1,L), 'MM(L) =', MM(L)
    ELSEIF (L .EQ. 1) THEN
        WRITE(8,*), 'RANGEM(1,L) =', RANGEM(1,L), 'MM(L) =', MM(L)
    ENDIF

10    CALL NORMGN (RAND, MU(L), SIG(L), X)

100   CONTINUE

RETURN
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***************************************************************************

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

C***************************************************************************
The random variates are generated using the "Direct Method"

Abramowitz, 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)

IMPLICIT NONE

COMMON IOUT

DOUBLE PRECISION RAND
REAL FRAC, MU, PI, SIGMA, X, UI, U2, Z1, Z2
PARAMETER (PI = 3.1415926536)
INTEGER IOUT

LIST OF VARIABLES

FRAC UNIFORM(0,1) RANDOM VARIATE
IOUT OUTPUT DUMP CONTROLLER
MU MEAN OF NORMAL DISTRIBUTION
RAND RANDOM NUMBER SEED
SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION
X NORMAL RANDOM VARIATE
U1 UNIFORM RANDOM NUMBER U(0,1)
U2 UNIFORM RANDOM NUMBER U(0,1)
Z1 NORMAL RANDOM NUMBER ON N(0,1)
Z2 NORMAL RANDOM NUMBER ON N(0,1)

IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(*,*) '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(*,*) '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(*,*) 'Z1 =', Z1, ' Z2 =', Z2, ' X =', X
RETURN

SUBROUTINE TRANSFM PERFORMS THE CALCULATIONS NECESSARY TO TRANSFORM
THE S/N DATA INTO THE VARIABLE Z = Ln(X)

PROGRAMMER: L. NEWLIN
DATE: CODE: 7JUN88 COMMENTS: 13JUL89
SUBROUTINE TRNSFM (NPTS• STR, NF• NUMREG• MM• NBND• NP• ZZ)
C INPUTS: NPTS, STR, NF, NUMREG, MM, NBND
C OUTPUTS: NP, ZZ
C
LIST OF VARIABLES
CONTROLS DO LOOP FOR EACH DATA POINT
IOUT OUTPUT DUMP CONTROLLER
K CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
L CONTROLS DO LOOP FOR EACH REGION
MAXDAT MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED
MML 1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
MM() 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
RATSTR() 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
C INITIALIZE VARIABLES
NP = 0
DO 50 I = 1, MAXDAT
    ZZ(I) = 0.0
50 CONTINUE
C BEGIN CALCULATIONS

   DO 100 L = 1, NUMREG
       MML = MM(L)
       IF (IOUT .EQ. 10) WRITE(8,*)('L = ', L, ', MM = ', MM(L), ', MML = ', MML, ', NPTS = ', NPTS(L))
       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 = ', NP, ', NF = ', NF(K,L), ', STR = ', STR(K,L), ', ZZ = ', ZZ(NP))
       200 CONTINUE
   100 CONTINUE

   DO 300 LL = 2, L
       ZZ(NP) = ZZ(NP) + ALOG(NBND(LL-1))
       IF (IOUT .EQ. 10) WRITE(8,*)('LL = ', LL, ', NBND(LL-1) = ', NBND(LL-1), ', MM(LL-1) = ', MM(LL-1), ', MM(LL) = ', MM(LL), ', ZZ = ', ZZ(NP))
   300 CONTINUE
C SUBROUTINE SMNVAR CALCULATES THE Sample Mean and Variance OF 
Z = F(STR, NF, NBND, MM)
PROGRAMMER: L. NEWLIN
DATE: CODE: 24AUG87 COMMENTS: 13JUL89
VERSION: MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
V8.3, V8.4, V8.5

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)

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
SUBROUTINE KBETA (MEANZ, SZ2, K, BZERO)

C INPUTS: MEANZ, SZ2
C OUTPUTS: K, BZERO
C IMPLICIT NONE
REAL PI
PARAMETER (PI = 3.1415926536)
COMMON IOUT
INTEGER IOUT
REAL BZERO, K, MEANZ, SZ, SZ2

C LIST OF VARIABLES
C BZERO VALUE OF WEIBULL PARAMETER, Beta0, CHARACTERIZING THE SPECIFIC MATERIAL S/N DATA SET
C IOUT OUTPUT DUMP CONTROLLER
C K VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C MEANZ SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C PI SELF EXPLANATORY CONSTANT
C SZ SZ2 ** 0.5
C SZ2 SAMPLE VARIANCE OF THE TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)

C PERFORM CALCULATIONS
SZ = SZ2 ** 0.5
BZERO = PI / (SZ * (6.0 ** 0.5))
K = MEANZ

C DATA DUMP STATEMENTS
IF (IOUT .EQ. 10) THEN
WRITE(8,*) 'SZ2 = ', SZ2,
WRITE(8,*) 'SZ = ', SZ
ENDIF
RETURN
END
SUBROUTINE FINDK calculates the value of \( k \), where \( A = k^{**m} \) for each region.

Programmer: L. Newlin
Date: 7Jun88

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 \( A = k^{**m} \)
BZERO Value of Weibull parameter, betao, characterizing specific material data base
GAMMA Euler's constant
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)
IF (IOUT .EQ. 10) WRITE(7,*) 'REGION: 1, K =', BIGK(1)
C
C Calculate \( k \) for remaining regions
DO 100 L = 2, NUMREG
  BIGK(L) = BIGK(L-1) * NBND(L-1)
  IF (IOUT .EQ. 10) WRITE(7,*) 'REGION: L, K =', BIGK(L)
100 CONTINUE
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
MIZATION V8.4, V8.5

SUBROUTINE FINDSB (NUMREG, ZROREG, NBND, BIGK, MM, SBND)
C INPUTS: NUMREG, ZROREG, NBND, BIGK, MM
C OUTPUTS: SBND
C
C IMPLICIT NONE
C INTEGER MAXREG
C PARAMETER (MAXREG = 3)
C COMMON IOUT
C INTEGER IOUT, L, NUMREG, ZROREG
C REAL & BIGK(0:MAXREG), MM(0:MAXREG), NBND(0:MAXREG),
C SBND(0:MAXREG)

LIST OF VARIABLES

BIGK() 1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
IOUT OUTPUT DUMP CONTROLLER
L MAXREG CONTROLS DO LOOP FOR EACH REGION
NBND() 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
NUMREG NUMBER OF REGIONS OF INTEREST
ZROREG ZROREG -- VALUES CHOSEN TO FACILITATE REGION DO LOOP

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

7-441
SUBROUTINE ADJSTM CONTROLS THE CALCULATIONS NECESSARY TO CONSTRAIN 
Me TO BE GREATER THAN THE SELECTED Mp BY ADJUSTING THE POSTERIOR 
RANGE ON Me 
PROGRAMMER: L. NEWLIN 
DATE: 7JUN88 
VERSION: MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5 
Copyright (C) 1990, California Institute of Technology. 
U.S. Government Sponsorship under NASA Contract NAS7-918 
is acknowledged. 

SUBROUTINE ADJSTM (MP, RANGME, MEBND) 

INPUTS: MP, RANGME 
OUTPUTS: MEBND 

IMPLICIT NONE 
INTEGER MAXREG 
PARAMETER (MAXREG = 3) 
COMMON IOUT 
INTEGER IOUT, L 
REAL MEBND(2, MAXREG), MP(0:MAXREG), RANGME(2, MAXREG) 

LIST OF VARIABLES 
IOUT OUTPUT DUMP CONTROLLER 
L CONTROLS DO LOOP FOR EACH POINT IN A REGION 
MAXREG MAXIMUM NUMBER OF REGIONS ALLOWED 
MEBND() 2-D ARRAY CONTAINING ALLOWABLE RANGE ON Me, FOR ELASTIC 
COMPONENTS, AFTER SELECTION OF Mp 
MP() 1-D ARRAY CONTAINING SELECTED VALUE OF Mp FOR PLASTIC COMPONENTS 
RANGME() 2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON Me 
FOR ELASTIC COMPONENTS -- RANGME(1,1) IS THE LOWER 
BOUND AND RANGME(2,1) IS THE UPPER BOUND 

DO 50 L = 1, MAXREG 
MEBND(1,L) = 0.0 
MEBND(2,L) = 0.0 
50 CONTINUE 
MEBND(1,1) = MAX(MP(1), RANGME(1,1)) 
MEBND(2,1) = RANGME(2,1) 
RETURN 
END 

C THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
SUBROUTINE WEIBGN (BETA, RAND, WEIB)

C INPUTS:  BETA, RAND
C OUTPUTS: WEIB
C SUBPROGRAMS: RANDOM

C IMPLICIT NONE
C COMMON IOUT
C INTEGER IOUT
C REAL ARG, BETA, ETA, FRAC, WEIB
C DOUBLE PRECISION RAND

C 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

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,*('BETA = ', ETA, 'ETA = ', ETA, &
' FRAC = ', FRAC, ' ARG = ', ARG, ' WEIB = ', WEIB
RETURN
END

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

SUBROUTINE KOMO CALCULATES K0 AND Mc 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: AUG91
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.

7 - 443
SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
& FACTR, NUMREG)
C
C INPUTS:  SZERO, BIGK, MM, NBND, TRSBND, FACTR
C OUTPUTS: TRBIGK, MM, TRSBND
C
C IMPLICIT NONE
C
INTEGER MAXREG
PARAMETER (MAXREG = 3)
COMMON IOUT
INTEGER IOUT, L, NUMREG
REAL BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NI_ND(0:MAXREG),1 SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)

LIST OF VARIABLES

I-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 ALLOWED
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 S_ < 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
TRBIGK(L) = BIGK(L) * SCLK
TRSBND(L) = TRSBND(L) * SCLK
100 CONTINUE ELSE
TRBIGK(0) = SZERO/FACTR
MM(0) = MM(1) * ((ALOG (BIGK(1)) - ALOG (TRSBND(0))) & + ALOG (FACTR)) / (ALOG (SZERO) - ALOG (TRSBND(0))))
ENDIF
C
IF (IOUT .EQ. 10) THEN
WRITE(8,*) 'SZERO = ', SZERO, ' BIGKO = ', TRBIGK(0)
WRITE(8,*) 'FACTOR = ', FACTR, ' BIGKI = ', TRBIGK(1)
WRITE(8,*) 'MH1 = ', MM(1), ' MMO = ', MM(0)
ENDIF
C
RETURN
END

CFUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C PROGRAMMER: L. NEWLIN
C DATE: 10FEB89

7 - 444
REAL FUNCTION GTLIFE (S, MM, BIGK, PHI, KRATIO, LNZ, SBND, & ZROREG, NUMREG, SZERO)

C INPUTS: S, MM, BIGK, PHI, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO

C IMPLICIT NONE

INTEGER IOUT, L, MAXREG, NUMREG, ZROREG
PARAMETER (MAXREG = 3)
COMMON 1OUT
REAL & ZROREG, NUMREG, SZERO
BIGK(0:MAXREG), GETLIF, KRATIO, LNZ, MM(0:MAXREG), PHI, S, SBND(0:MAXREG), SZERO, TEMP

C
C
C BIGK()
C
C GETLIF
C
C IOUT
C KRATIO
C MAXREG
C LNZ
C MAXIMUM NUMBER OF REGIONS ALLOWED
C MM() 1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C NUMREG NUMBER OF REGIONS OF INTEREST
C PHI WEIBULL(BETAO, ETAo) GENERATED RANDOM VARIATE
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 CONTAINED IN NBND()
C SZERO STRESS TENSILE TEST 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
TEM = MM(L) * (ALOG(BIGK(L)) - ALOG(S) + ALOG(PHI) & + ALOG (KRATIO) + LNZ)
IF (TEM .GT. 86.0) THEN
TEM = 86.0
ENDIF
GETLIF = EXP (TEM)
GOTO 150
ENDIF
100 CONTINUE
ENDIF
150 CONTINUE

GTLIFE = GETLIF
C FUNCTION GTLIF2 CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR VALUE OF TOTAL STRAIN BASED UPON THE MATERIALS CHARACTERIZATION COMPONENT STRAIN S/N EQUATION. THE SOLUTION IS FOUND USING A NEWTON'S METHOD ITERATION SUBROUTINE AND THE LIFE CORRESPONDING TO THE ELASTIC STRAIN IS USED AS THE INITIAL LIFE VALUE, No.

PROGRAMMER: L. NEWLIN
DATE: CODE: 15FEB89 COMMENTS: 13JUL89

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

REAL FUNCTION GTLIF2 (STR, KP, KE, MP, ME, PHI, KRATIO, Z)

INPUTS: STR, KP, KE, MP, ME, PHI, KRATIO, Z
OUTPUTS: GTLIF2
SUBPROGRAMS: NEWTON, TRMNAT

IMPLICIT NONE
COMMON IOUT
INTEGER IEND, IER, IOUT
REAL DERF, EPS, F, GETLIF, KE, KP, KRATIO, ME, MP, NZERO,
PHI, STR, Z

LIST OF VARIABLES

DERF RESULTANT VALUE OF DERIVATIVE AT ROOT
EPS INPUT VALUE WHICH SPECIFIES THE UPPER BOUND OF THE ERROR OF THE RESULT
F RESULTANT FUNCTION VALUE AT ROOT
GETLIF VALUE TO BE ASSIGNED TO GTLIF2 -- CYCLES TO FAILURE FOR THE REQUIRED STRAIN LEVEL
IEND MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED
IER RESULTANT ERROR PARAMETER, CODED -- 0 - NO ERROR; 1 - NO CONVERGENCE AFTER IEND ITERATION STEPS; 2 - AT ANY ITERATION STEP DERIVATIVE DERF WAS EQUAL TO ZERO
IOUT OUTPUT DUMP CONTROLLER
KE S/N CURVE LOCATION PARAMETER, Ke, FOR THE ELASTIC STRAIN S/N CURVE
KP S/N CURVE LOCATION PARAMETER, KP, FOR THE PLASTIC STRAIN S/N CURVE
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
ME S/N CURVE SHAPE PARAMETER, Me, FOR THE ELASTIC STRAIN S/N CURVE
MP S/N CURVE SHAPE PARAMETER, Mp, FOR THE PLASTIC STRAIN S/N CURVE
NZERO EQUAL TO THE LIFE FROM THE ELASTIC PORTION OF THE CURVE USED AS THE INITIAL VALUE IN THE NEWTON'S METHOD ITERATION
PHI WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIABLE
STR VALUE OF TOTAL STRAIN (%) FOR WHICH CYCLES TO FAILURE IS DESIRED
Z LOG-NORMAL(0,PVAR) RANDOM VARIATE

EPS = 1.0E-6
IEND = 1000

C CALCULATE INITIAL VALUE OF LIFE FOR NEWTON
NZERO = (KE * PHI * KRATIO * Z / STR) ** ME

C LET NEWTON ITERATE TO FIND A SOLUTION
CALL NEWTON (GETLIF, F, DERF, NZERO, EPS, IEND, IER, STR, & KP, KE, MP, ME, PHI, KRATIO, Z)

C NOW CHECK IER FOR ANY PROBLEMS AND ASSIGN GTLIF2 ACCORDINGLY
IF (IER .EQ. 0) THEN
  GTLIF2 = GETLIF
END IF (IER .EQ. 1) THEN
  WRITE(8,*) 'NO CONVERGENCE AFTER SPECIFIED NO. ITERATION STEPS'
  CALL TRNAT
ELSEIF (IER .EQ. 2) THEN
  WRITE(8,*) 'DERIVATIVE EQUAL TO ZERO'
  CALL TRNAT
ELSE
  WRITE(8,*) 'ERROR CODE INCORRECTLY SPECIFIED'
ENDIF

RETURN
END

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

C SUBROUTINE FCT IS USED BY SUBROUTINE NEWTON TO CALCULATE THE VALUE
C OF THE FUNCTION AND ITS DERIVATIVE AT THE VALUE 'LIFE', IN ORDER TO
C FIND THE SOLUTION OF THE COMPONENT STRAIN S/N CURVE
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 21JUN88 COMMENTS: 13JUL89
C
C SUBROUTINE FCT (LIFE, F, DERF, STR, KP, KE, ME, PHI, KRATIO, & Z)
C INPUTS: LIFE, STR, KP, KE, ME, PHI, KRATIO, Z
C OUTPUTS: F, DERF
C IMPLICIT NONE
COMMON IOUT
INTEGER IOUT
REAL DERF, F, INVME, INVMP, KE, KEPROD, KP, KPPROD, KRATIO, & LIFE, ME, MP, PHI, STR, Z

LIST OF VARIABLES

DERF RESULTANT VALUE OF DERIVATIVE AT ROOT
F RESULTANT FUNCTION VALUE AT ROOT
INVME EQUAL TO 1/ME (INVERSE OF ME)
INVMP EQUAL TO 1/MP (INVERSE OF MP)
IOUT OUTPUT DUMP CONTROLLER
KE S/N CURVE LOCATION PARAMETER, Ke, FOR THE ELASTIC STRAIN
S/N CURVE
KEPROD EQUAL TO KE * PHI * KRATIO * Z
KP S/N CURVE LOCATION PARAMETER, KP, FOR THE PLASTIC STRAIN
S/N CURVE
KPPROD EQUAL TO KP * PHI * KRATIO * Z
KRATIO RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
LIFE VALUE OF N (CYCLES TO FAILURE)
ME S/N CURVE LOCATION PARAMETER, Me, FOR THE ELASTIC STRAIN
S/N CURVE
MP S/N CURVE LOCATION PARAMETER, Mp, FOR THE PLASTIC STRAIN
S/N CURVE
THE FOLLOWING SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE RTNH,
TAKEN FROM "MATHEMATICS -- ROOTS OF NONLINEAR EQUATIONS" (SEE P. 220)
PURPOSE
TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM \( F(X) = 0 \) BY MEANS
OF NEWTON'S ITERATION METHOD.
USAGE
CALL NEWTON (X, F, DERF, XST, EPS, IEND, IER)
DESCRIPTION OF PARAMETERS
X RESULTANT ROOT OF EQUATION \( F(X) = 0 \).
F RESULTANT FUNCTION VALUE AT ROOT X.
DERF RESULTANT VALUE OF DERIVATIVE AT ROOT X.
FCT NAME OF THE EXTERNAL SUBROUTINE USED. IT COMPUTES TO GIVEN
ARGUMENT X FUNCTION VALUE F AND DERIVATIVE DERF. ITS
PARAMETER LIST MUST BE X, F, DERF.
XST INPUT VALUE WHICH SPECIFIES THE INITIAL GUESS OF THE ROOT X.
EPS INPUT VALUE WHICH SPECIFIES THE UPPER BOUND OF THE ERROR OF
RESULT X.
IEND MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED
IER RESULTANT ERROR PARAMETER CODED AS FOLLOWS
IER = 0 -- NO ERROR
IER = 1 -- NO CONVERGENCE AFTER IEND ITERATION STEPS
IER = 2 -- AT ANY ITERATION STEP DERIVATIVE DERF WAS
EQUAL TO ZERO
REMARKS
THE PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER = 2 IF AT
ANY ITERATION STEP DERIVATIVE OF \( F(X) \) IS EQUAL TO 0. POSSIBLY THE
PROCEDURE WOULD BE SUCCESSFUL IF IT IS STARTED ONCE MORE WITH
ANOTHER INITIAL GUESS XST.
SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
THE EXTERNAL SUBROUTINE FCT(X, F, DERF) MUST BE FURNISHED BY THE USER
METHOD
SOLUTION OF EQUATION \( F(X) = 0 \) IS DONE BY MEANS OF NEWTON'S ITERATION
METHOD, WHICH STARTS AT THE INITIAL GUESS XST OF A ROOT X. CONVERGENCE IS
QUADRATIC IF THE DERIVATIVE OF \( F(X) \) AT ROOT X IS NOT EQUAL

INVMP = 1.0 / MP
INVME = 1.0 / ME
KPPROD = KP * PHI * KRATIO * Z
KEPROD = KE * PHI * KRATIO * Z
F = KPPROD * LIFE ** (-INVMP) + KEPROD * LIFE ** (-INVME) - STR
DERF = - (KPPROD * INVMP * LIFE ** (-1.0 - INVMP)
& + KEPROD * INVME * LIFE ** (-1.0 - INVME))
IF (IOUT .EQ. 5) THEN
WRITE(8,*) 'INVMP = ', INVMP, ' INVME = ', INVME
WRITE(8,*) 'KPPROD = ', KPPROD, ' KEPROD = ', KEPROD
WRITE(8,*) 'F = ', F, ' DERF = ', DERF
WRITE(8,*) 'LIFE = ', LIFE
ENDIF
RETURN
END
SUBROUTINE NEWTON (X, F, DERF, XST, EPS, IEND, IER, STR, & KP, KE, MP, ME, PHI, KRATIO, Z)
C IMPLICIT NONE
COMMON IOUT
INTEGER I, IEND, IER, IOUT
REAL A, DERF, DX, EPS, F, TOL, TOLF, X, XST, & STR, KP, KE, MP, ME, PHI, KRATIO, Z

C PREPARE ITERATION
IER = 0
X = XST
TOL = X
IF (IOUT .EQ. 5) WRITE(8,*) 'XST =', XST, 'X =', X, 'TOL =', TOL
CALL FCT (TOL, F, DERF, STR, KP, KE, MP, ME, PHI, KRATIO, Z)
TOLF = 100. * EPS
IF (IOUT .EQ. 5)
& WRITE(8,*) 'TOL =', TOL, 'F =', F, 'DERF =', DERF, 'TOLF =', TOLF

C START ITERATION LOOP
DO 6 I = 1, IEND
IF (F) 1, 7

C EQUATION IS NOT SATISFIED BY X
1 IF (DERF) 2, 8, 2

C ITERATION IS POSSIBLE
2 DX = F / DERF
X = X - DX
TOL = X
IF (IOUT .EQ. 5)
& WRITE(8,*) 'I =', I, 'DX =', DX, 'X =', X, 'TOL =', TOL
CALL FCT (TOL, F, DERF, STR, KP, KE, MP, ME, PHI, KRATIO, Z)

C TEST ON SATISFACTORY ACCURACY
TOL = EPS
A = ABS(X)
IF (A - 1. ) 4, 4, 3
3 TOL = TOL * A
4 IF (ABS(DX) - TOL) 5, 5, 6
5 IF (ABS(F) - TOLF) 7, 7, 6
6 CONTINUE

C END OF ITERATION LOOP

C NO CONVERGENCE AFTER IEND ITERATION STEPS. ERROR RETURN
IER = 1
7 RETURN

C ERROR RETURN IN CASE OF ZERO DIVISOR
8 IER = 2
RETURN
END
C SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
M FOR EACH REGION
PROGRAMMER: L. NEWLIN
DATE: 10FEB88

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

SUBROUTINE SORTM (ALLM, NUMREG, NUM)
C INPUTS:  ALLM, NUMREG, NUM
C OUTPUTS: ALLM
C IMPLICIT NONE
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  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
         INORDR = .TRUE.
         DO 300 I = 1, (NUM - INC)
            IF (ALLM(I,L) .GT. ALLM(I + INC, L)) THEN
               TEMP = ALLM(I,L)
               ALLM(I,L) = ALLM(I + INC, L)
               ALLM(I + INC, L) = TEMP
               INORDR = .FALSE.
            ENDIF
      ENDIF
   300 CONTINUE
      IF (.NOT. INORDR) GOTO 20
      GOTO 10
   20 CONTINUE
   400 CONTINUE

RETURN
END
Section 7.4
Prior Distribution Parameter Estimation Program

The program tree structure, list of subprograms, description of the key variables, and the FORTRAN source listing for the prior failure distribution parameter estimation codes BFIT and ABTFIT, and the assurance calculation code LZERO are given here. The pertinent methodology is given in Section 2.1.1. The overall descriptions of the programs and the flowcharts are given in Section 4.2. The user’s guide for running BFIT, ABTFIT, and LZERO is given in Section 6.4.

7.4.1 BFIT Program

7.4.1.1 List of Subprograms

A list of subprograms and their purposes is given in Table 7-10. The section number where each subprogram is described is given next to the name. The program, subprogram, and file names are indicated by UPPERCASE letters.

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFIT</td>
<td>4.2.2</td>
<td>The main routine that controls the logical flow of the prior failure distribution parameter $\beta$ estimation calculations.</td>
</tr>
<tr>
<td>LLS</td>
<td>4.2.2.1</td>
<td>Solves a linear least squares problem.</td>
</tr>
</tbody>
</table>

7.4.1.2 Description of Variables

A list of variables used in the prior failure distribution parameter $\beta$ estimation code, BFIT, is given in Table 7-11. The variable names are indicated by BOLD UPPERCASE letters; the variable “type” can be interpreted as follows: INT is a standard integer variable, and DRE is a double precision variable. The various array dimensions are defined by using the MAXM parameter.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>DRE</td>
<td>Slope parameter calculated by LLS, estimate of $\beta$.</td>
</tr>
<tr>
<td>DIFFX(MAXM)</td>
<td>DRE</td>
<td>1-D array containing values of the difference between $X(I)$ and MEANX for each point used in the linear regression.</td>
</tr>
</tbody>
</table>
Table 7-11 List of Variables for Program BFIT (Cont’d)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIFFY(MAXM)</td>
<td>DRE</td>
<td>1-D array containing values of the difference between Y(I) and MEANY for each point used in the linear regression.</td>
</tr>
<tr>
<td>END</td>
<td>INT</td>
<td>Ending index position for regression performed by LLS.</td>
</tr>
<tr>
<td>FOFN</td>
<td>DRE</td>
<td>$F(N_i)$ in Equation 2-8.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls DO loop.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>LIFE(MAXM)</td>
<td>DRE</td>
<td>1-D array containing values of $N_i$, Equation 2-8, generated by the PFM.</td>
</tr>
<tr>
<td>LNC</td>
<td>DRE</td>
<td>Intercept parameter calculated by LLS.</td>
</tr>
<tr>
<td>M</td>
<td>INT</td>
<td>Length of F, the number of $(F(N_i), N_i)$ pairs provided by the PFM.</td>
</tr>
<tr>
<td>MAXM</td>
<td>INT</td>
<td>Maximum value for M.</td>
</tr>
<tr>
<td>MEANX</td>
<td>DRE</td>
<td>The sample mean of X.</td>
</tr>
<tr>
<td>MEANY</td>
<td>DRE</td>
<td>The sample mean of Y.</td>
</tr>
<tr>
<td>START</td>
<td>INT</td>
<td>Starting index position for regression performed by LLS.</td>
</tr>
<tr>
<td>SX2</td>
<td>DRE</td>
<td>The sample variance of X.</td>
</tr>
<tr>
<td>SXY</td>
<td>DRE</td>
<td>The sample covariance of X and Y.</td>
</tr>
<tr>
<td>X(MAXM)</td>
<td>DRE</td>
<td>1-D array containing values of $\ln N_i$ corresponding to the $N_i$’s in the array LIFE().</td>
</tr>
<tr>
<td>Y(MAXM)</td>
<td>DRE</td>
<td>1-D array containing values of $Y_i$, Equation 2-8, corresponding to the $N_i$’s in the array LIFE().</td>
</tr>
</tbody>
</table>

7.4.1.3 Program BFIT Listing

```c
C PROGRAM BFIT CONTROLS THE LINEAR LEAST SQUARES CALCULATIONS
C REQUIRED TO ESTIMATE THE PRIOR DISTRIBUTION SHAPE PARAMETER
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 20AUG90 COMMENTS: 20SEP91
C VERSION: 1.1
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918 is acknowledged.

PROGRAM BFIT
C DECLARATION OF VARIABLES
```
INTEGER DUM, END, I, IOUT, M, MAXM, START

PARAMETER (MAXM = 200)

DOUBLE PRECISION B, FOFN, LIFE(MAXM), LNC, Y(MAXM)

LIST OF VARIABLES

B Shape parameter.
DUM Dummy variable used during data entry.
END Ending position for regression loops.
FOFN Controls DO loop.
IOUT Output dump controller.
LIFE() 1-D array containing values of N generated by the PFM.
LNC Intercept parameter.
M The number of (F(N), N) pairs provided by the PFM.
MAXM Maximum value for M.
START Starting position for regression loops.
Y() 1-D array containing values of Y corresponding to the N's in the array LIFE().

INITIALIZE VARIABLES AND READ INPUT PARAMETERS

OPEN (11, FILE = 'BFITD', STATUS = 'OLD')
OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
OPEN (13, FILE = 'BFITO', STATUS = 'NEW')

READ(11,*) IOUT, START, END, M
DO 100 I = 1, M
   READ(12,*) DUM, FOFN, LIFE(I)
   Y(I) = DLOG ( - DLOG (1.0 - FOFN))
100 CONTINUE

PERFORM LINEAR LEAST SQUARES TO FIND SHAPE PARAMETER B

CALL LLS (LIFE, Y, START, END, B, LNC, IOUT)

PRINT RESULTS

WRITE(13,900) B

900 FORMAT (2X,'Copyright (C) 1990, California Institute of ',
 & 'Technology. U.S. Government, /,2X,' 'Sponsorship under ',
 & 'NASA Contract NAS7-918 is acknowledged. \///, 
 & 2X,' 'The solution is ', /,5X,'Beta: ', D12.7)

STOP
END

SUBROUTINE LLS PERFORMS A Linear Least Squares FIT

PROGRAMMER: L. NEWLIN
DATE: 17AUG90
VERSION: BFIT VI, VI.1

SUBROUTINE LLS (LIFE, Y, START, END, B, LNC, IOUT)

INPUTS: LIFE, Y, START, END, IOUT
OUTPUTS: B, LNC

IMPLICIT NONE

INTEGER MAXM

PARAMETER (MAXM = 200)

INTEGER END, I, IOUT, START

DOUBLE PRECISION B, DIFFX(MAXM), DIFFY(MAXM), LIFE(MAXM),
 & LNC, MEANX, MEANY, SX2, SXY, X(MAXM), Y(MAXM)
LIST OF VARIABLES

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Shape parameter.</td>
</tr>
<tr>
<td>DIFFX()</td>
<td>1-D array containing the difference between X and MEANX for each LIFE.</td>
</tr>
<tr>
<td>DIFFY()</td>
<td>1-D array containing the difference between Y and MEANX for each LIFE.</td>
</tr>
<tr>
<td>END</td>
<td>Ending position for regression loops.</td>
</tr>
<tr>
<td>IOUT</td>
<td>Controls DO loops.</td>
</tr>
<tr>
<td>LIFE()</td>
<td>1-D array containing values of the lives generated by the PFM (already sorted).</td>
</tr>
<tr>
<td>LNC</td>
<td>Intercept parameter.</td>
</tr>
<tr>
<td>MAXM</td>
<td>Maximum number of fatigue lives allowed for regression.</td>
</tr>
<tr>
<td>MEANX</td>
<td>Sample mean of X.</td>
</tr>
<tr>
<td>MEANY</td>
<td>Sample mean of Y.</td>
</tr>
<tr>
<td>START</td>
<td>Starting position for regression loops.</td>
</tr>
<tr>
<td>SX2</td>
<td>Sample variance of X.</td>
</tr>
<tr>
<td>SXY</td>
<td>Sample covariance of X and Y.</td>
</tr>
<tr>
<td>X()</td>
<td>( X = \ln \left( \text{Life} \right) )</td>
</tr>
<tr>
<td>Y()</td>
<td>( Y = \ln \left( - \ln (1 - F(\text{Life})) \right) )</td>
</tr>
</tbody>
</table>

INITIALIZE VARIABLES

MEANX = 0.0D0
MEANY = 0.0D0
SX2 = 0.0D0
SXY = 0.0D0
DO 50 I = 1, MAXM
   X(I) = 0.0D0
   DIFFX(I) = 0.0D0
   DIFFY(I) = 0.0D0
50 CONTINUE

BEGIN CALCULATIONS -- FIRST CALCULATE THE SAMPLE MEANS

DO 100 I = START, END
   X(I) = DLOG (LIFE(I))
   MEANX = MEANX + X(I)
   MEANY = MEANY + Y(I)
   IF (IOUT .EQ. 10) THEN
      WRITE(8,*), 'I = ', I, ', LIFE = ', LIFE(I)
      WRITE(8,*), 'X = ', X(I), ', Y = ', Y(I)
      WRITE(8,*), 'MEANX = ', MEANX, ', MEANY = ', MEANY
   ENDIF
100 CONTINUE

MEANX = MEANX / DFLOAT(END - (START - 1))
MEANY = MEANY / DFLOAT(END - (START - 1))
WRITE(8,*), 'MEANX = ', MEANX, ', MEANY = ', MEANY

CALCULATE THE SAMPLE VARIANCE AND COVARIANCE

DO 150 I = START, END
   DIFFX(I) = X(I) - MEANX
   DIFFY(I) = Y(I) - MEANY
   SX2 = SX2 + DIFFX(I) ** 2
   SXY = SXY + DIFFX(I) * DIFFY(I)
   IF (IOUT .EQ. 10) THEN
      WRITE(8,*), 'DIFFX = ', DIFFX(I), ', DIFFY = ', DIFFY(I)
      WRITE(8,*), 'SX2 = ', SX2, ', SXY = ', SXY
   ENDIF
150 CONTINUE
sx2 = sk2 / dfloat(end - start)

sxy = sxy / dfloat(end - start)

write(8, *) 'sx2 = ', sx2, ' sxy = ', sxy

calculate regression parameters

b = sdy / sx2
lnc = meany - b * meanx

write(8, *) 'b = ', b, ' lnc = ', lnc

return
end

7.4.2 ABTFIT Program

7.4.2.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for ABTFIT is given in Figure 7-11. The program, subprogram, and file names are indicated by UPPERCASE letters. Subprogram DUNLSJ is described in "User's Manual," IMSL Math/ Library FORTRAN Subroutines for Mathematical Applications MALB-USM-UNBND-EN8901-1.1, Version 1.1, Volume 3, IMSL Inc., January 1989, pp. 841-846.

Figure 7-11 Tree Structure For Program ABTFIT

7.4.2.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7-12. The section number where each subprogram is described is given next to the name.

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABT</td>
<td>4.2.3.1</td>
<td>Evaluates the function which defines the least squares problem.</td>
</tr>
<tr>
<td>ABTFIT</td>
<td>4.2.3</td>
<td>The main routine that controls the logical flow of the prior failure</td>
</tr>
<tr>
<td></td>
<td></td>
<td>distribution parameter estimation calculations.</td>
</tr>
<tr>
<td>DUNLSJ</td>
<td>N/A</td>
<td>An IMSL routine that solves a nonlinear least squares problem by using a</td>
</tr>
<tr>
<td></td>
<td></td>
<td>modified Levenberg-Marquardt algorithm.</td>
</tr>
<tr>
<td>JABT</td>
<td>4.2.3.2</td>
<td>Evaluates the Jacobian at a point X.</td>
</tr>
</tbody>
</table>

Table 7-12 List of Subprograms for Program ABTFIT

7 - 455
7.4.2.3 Description of Variables

A list of variables used in the prior failure distribution parameter estimation code, ABTFIT, is given in Table 7-13. The variable names are indicated by BOLD UPPER-CASE letters; the variable “type” can be interpreted as follows: INT is a standard integer variable, and DRE is a double precision variable. The various array dimensions are defined by using the following parameters: MAXJAC, MAXM, N, and N2.

Table 7-13 List of Variables for Program ABTFIT

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABT</td>
<td>DRE</td>
<td>Evaluates the function given by Equation 2-10 which defines the least squares problem.</td>
</tr>
<tr>
<td>ARGA, ARGB</td>
<td>DRE</td>
<td>Intermediate calculation variables.</td>
</tr>
<tr>
<td>B</td>
<td>DRE</td>
<td>Slope parameter, b.</td>
</tr>
<tr>
<td>END</td>
<td>INT</td>
<td>Ending index position for regression.</td>
</tr>
<tr>
<td>F(M)</td>
<td>DRE</td>
<td>Vector of length M containing the function values at X.</td>
</tr>
<tr>
<td>FJAC(MAXJAC, N)</td>
<td>DRE</td>
<td>The M by N matrix containing the calculated Jacobian at the approximate solution.</td>
</tr>
<tr>
<td>FOFN</td>
<td>DRE</td>
<td>( F(N_i) ) in Equation 2-10.</td>
</tr>
<tr>
<td>FSCALE(MAXM)</td>
<td>DRE</td>
<td>Vector of length M containing the diagonal scaling matrix for the functions.</td>
</tr>
<tr>
<td>FVEC(MAXM)</td>
<td>DRE</td>
<td>Vector of length M containing the residuals at the approximate solution.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls DO loop.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>IPARAM(6)</td>
<td>INT</td>
<td>Parameter vector of length 6, required by DUNLSJ.</td>
</tr>
<tr>
<td>J</td>
<td>INT</td>
<td>Array index variable.</td>
</tr>
<tr>
<td>JABT</td>
<td>DRE</td>
<td>Evaluates the Jacobian at a point X.</td>
</tr>
<tr>
<td>LDFJAC</td>
<td>INT</td>
<td>Leading dimension of FJAC.</td>
</tr>
<tr>
<td>LIFE(MAXM)</td>
<td>DRE</td>
<td>1-D array containing values of ( N_i ) Equation 2-10, generated by the PFM and scaled by LSCALE.</td>
</tr>
<tr>
<td>LIFEM(MAXM)</td>
<td>DRE</td>
<td>1-D array containing all values of ( N_i ) Equation 2-10, generated by the PFM.</td>
</tr>
<tr>
<td>LSCALE</td>
<td>DRE</td>
<td>Used to scale ( N_i ) values to control instability in ( \theta ) estimation.</td>
</tr>
</tbody>
</table>
### Table 7-13 List of Variables for Program ABTFIT (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>INT</td>
<td>Length of F, the number of ((F(N_i), N_i)) pairs to be used.</td>
</tr>
<tr>
<td>MAXJAC</td>
<td>INT</td>
<td>Maximum value for LDFJAC.</td>
</tr>
<tr>
<td>MAXM</td>
<td>INT</td>
<td>Maximum value for M.</td>
</tr>
<tr>
<td>MTOT</td>
<td>INT</td>
<td>The total number of ((F(N_i), N_i)) pairs provided by the PFM.</td>
</tr>
<tr>
<td>N</td>
<td>INT</td>
<td>Length of X, two for this application.</td>
</tr>
<tr>
<td>N2</td>
<td>INT</td>
<td>Dummy parameter replacing N in routines ABT and JABT.</td>
</tr>
<tr>
<td>RPARAM(7)</td>
<td>DRE</td>
<td>Parameter vector of length 7, required by DUNLSJ.</td>
</tr>
<tr>
<td>START</td>
<td>INT</td>
<td>Starting index position for regression.</td>
</tr>
<tr>
<td>X(N)</td>
<td>DRE</td>
<td>Vector of length N containing the approximate solution. (X(1)) is (\theta), and (X(2)) is (\alpha).</td>
</tr>
<tr>
<td>XGUESS(N)</td>
<td>DRE</td>
<td>Vector of length N containing the initial guess.</td>
</tr>
<tr>
<td>XSCALE(N)</td>
<td>DRE</td>
<td>Vector of length N containing the diagonal scaling matrix for the variables.</td>
</tr>
<tr>
<td>Y(MAXM)</td>
<td>DRE</td>
<td>1-D array containing values of (Y_i), Equation 2-10, corresponding to the (N_i)'s in the array LIFE().</td>
</tr>
</tbody>
</table>

### 7.4.2.4 Program ABTFIT Listing

```fortran
C PROGRAM ABTFIT CONTROLS THE NONLINEAR LEAST SQUARES CALCULATIONS
C REQUIRED TO ESTIMATE THE PRIOR DISTRIBUTION PARAMETERS THETA & ALPHA
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 24AUG90 COMMENTS: 20SEP91
C VERSION: 4.2
C
C Copyright (C) 1990, California Institute of Technology.
C U.S. Government Sponsorship under NASA Contract NAS7-918
C is acknowledged.

PROGRAM ABTFIT

C DECLARATION OF VARIABLES

COMMON Y, LIFE, IOUT, XGUESS, B
INTEGER DUM, END, I, IPARAM(6), IOUT, J, LDFJAC, M, MAXJAC,
&    MAXM, MTOT, N, START
PARAMETER (MAXJAC = 200, MAXM = 200; N = 2)

DOUBLE PRECISION ABT, B, FJAC(MAXJAC, N), FOFN(MAXM),
&    FSSCALE(MAXM), FVEC(MAXM), JABT, LIFE(MAXM),
&    LIFEM(MAXM), LSSCALE, RPARAM(7), X(N), XGUESS(N),
&    XSCALE(N), Y(MAXM)
```

7 - 457
EXTERNAL ABT, JABT, DUNLSJ

LIST OF VARIABLES

ABT Evaluates the function which defines the nonlinear least squares problem.
B Shape parameter.
DUM Dummy variable used during data entry.
END Ending position for regression loops.
FJAC() The M by N matrix containing the calculated Jacobian at the approximate solution.
FFON() 1-D array containing values of F(Ni).
FSCALE() Vector of length N containing the diagonal scaling matrix for the functions.
FVEC() Vector of length M containing the residuals at the approximate solution.
I Controls DO loop.
IOUT Output dump controller.
IPARAM() Parameter vector of length 6, required by DUNLSJ.
J Array index variable.
JABT Evaluates the Jacobian at a point X.
LDFJAC Leading dimension of FJAC.
LIFE() 1-D array containing values of the life, Ni, generated by the PFM to be used by the nonlinear least squares algorithm.
LIFEM() 1-D array containing all MTOT values of the life, Ni, generated by the PFM.
LSCALE LIFE() scaling parameter to enable convergence.
M Length of F, the number of (F(Ni), Ni) pairs provided by the PFM to be used by the nonlinear least squares algorithm.
MAXJAC Maximum value for LDFJAC.
MAXN Maximum value for M and MTOT.
MTOT Length of FOFN() & LIFEM(), the number of (F(Ni), Ni) pairs provided by the PFM.
N Length of X, two for this application.
RPARAM() Parameter vector of length 7, required by DUNLSJ.
START Starting position for regression loops.
X() Vector of length N containing the approximate solution. X(1) is theta, X(2) is alpha.
XGUESS() Vector of length N containing the initial guess.
XSCALE() Vector of length N containing the diagonal scaling matrix for the variables.
Y() 1-D array containing values of Y corresponding to the Ni's in the array LIFE().

C INITIALIZE VARIABLES AND READ Input PARAMETERS

DATA FSSCALE / MAXN*1.0D0 / 
IPARAM(1) = 0
OPEN (11, FILE = 'PARAMS', STATUS = 'OLD') 
READ(11,*) IOU, START, END, MTOT
READ(11,*) XGUESS(1), XGUESS(2), B
READ(11,*) XSCALE(1), XSCALE(2), LSCALE
CLOSE (11)
M = END - START + 1 
LDFJAC = M
IF (IOUT .EQ. 10) OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW') 
IF (IOUT .EQ. 10) THEN
OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
WRITE(8,*) 'THETA ALPH' 
ENDIF
OPEN (12, FILE = 'LOWLIF', STATUS = 'OLD')
DO 100 I = 1, MTOT
READ(12,*) DUM, FOFN(I), LIFEM(I)
CONTINUE
CLOSE (12)
DO 200 I = 1, M
J = I + START - 1

100 CONTINUE

C C
LIFE(I) = LIFEM(J) * LSCALE  
Y(I) = - DLOG (1.50 - FOFN(J))

200 CONTINUE
C PERFORM NONLINEAR LEAST SQUARES TO FIND THETA AND ALPHA
CALL DUNLSJ (ABT, JABT, M, N, XGUESS, XSCALE, FSSCALE,  
& IPARAM, RPARAM, X, FVEC, FJAC, LDFJAC)
C PRINT RESULTS
OPEN (13, FILE = 'ABTOUT', STATUS = 'NEW')
WRITE(13,900) X(2), B, (X(1)*LSCALE**(-B)),  
& IPARAM(3), IPARAM(4), IPARAM(5)
CLOSE (13)
OPEN (14, FILE = 'BAYESD', STATUS = 'NEW')
WRITE(14,*) B, (X(1)*LSCALE**(-B)), X(2)
CLOSE (14)
900 FORMAT (2X,'Copyright (c) 1990, California Institute of ',  
& 'Technology. U.S. Government',//,2X,'Sponsorship under ',  
& 'NASA Contract NAS7-918 is acknowledged.',  
& '//.2X,'The solution is',  
& '5X,'Alpha: ', D11.6,  
& '5X,'Beta: ', D11.6,  
& '5X,'Theta: ', D11.6,  
& '//.2X,'The number of iterations is ',10X,I3,  
& '//.2X,'The number of function evaluations is ',I3,  
& '//.2X,'The number of Jacobian evaluations is ',I3,/)
STOP END

C*****************************************************************************
C SUBROUTINE ABT EVALUATES THE FUNCTION AT THE POINT X
C PROGRAMMER: L.NEWLIN
C DATE: CODE: 23AUG90 COMMENTS: 20SEP91
C VERSION: ABTFIT V4.1, V4.2
SUBROUTINE ABT (M, N, X, F)
COMMON Y, LIFE, IOUT, XGUESS, B
INTEGER I, IOUT, M, MAXM, N, N2
PARAMETER (MAXM = 200, N2 = 2)
DOUBLE PRECISION B, F(M), LIFE(MAXM), X(N), XGUESS(N2), Y(MAXM)

LIST OF VARIABLES

B  Shape parameter.
F()  Vector of length M containing the function values at X.
I  Controls DO loop.
IOUT  Output dump controller.
LIFE()  1-D array containing values of the life, Ni, generated by  
the PFM.
M  Length of F, the number of (F(Ni), Ni) pairs provided by  
the PFM.
MAXM  Maximum value for M.
N  Length of X, two for this application.
N2  Dummy parameter replacing N in routines ABT and JABT.
X()  Vector of length N containing the approximate solution.
C  X(1) is theta, X(2) is alpha.
XGUESS()  Vector of length N containing the initial guess.
C  Y()  1-D array containing values of Y corresponding to the Ni's  
in the array LIFE().

IF (IOUT .EQ. 20) WRITE(8,175) X(1), X(2)
DO 100 I = 1, M
  F(I) = Y(I) - X(2) * DLOG (1.0 + (LIFE(I) ** B) / X(1))
  IF (IOUT .EQ. 10)
    WRITE(8,150) I, F(I), X(1), X(2), LIFE(I), Y(I)
  CONTINUE
100 FORMAT (I5,1X,D10.3,1X,D10.3,1X,D10.3,1X,D10.3,1X,D10.3)
150 FORMAT (2X,'F',DI3.6,1X,DI3.6)
RETURN
END

C*************************************************************************
C SUBROUTINE JABT EVALUATES THE JACOBIAN OF THE FUNCTION AT THE POINT X
C PROGRAMMER: L. NEWLIN
C DATE: CODE: 23AUG90 COMMENTS: 20SEP91
C VERSION: ABTFIT V4.1, V4.2
C*************************************************************************

SUBROUTINE JABT (M, N, X, FJAC, LDFJAC)
COMMON Y, LIFE, IOUT, XGUESS, B
INTEGER I, IOUT, LDFJAC, M, MAXM, N, N2
PARAMETER (MAXM = 200, N2 = 2)
DOUBLE PRECISION ARGA, ARGB, B, FJAC(LDFJAC, N),
& LIFE(MAXM), X(N), XGUESS(N2), Y(MAXM)

LIST OF VARIABLES
Intermediate calculation variables.
Intermediate calculation variables.
Shape parameter.
Vector of length M containing the function values at X.
The M by N matrix containing the calculated Jacobian at the
approximate solution.
Controls DO loop.
Output dump controller.
Evaluates the Jacobian at a point X.
Leading dimension of FJAC exactly as specified in the dimension
statement of the calling program.
1-D array containing values of the life, Ni, generated by
the PFM.
Length of F, the number of (F(Ni), Ni) pairs provided by
the PFM.
Maximum value for M.
Length of X, two for this application.
Dummy parameter replacing N in routines ABT and JABT.
Vector of length N containing the approximate solution.
  X(1) is theta, X(2) is alpha.
Vector of length N containing the initial guess.
  Y() 1-D array containing values of Y corresponding to the Ni's
  in the array LIFE().

IF (IOUT .EQ. 20) WRITE(8,175) X(1), X(2)
DO 100 I = 1, M
  ARGA = (LIFE(I) ** B) / X(1)
  ARGB = 1.0 + ARGA
  FJAC(I,1) = X(2) * ARGA / (X(1) * ARGB)
  FJAC(I,2) = - DLOG (ARGB)
  IF (IOUT .EQ. 10) THEN
    WRITE(8,*), 'I = ', I, ' N = ', LIFE(I)
    WRITE(8,*), 'X1 = ', X(1), ' X2 = ', X(2)
    WRITE(8,*), 'ARGA = ', ARGA, ' ARGB = ', ARGB
    WRITE(8,*), 'FJAC1 = ', FJAC(I,1), ' FJAC2 = ', FJAC(I,2)
  END IF
100 FORMAT (I5,1X,D10.3,1X,D10.3,1X,D10.3,1X,D10.3,1X,D10.3,1X,D10.3)
7.4.3 LZERO Program

7.4.3.1 Program Tree Structure

The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for LZERO is given in Figure 7-12. The program, subprogram, and file names are indicated by UPPERCASE letters.

![Figure 7-12 Tree Structure For Program LZERO](image)

7.4.3.2 List of Subprograms

A list of subprograms and their purposes is given in Table 7-14. The section number where each subprogram is described is given next to the name.

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLGAM</td>
<td>4.2.4.2</td>
<td>Calculates the double precision value of ln[ Γ(x) ] for a given value of x.</td>
</tr>
<tr>
<td>FCT</td>
<td>4.2.4.4</td>
<td>Calculates the value of the function at the current iteration value of λ, in order to find the solution for the desired assurance level.</td>
</tr>
<tr>
<td>GAMMA</td>
<td>4.2.4.1</td>
<td>Calculates the cumulative distribution function for a Gamma variate.</td>
</tr>
<tr>
<td>LZERO</td>
<td>4.2.4</td>
<td>The main routine that controls the logical flow of the assurance calculation.</td>
</tr>
</tbody>
</table>
Table 7-14 List of Subprograms for Program LZERO (Cont'd)

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUELLR</td>
<td>4.2.4.3</td>
<td>Solves the general nonlinear equation of the ( f(x) = 0 ) form by means of Mueller's iteration method.</td>
</tr>
<tr>
<td>TRMNAT</td>
<td>4.2.4.5</td>
<td>Performs premature program termination when required.</td>
</tr>
</tbody>
</table>

7.4.3.3 Description of Variables

A list of variables used in the assurance calculation code, LZERO, is given in Table 7-15. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" can be interpreted as follows: INT is a standard integer variable, and DRE is a double precision variable.

Table 7-15 List of Variables for Program LZERO

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>DRE</td>
<td>Desired assurance level requested by the user.</td>
</tr>
<tr>
<td>ALPHA</td>
<td>DRE</td>
<td>Gamma distribution parameter ( \alpha ) in Equation 2-1.</td>
</tr>
<tr>
<td>BETA</td>
<td>DRE</td>
<td>Weibull distribution parameter ( \beta ) in Equation 2-1.</td>
</tr>
<tr>
<td>DLGAM</td>
<td>DRE</td>
<td>Function that calculates the natural logarithm of a Gamma function.</td>
</tr>
<tr>
<td>EPS</td>
<td>DRE</td>
<td>Value of the maximum error allowed in the result of the Mueller's iteration method calculations.</td>
</tr>
<tr>
<td>F</td>
<td>DRE</td>
<td>Cumulative distribution function in Equation 2-5. When ( \lambda ) equals ( \lambda_o ), F equals the desired assurance level.</td>
</tr>
<tr>
<td>FL</td>
<td>DRE</td>
<td>Cumulative distribution function in Equation 2-5. When ( \lambda ) equals ( \lambda_{lb} ), FL equals the assurance level.</td>
</tr>
<tr>
<td>FU</td>
<td>DRE</td>
<td>Cumulative distribution function in Equation 2-5. When ( \lambda ) equals ( \lambda_{ub} ), FU equals the assurance level.</td>
</tr>
<tr>
<td>IER</td>
<td>INT</td>
<td>Provides information on the type of error occurring in the Mueller's iteration method calculations. A value of 0 indicates that no error occurred; a value of 1 indicates no convergence after NS iteration steps; and a value of 2 indicates that the basic assumption, that the bounds provided do not bound the solution, is not satisfied.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>DRE</td>
<td>The random variate ( \lambda ) in Equation 2-1. When ( \lambda ) equals ( \lambda_o ), F equals the desired assurance level.</td>
</tr>
</tbody>
</table>
Table 7-15 List of Variables for Program LZERO (Cont'd)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAML</td>
<td>DRE</td>
<td>The lower bound of ( \lambda_0 ) in Equation 2-1. When ( \lambda ) equals ( \lambda_{lb} ), FL equals the assurance level.</td>
</tr>
<tr>
<td>LAMU</td>
<td>DRE</td>
<td>The upper bound of ( \lambda_0 ) in Equation 2-1. When ( \lambda ) equals ( \lambda_{ub} ), FU equals the assurance level.</td>
</tr>
<tr>
<td>NS</td>
<td>INT</td>
<td>Number of terms of the infinite series used to approximate the Gamma cumulative distribution function.</td>
</tr>
<tr>
<td>PSI</td>
<td>DRE</td>
<td>Computational precision for calculating the value of a Gamma cumulative distribution.</td>
</tr>
<tr>
<td>THETA</td>
<td>DRE</td>
<td>Gamma distribution parameter ( \theta ) in Equation 2-1.</td>
</tr>
</tbody>
</table>

7.4.3.4 Program LZERO Listing

```c
PROGRAM LZERO CALCULATES THE PARAMETER LAMBDA FOR A GIVEN ASSURANCE
PROGRAMMER: L. NEWLIN
DATE: 24SEP91
VERSION: V2

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

PROGRAM LZERO

SUBPROGRAMS: GAMMA, MUELLR, TRMNAT
FILES: 1: BAYESD-OLD, 2: LDAAT-OLD,
       3: LOUT-NEW, 8: IOUTPR-NEW

IMPLICIT NONE
COMMON IOUT
COMMON / INFO / A, ALPHA, THETA, NS, PSI
INTEGER IER, IOUT, NS
DOUBLE PRECISION A, ALPHA, BETA, EPS, F, FL, FU, LAMBDA, LAML, LAMU, PSI, THETA

LIST OF VARIABLES
A       REQUESTED ASSURANCE LEVEL
ALPHA   GAMMA DISTRIBUTION PARAMETER
BETA    WEIBULL DISTRIBUTION PARAMETER
EPS     COMPUTATIONAL PRECISION FOR SUBPROGRAM MUELLR
F       CUMULATIVE DISTRIBUTION FUNCTION AT LAMBDA
FL      CUMULATIVE DISTRIBUTION FUNCTION AT LAML
FU      CUMULATIVE DISTRIBUTION FUNCTION AT LAMU
IER     RESULTANT ERROR PARAMETER CODED AS FOLLOWS:
       IER = 0 - NO ERROR.
       IER = 1 - NO CONVERGENCE AFTER IEND ITERATION STEPS
                  FOLLOWED BY IEND SUCCESSIVE STEPS OF
                  BISECTION.
       IER = 2 - BASIC ASSUMPTION FCT(XLI) * FCT(XRI) LESS
                  THAN OR EQUAL TO ZERO IS NOT SATISFIED.
```

7 - 463
SUBROUTINE GAMMA CALCULATES THE CUMULATIVE DISTRIBUTION FUNCTION FOR A GAMMA VARIATE

SUBROUTINE GAMMA (LAML, ALPHA, THETA, NS, PSI, FL)

CALL GAMMA (LAMU, ALPHA, THETA, NS, PSI, FU)

WRITE(3,910) LAML, FL, LAMU, FU

IF ((A .LE. FL) .OR. (A .GE. FU)) CALL TRMNAT

CALL MUELLR (LAMBD, LAML, LAMU, EPS, NS, IER)

IF (IER .EQ. 0)

CALL GAMMA (LAML, ALPHA, THETA, NS, PSI, F)

ELSEIF (IER .EQ. 1) THEN

WRITE(2,*), 'NO CONVERGENCE AFTER SPECIFIED NUMBER OF ITERATIONS'

CALL TRMNAT

ELSEIF (IER .EQ. 2) THEN

WRITE(2,*), 'BASIC ASSUMPTION NOT SATISFIED'

CALL TRMNAT

ELSE

WRITE(2,*), 'ERROR INCORRECTLY SPECIFIED'

CALL TRMNAT

ENDIF

900 FORMAT (2X,'Copyright (C) 1990, California Institute of ', & & 'Technology. U.S. Government', & & '//,2X,'Sponsorship under', & & '//,NASA Contract NAS7-918 is acknowledged.', & & '///,2X,'The Gamma distribution parameters are', & & '/,,7X,'Alpha ',D12.7, & & '/,,7X,'Theta ' ,D12.7)


920 FORMAT ('//,2X,'At an assurance level of ',F9.7, & & '/,,4X,'The value of lambda is ',D12.7)

STOP

END

C**************************************************************************
C SUBROUTINE GAMMA CALCULATES THE CUMULATIVE DISTRIBUTION FUNCTION FOR
C A GAMMA VARIATE
C PROGRAMMER: D. EBBELER

7-464
SUBROUTINE GAMMA (X, ALPHA, THETA, NS, PSI, F)

LIST OF VARIABLES

WEIBULL DISTRIBUTION PARAMETER

FUNCTION THAT CALCULATES LOGARITHM OF GAMMA(XX)

INCREMENT OF F

COUNTER FOR SERIES SUMMATION

OUTPUT DUMP CONTROLLER

NUMBER OF TERMS OF THE INFINITE SERIES TO CALCULATE GAMMA

COMPUTATIONAL PRECISION

REAL VALUE OF I

WEIBULL DISTRIBUTION PARAMETER

IF (IOUT .EQ. 10) WRITE(8,*) 'ALPHA = ', ALPHA, ' THETA = ', THETA
I = -1
F = 0.0

SUM THE SERIES FROM I = 1, NS

1 I = I + 1
IF (I .GT. NS) GO TO 3

FB = DEXP (- DLGAM (ALPHA + 1.0 + RI) + (RI + ALPHA)
& * DLOG (X * THETA) - X * THETA)
IF (IOUT .EQ. 10) WRITE(8,*) 'I = ', I, ' FB = ', FB
IF (I .EQ. 0) GO TO 2
IF (DABS (FB / F) .LE. PSI) RETURN

2 F = F + FB
IF (IOUT .EQ. 10) WRITE(8,*) 'F = ', F
GO TO 1

3 F = 0.0
RETURN
END

THE FUNCTION CALCULATES THE DOUBLE PRECISION NATURAL LOGARITHM OF THE GAMMA FUNCTION OF A GIVEN DOUBLE PRECISION ARGUMENT.

NOTE: THIS IS A MODIFIED VERSION OF THE SUBROUTINE OF THE SAME NAME FOUND IN 'Mathematics -- Special Functions', IBM APPLICATION PROGRAM, SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE, VERSION III, PROGRAMMER'S MANUAL, Program Number 360-CM-03X, Page 362.

SUBPROGRAMS: TRMNAT

COMMON IOUT
DOUBLE PRECISION XX, ZZ, TERM, RZZ, DLNG
INTEGER IER, IOUT

XX
THE DOUBLE PRECISION ARGUMENT FOR THE LOG GAMMA FUNCTION
ZZ
VARIABLE ACTUALLY USED IN THE CALCULATION INSTEAD OF XX
TERM
USED TO CONVERT TO X <= 18
RZZ
1/ZZ + 1.0
DLNG
HOLDS VALUE OF DLGAM DURING CALCULATIONS
IER
ERROR CONTROLLER:
-1 -- XX IS WITHIN 10**(-9) OF BEING ZERO OR XX
      IS NEGATIVE. DLNG IS SET TO -1.0D35
0 -- NO ERROR.
+1 -- XX IS GREATER THAN 10**30. DLNG IS SET TO +1.0D35.

IER = 0
ZZ = XX
1 IF (XX-1.D10) 2,2,1
1 IF (XX-1.D30) 8,9,9
C SEE IF XX IS NEAR ZERO OR NEGATIVE
2 IF (XX-1.D-9) 3,3,4
3 IER = -1
   DLNG = -1.D35
   GO TO 10
C XX GREATER THAN ZERO AND LESS THAN OR EQUAL TO 1.D+10
4 TERM = 1.D0
5 IF (ZZ-18.D0) 6,6,7
6 TERM = TERM*ZZ
   ZZ = ZZ + 1.D0
C IF (IOUT .EQ. 1) WRITE(8,*)'TERM=',TERM,' ZZ=',ZZ
   GO TO 5
7 RZZ = (ZZ-0.5D0)*DLOG(ZZ)-ZZ +0.918938532046727 -DLOG(TERM)+
     1/(1.D0/ZZ)*(.833333333333333D-1 -((RZZ*(.2777777777777777D-2 +RZZ*
     2(.7936507936507936D-3 -(RZZ*(.5952380952380952D-3))))))
C IF (IOUT .EQ. 1) WRITE(8,*)'RZZ=',RZZ,' DLNG=',DLNG
   GO TO 10
C XX GREATER THAN 1.D+10 AND LESS THAN 1.D+30
8 DLNG = ZZ * (DLOG(ZZ) - 1.D0)
   GO TO 10
C XX GREATER THAN OR EQUAL TO 1.D+30
9 IER = +1
   DLNG = 1.D35
10 IF (IER .NE. 0) THEN
   WRITE(2,*)'ATTEMPTED TO TAKE DLGAM OF NUMBER ',
    'TOO LARGE OR TOO SMALL'
   CALL TRMNAT
ENDIF
C IF (IOUT .EQ. 1) WRITE(8,*)'AT 10, DLNG=', DLNG
DLGAM = DLNG
RETURN
END

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

SUBROUTINE TRMNAT
STOP
END
C FUNCTION FCT CALCULATES THE CUMULATIVE DISTRIBUTION FUNCTION FOR
A GAMMA VARIATE MINUS THE DESIRED ASSURANCE
PROGRAMMER: L. NEWLIN
DATE: 24SEP91
VERSION: LZERO V2

DOUBLE PRECISION FUNCTION FCT (X)
SUBPROGRAMS: GAMMA
COMMON IOUT
COMMON / INFO / A, ALPHA, THETA, NS, PSI
INTEGER IOUT, NS
DOUBLE PRECISION A, ALPHA, F, PSI, THETA, X

LIST OF VARIABLES
A REQUESTED ASSURANCE LEVEL
ALPHA GAMMA DISTRIBUTION PARAMETER
F CUMULATIVE DISTRIBUTION FUNCTION
IOUT OUTPUT DUMP CONTROLLER
X THRESHOLD LAMBDA AT WHICH F() IS CALCULATED
NS NUMBER OF TERMS OF THE INFINITE SERIES TO CALCULATE GAMMA
PSI COMPUTATIONAL PRECISION
THETA WEIBULL DISTRIBUTION PARAMETER

CALL GAMMA (X, ALPHA, THETA, NS, PSI, F)
IF (IOUT .EQ. I0) WRITE(8,*) 'X = ', X, ' F = ', F
FCT = F - A
END

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

SUBROUTINE MUELLR
THE FOLLOWING SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE
DRTMI TAKEN FROM "Mathematics -- Roots of Nonlinear Equations,"
IBM APPLICATION PROGRAM, SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE,
VERSION III, PROGRAMMER'S MANUAL, Program Number 360-CM-03X, Page 219.
PURPOSE
TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM FCT(X) = 0
BY MEANS OF MUELLER'S ITERATION METHOD.
USAGE
CALL MUELLR (X, XLI, XRI, EPS, IEND, IER)

DESCRIPTION OF PARAMETERS
X - DOUBLE PRECISION RESULTANT ROOT OF EQUATION FCT(X) = 0.
F - DOUBLE PRECISION RESULTANT FUNCTION VALUE AT ROOT X.
FCT - NAME OF THE EXTERNAL DOUBLE PRECISION FUNCTION
SUBPROGRAM USED.
XLI - DOUBLE PRECISION INPUT VALUE WHICH SPECIFIES THE
INITIAL LEFT BOUND OF THE ROOT X.
XRI - DOUBLE PRECISION INPUT VALUE WHICH SPECIFIES THE
UPPER BOUND OF THE ERROR OF RESULT X.
IEND - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.
IER - RESULTANT ERROR PARAMETER CODED AS FOLLOWS
      IER = 0 - NO ERROR.
      IER = 1 - NO CONVERGENCE AFTER IEND ITERATION STEPS
      FOLLOWED BY IEND SUCCESSIVE STEPS OF
      BISECTION.
      IER = 2 - BASIC ASSUMPTION FCT(XLI) * FCT(XRI) LESS
      THAN OR EQUAL TO ZERO IS NOT SATISFIED.

REMARKS
      THE PROCEDURE ASSUMES THAT FUNCTIONS VALUES AT INITIAL
      BOUNDS XLI AND XRI HAVE NOT THE SAME SIGN. IF THIS BASIC
      ASSUMPTION IS NOT SATISFIED BY INPUT VALUES XLI AND XRI, THE
      PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER = 2.

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
      THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
      MUST BE FURNISHED BY THE USER.

METHOD
      SOLUTION OF EQUATION FCT(X) = 0 IS DONE BY MEANS OF MUELLER'S
      ITERATION METHOD OF SUCCESSIVE BISECTIONS AND INVERSE
      PARABOLIC INTERPOLATION, WHICH STARTS AT THE INITIAL BOUNDS
      XLI AND XRI. CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF
      FCT(X) AT ROOT X IS NOT EQUAL TO ZERO. ONE ITERATION STEP
      REQUIRES TWO EVALUATIONS OF FCT(X). FOR TEST ON SATISFACTORY
      ACCURACY SEE FORMULAE (3,4) OF MATHEMATICAL DESCRIPTION.
      FOR REFERENCE, SEE G. K. KRISTIANSEN, ZERO OF ARBITRARY
      FUNCTION, BIT, VOL. 3 (1963), PP. 205-206.

SUBROUTINE MUELLR (X, XLI, XRI, EPS, IEND, IER)
COMMON IOUT
DOUBLE PRECISION A, DX, EPS, F, FCT, FL, FM, FR, &
      TOL, TOLF, X, XL, XLI, XM, XR, XRI
INTEGER I, IEND, IER, IOUT, K

C CHECK INPUTS
C IF (IOUT .EQ. 20) THEN
  WRITE(8,* 'X = ', XLI, ' XRI = ', XRI
ENDIF
C PREPARE ITERATION
C IER = 0
C XL = XLI
C XR = XRI
C X = XL
C F = FCT(XL)
C IF (F) 1, 16, 1
  FL = F
  X = XR
  TOL = X
  F = FCT(XR)
  IF (F) 2, 16, 2
  FR = F
  IF (DSIGN(1.D0, FL) + DSIGN(1.D0, FR)) 25, 3, 25
C BASIC ASSUMPTION FL * FR LESS THAN 0 IS SATISFIED
C GENERATE TOLERANCE FOR FUNCTION VALUES
C 3 I = 0
C TOLF = 100. * EPS
C START ITERATION LOOP
C 4 I = I + 1
C START BISECTION LOOP
DO 13 K = 1, IEND
X = .5D0 * (XL + XR)
TOL = X
F = FCT(TOL)
IF (F) 5, 16, 5
5 IF (DSIGN(1.D0, F) + DSIGN(1.D0, FR)) 7, 6, 7
C INTERCHANGE XL AND XR IN ORDER TO GET THE SAME SIGN IN F AND FR
6 TOL = XL
XL = XR
XR = TOL
TOL = FL
FL = FR
FR = TOL
7 TOL = F - FL
A = F + TOL
A = A + A
IF (A - FR * (FR - FL)) 8, 9, 9
8 IF (I - IEND) 17, 17, 9
9 XR = X
FR = F
C TEST ON SATISFACTORY ACCURACY IN BISECTION LOOP
TOL = EPS
A = DABS(XR)
IF (A - 1.D0) 11, 11, 10
10 TOL = TOL + A
11 IF (DABS(XR - XL) - TOL) 12, 12, 13
12 IF (DABS(FR - FL) - TOLF) 14, 14, 13
13 CONTINUE
C END OF BISECTION LOOP
C NO CONVERGENCE AFTER IEND ITERATION STEPS FOLLOWED BY IEND
C SUCCESSIVE STEPS OF BISECTION OF STEADILY INCREASING FUNCTION
C VALUES AT RIGHT BOUNDS. ERROR RETURN.
14 IF (DABS(FR) - DABS(FL)) 16, 16, 15
15 X = XL
F = FL
16 RETURN
C COMPUTATION OF ITERATED X-VALUE BY INVERSE PARABOLIC INTERPOLATION
17 A = FR - F
DX = (X - XL) * FL + (1.D0 + F * (A - TOL))
& * /
(A + (FR - FL)) / TOL
XM = X
FM = F
X = XL - DX
TOL = X
F = FCT(TOL)
IF (F) 18, 16, 18
C TEST ON SATISFACTORY ACCURACY IN ITERATION LOOP
18 TOL = EPS
A = DABS(X)
IF (A - 1.D0) 20, 20, 19
19 TOL = TOL * A
20 IF (DABS(DX) - TOL) 21, 21, 22
21 IF (DABS(F) - TOLF) 14, 14, 13
C PREPARATION OF NEXT BISECTION LOOP
22 IF (DSIGN(1.D0, F) + DSIGN(1.D0, FL)) 24, 23, 24
23 XR = X
FR = F
GO TO 4
24 XL = X
FL = X
XR = XM
FR = FM
GO TO 4

C END OF ITERATION LOOP

C ERROR RETURN IN CASE OF WRONG INPUT DATA

25 IER = 2
RETURN
END
Section 7.5
Bayesian Statistical Procedure Program

A description of the key variables and the FORTRAN source listing for the Bayesian statistical procedure code BAYES are given here. The pertinent statistical methodology is given in Section 2.1.1. The overall description of the program and the flowchart are given in Section 4.3. The user’s guide for running BAYES is given in Section 6.5.

7.5.1 BAYES Program

7.5.1.1 Description of Variables

A list of variables used in the Bayesian statistical procedure code, BAYES, is given in Table 7-16. The variable names are indicated by BOLD UPPERCASE letters; the variable “type” is specified as follows: INT is a standard integer variable and RE is a standard real variable. The array dimensions are defined by using the parameter MAXTYM. The program name is indicated by UPPERCASE letters.

Table 7-16 List of Variables for Program BAYES
(Footnote is at the end of the table)

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>RE</td>
<td>Gamma distribution parameter $\alpha$ for prior distribution of Weibull parameter $\lambda$, Equation 2-1.</td>
</tr>
<tr>
<td>ALPHUP</td>
<td>RE</td>
<td>Gamma distribution parameter $\alpha'$ for posterior distribution of Weibull parameter $\lambda$, Equation 2-2.</td>
</tr>
<tr>
<td>B1</td>
<td>RE</td>
<td>B1-life$^1$ calculated from Equation 2-6 by using prior distribution parameters $\alpha$, $\theta$, and $\beta$.</td>
</tr>
<tr>
<td>B1UP</td>
<td>RE</td>
<td>B1-life$^1$ calculated from Equation 2-6 by using posterior distribution parameters $\alpha'$, $\theta'$, and $\beta$.</td>
</tr>
<tr>
<td>BETA</td>
<td>RE</td>
<td>Weibull distribution shape parameter $\beta$, Equation 2-1.</td>
</tr>
<tr>
<td>BP01</td>
<td>RE</td>
<td>B.01-life$^1$ calculated from Equation 2-6 by using prior distribution parameters $\alpha$, $\theta$, and $\beta$.</td>
</tr>
<tr>
<td>BP01UP</td>
<td>RE</td>
<td>B.01-life$^1$ calculated from Equation 2-6 by using posterior distribution parameters $\alpha'$, $\theta'$, and $\beta$.</td>
</tr>
<tr>
<td>BP1</td>
<td>RE</td>
<td>B.1-life$^1$ calculated from Equation 2-6 by using prior distribution parameters $\alpha$, $\theta$, and $\beta$.</td>
</tr>
<tr>
<td>BP1UP</td>
<td>RE</td>
<td>B.1-life$^1$ calculated from Equation 2-6 by using posterior distribution parameters $\alpha'$, $\theta'$, and $\beta$.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ETA</td>
<td>RE</td>
<td>Estimate of $E(\eta)$ where $\eta = \lambda^{-1/\beta}$ with $\lambda \sim \Gamma(\alpha, \theta)$</td>
</tr>
<tr>
<td>ETUP</td>
<td>RE</td>
<td>Estimate of $E(\eta)$ where $\eta = \lambda^{-1/\beta}$ with $\lambda \sim \Gamma(\alpha', \theta')$.</td>
</tr>
<tr>
<td>FAIL</td>
<td>INT</td>
<td>Number of test failure times in operating experience data.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls DO loop.</td>
</tr>
<tr>
<td>INVALP</td>
<td>RE</td>
<td>Value of $1/\alpha$.</td>
</tr>
<tr>
<td>INVALU</td>
<td>RE</td>
<td>Value of $1/\alpha'$.</td>
</tr>
<tr>
<td>INVBET</td>
<td>RE</td>
<td>Value of $1/\beta$.</td>
</tr>
<tr>
<td>LAMBDA</td>
<td>RE</td>
<td>Expected value of $\lambda \sim \Gamma(\alpha, \theta)$ in Equation 2-1.</td>
</tr>
<tr>
<td>LAMBUP</td>
<td>RE</td>
<td>Expected value of $\lambda \sim \Gamma(\alpha', \theta')$ in Equation 2-2.</td>
</tr>
<tr>
<td>SUSP</td>
<td>INT</td>
<td>Number of test suspension times in operating experience data.</td>
</tr>
<tr>
<td>MAXTYM</td>
<td>INT</td>
<td>Maximum number of failure and/or suspension times allowed. The maximum number of times allowed is 50.</td>
</tr>
<tr>
<td>THETA</td>
<td>RE</td>
<td>Gamma distribution parameter $\theta$ for prior distribution of Weibull parameter $\lambda$, Equation 2-1.</td>
</tr>
<tr>
<td>THETUP</td>
<td>RE</td>
<td>Gamma distribution parameter $\theta'$ for posterior distribution of Weibull parameter $\lambda$, Equation 2-2.</td>
</tr>
<tr>
<td>TYME(MAXTYM)</td>
<td>RE</td>
<td>1-D array containing operating experience as failure and suspension times $t_i$, Equation 2-2.</td>
</tr>
</tbody>
</table>

1 A B-life is the value of the failure parameter (e.g., time) 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%. 

7 - 472
7.5.1.2 Program BAYES Listing

PROGRAM BAYES PERFORMS THE BAYESIAN UPDATING OF THE PRIOR DISTRIBUTION (FROM THE PFM) WITH THE OPERATING HISTORY (BOTH FAILURES AND SUSPENSIONS) 
PROGRAMMER: L. NEWLIN 
DATE: CODE 12MAY88 COMMENTS: 11SEP91 
VERSION: 3.1 

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

PROGRAM BAYES 

IMPLICIT NONE 
INTEGER MAXTYM 
PARAMETER (MAXTYM = 50) 
INTEGER FAIL, I, SUSP 
REAL ALPHA, ALPHUP, B1, B1UP, ETA, ETUP, INVALP, INVALU, INVET, LAMBDA, LAMBUP, 
& THETA, THETUP, TYME(MAXTYM) 

LIST OF VARIABLES 

ALPHA PRIOR GAMMA DISTRIBUTION PARAMETER ALPHA 
ALPHUP POSTERIOR OR UPDATED GAMMA DISTRIBUTION PARAMETER ALPHA 
B1 PRIOR DISTRIBUTION B1 LIFE CALCULATED FROM BETA, THETA AND ALPHA 
B1UP POSTERIOR OR UPDATED DISTRIBUTION B1 LIFE CALCULATED FROM BETA, THETUP AND ALPHUP 
BETA WEIBULL DISTRIBUTION SHAPE PARAMETER 
BP01 PRIOR DISTRIBUTION B.01 LIFE CALCULATED FROM BETA, THETA AND ALPHA 
BP01UP POSTERIOR OR UPDATED DISTRIBUTION B.01 LIFE CALCULATED FROM BETA, THETUP AND ALPHUP 
BP1 PRIOR DISTRIBUTION B.1 LIFE CALCULATED FROM BETA, THETA AND ALPHA 
BP1UP POSTERIOR OR UPDATED DISTRIBUTION B.1 LIFE CALCULATED FROM BETA, THETUP AND ALPHUP 
ETA PRIOR WEIBULL DISTRIBUTION LOCATION PARAMETER 
ETUP POSTERIOR OR UPDATED WEIBULL DISTRIBUTION LOCATION PARAMETER 
FAIL NUMBER OF FAILURES IN OPERATING EXPERIENCE 
I CONTROLS DO LOOP 
INVALP INVERSE OF ALPHA -- EQUAL TO 1/ALPHA 
INVALU INVERSE OF ALPHUP -- EQUAL TO 1/ALPHUP 
INVET INVERSE OF BETA -- EQUAL TO 1/BETA 
LAMBDA PRIOR WEIBULL DISTRIBUTION PARAMETER EQUAL TO (1/ETA)**BETA 
LAMBUP POSTERIOR OR UPDATED WEIBULL DISTRIBUTION PARAMETER EQUAL TO (1/ETUP)**BETA 
MAXTYM MAXIMUM NUMBER OF FAILURE AND/OR SUSPENSION TIMES ALLOWED 
SUSP NUMBER OF SUSPENSIONS IN OPERATING EXPERIENCE 
THETA PRIOR GAMMA DISTRIBUTION PARAMETER THETA 
THETUP POSTERIOR OR UPDATED GAMMA DISTRIBUTION PARAMETER THETA 
TYME() 1-D ARRAY CONTAINING FAILURE AND SUSPENSION OPERATING EXPERIENCE 

OPEN (1, FILE = 'BAYESD', STATUS = 'OLD') 
OPEN (2, FILE = 'BAYESO', STATUS = 'NEW') 
OPEN (3, FILE = 'UBAYES', STATUS = 'NEW') 

C INITIALIZE OPERATING EXPERIENCE ARRAY 

DO 10 I = 1, MAXTYM 
TYME(I) = 0.0 
10 CONTINUE 

7 - 473
C READ DATA FROM FILE BAYESD

READ(1,*) BETA, THETA, ALPHA
READ(1,*) FAIL, SUSP

DO 50 I = 1, (FAIL + SUSP)
   READ(I,*) TYME(I)
50 CONTINUE

C BEGIN CALCULATIONS -- UPDATE ALPHA AND THETA

ALPHUP = ALPHA + FLOAT (FAIL)
THETUP = THETA

DO 100 I = 1, (FAIL + SUSP)
   THETUP = THETUP + TYME(I) ** BETA
100 CONTINUE

C CALCULATE LAMBDA, ETA, AND BLIVES FOR BOTH PRIOR AND
C POSTERIOR DISTRIBUTIONS

LAMBDAP = ALPHA / THETA
LAMBDUP = ALPHUP / THETUP
INVBET = 1.0 / BETA
INVALP = 1.0 / ALPHA
INVALP = 1.0 / ALPHUP

ETA = (1.0 / LAMBDAP) ** INVBET
ETUP = (1.0 / LAMBDUP) ** INVBET

BP01 = (THETA * (0.9999 ** (- INVALP) - 1)) ** INVBET
BP1 = (THETA * (0.999 ** (- INVALP) - 1)) ** INVBET
B1 = (THETA * (0.99 ** (- INVALP) - 1)) ** INVBET

BP01UP = (THETUP * (0.9999 ** (- INVALP) - 1)) ** INVBET
BP1UP = (THETUP * (0.9999 ** (- INVALP) - 1)) ** INVBET
BIUP = (THETUP * (0.99 ** (- INVALP) - 1)) ** INVBET

C WRITE RESULTS TO FILE BAYESO AND THE POSTERIOR DISTRIBUTION
C TO FILE UBayes FOR ANY FURTHER UPDATING

WRITE(3,*) BETA, THETUP, ALPHUP
WRITE(2,900) BETA, BETA, ALPHUP, THETA, THETUP, LAMBDAP, LAMBDUP, ETA, ETUP
WRITE(2,910) BP01, BP01UP, BP1, BP1UP, B1, BIUP
WRITE(2,920)

IF (FAIL .GT. 0) THEN
   WRITE(2,930) FAIL
   DO 200 I = 1, FAIL
      WRITE(2,940) TYME(I)
   200 CONTINUE
ENDIF

IF (SUSP .GT. 0) THEN
   WRITE(2,950) SUSP
   DO 250 I = (FAIL + 1), (FAIL + SUSP)
      WRITE(2,940) TYME(I)
   250 CONTINUE
ENDIF

900 FORMAT (2X,'Copyright (C) 1990, California Institute of ','Technology. U.S. Government','
& //2X,'Sponsorship under ','
& "NASA Contract NAS7-918 is acknowledged."
& ///29X,'BAYESIAN UPDATING SUMMARY',///
& 18X,'PRIOR DISTRIBUTION',10X,'POSTERIOR DISTRIBUTION',
& ///2X,'PARAMETERS:'//8X,'BETA',10X,F9.6,22X,F9.6,
& ///7X,'ALPHA',9X,E12.6,19X,E12.6,
& ///7X,'THETA',9X,E12.6,19X,E12.6
& ///6X,'LAMBDAP',9X,E12.6,19X,E12.6,
& ///9X,'ETA',9X,E12.6,19X,E12.6)
910 FORMAT (//,6X,'BLIVES',//,8X,'B.01',9X,E12.6,19X,E12.6, 
& //,9X,'B.1',9X,E12.6,19X,E12.6, 
& //,10X,'B1',9X,E12.6,19X,E12.6)

920 FORMAT (///,26X,'OPERATING EXPERIENCE')

930 FORMAT (///,5X,'NUMBER OF FAILURES:',5X,I2, 
& ///,10X,'FAILURE TIMES:'/}

940 FORMAT (29X,E12.6)

950 FORMAT (///,2X,'NUMBER OF SUSPENSIONS:',5X,I2, 
& ///,7X,'SUSPENSION TIMES:'/}

STOP
END
Section 7.6
Random Number Generation Subprograms

Descriptions of the key variables and the FORTRAN source listings for the random number generator codes are given here. The pertinent statistical methodology is given in Section 2.1.3.1. The overall descriptions of the subprograms are given in Section 4.4.

7.6.1 RANDOM Subprogram

7.6.1.1 Description of Variables
A list of variables used in the Uniform(0,1) random number generator code, RANDOM, is given in Table 7-17. The variable names are indicated by BOLD UPPERCASE letters; the variable “type” is specified as follows: INT is a standard integer variable, RE is a standard real variable, and DRE is a double precision real variable. The subprogram names are indicated by UPPERCASE letters.

Table 7-17 List of Variables for Subprogram RANDOM

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAC</td>
<td>RE</td>
<td>The Uniform(0,1) random variate generated by RANDOM.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>RANA</td>
<td>DRE</td>
<td>Constant used by the linear congruential algorithm.</td>
</tr>
<tr>
<td>RANC</td>
<td>DRE</td>
<td>Constant used by the linear congruential algorithm.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>RANDIV</td>
<td>DRE</td>
<td>Intermediate calculation variable.</td>
</tr>
<tr>
<td>RANM</td>
<td>DRE</td>
<td>Constant used by the linear congruential algorithm.</td>
</tr>
<tr>
<td>RANSUB</td>
<td>DRE</td>
<td>Intermediate calculation variable.</td>
</tr>
<tr>
<td>RANT</td>
<td>DRE</td>
<td>Intermediate calculation variable.</td>
</tr>
<tr>
<td>RANX</td>
<td>DRE</td>
<td>Intermediate calculation variable.</td>
</tr>
</tbody>
</table>
7.6.1.2 Subprogram RANDOM Listing

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

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

PROGRAMMER: L. GRONDALSKI, 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

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

C SUBROUTINE RANDOM (FRAC, RAND)
C 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
7.6.2 NORMGN Subprogram

7.6.2.1 Description of Variables

A list of variables used in the Normal(μ, σ²) random number generator code, NORMGN, is given in Table 7-18. The variable names are indicated by **BOLD UPPERCASE** letters; the variable “type” is specified as follows: INT is a standard integer variable, RE is a standard real variable, and DRE is a double precision real variable. The subprogram names are indicated by UPPERCASE LETTERS.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAC</td>
<td>RE</td>
<td>A Uniform(0,1) random variate generated by RANDOM.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>MU</td>
<td>RE</td>
<td>Mean μ of the Normal distribution.</td>
</tr>
<tr>
<td>PI</td>
<td>RE</td>
<td>π, constant equal to 3.1415926536.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>SIGMA</td>
<td>RE</td>
<td>Standard deviation σ of the Normal distribution.</td>
</tr>
<tr>
<td>X</td>
<td>RE</td>
<td>The Normal(μ, σ²) random variate generated by NORMGN.</td>
</tr>
<tr>
<td>U1</td>
<td>RE</td>
<td>U(0,1) random variate.</td>
</tr>
<tr>
<td>U2</td>
<td>RE</td>
<td>U(0,1) random variate.</td>
</tr>
<tr>
<td>Z1</td>
<td>RE</td>
<td>N(0,1) random variate.</td>
</tr>
<tr>
<td>Z2</td>
<td>RE</td>
<td>N(0,1) random variate.</td>
</tr>
</tbody>
</table>

7.6.2.2 Subprogram NORMGN Listing

```
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)
```
SUBPROGRAM: RANDOM
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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAC</td>
<td>UNIFORM(0,1) RANDOM VARIATE</td>
</tr>
<tr>
<td>IOUT</td>
<td>OUTPUT DUMP CONTROLLER</td>
</tr>
<tr>
<td>MU</td>
<td>MEAN OF NORMAL DISTRIBUTION</td>
</tr>
<tr>
<td>RAND</td>
<td>RANDOM NUMBER SEED</td>
</tr>
<tr>
<td>SIGMA</td>
<td>STANDARD DEVIATION OF NORMAL DISTRIBUTION</td>
</tr>
<tr>
<td>X</td>
<td>NORMAL RANDOM VARIATE</td>
</tr>
<tr>
<td>U1</td>
<td>UNIFORM RANDOM NUMBER U(0,1)</td>
</tr>
<tr>
<td>U2</td>
<td>UNIFORM RANDOM NUMBER U(0,1)</td>
</tr>
<tr>
<td>Z1</td>
<td>NORMAL RANDOM NUMBER ON N(0,1)</td>
</tr>
<tr>
<td>Z2</td>
<td>NORMAL RANDOM NUMBER ON N(0,1)</td>
</tr>
</tbody>
</table>

IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*),'RAND =', RAND, ' MU =', MU, ' SIGMA =', SIGMA
CALL RANDOM (FRAC, RAND)
U1 = FRAC
CALL RANDOM (FRAC, RAND)
U2 = FRAC
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*),'U1 =', U1, ' U2 =', U2
Z1 = SQRT (- 2. * ALOG(U1)) * COS(2. * PI * U2)
Z2 = SQRT (- 2. * ALOG(U1)) * SIN(2. * PI * U2)
X = SIGMA * Z1 + MU
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*),'Z1 =', Z1, ' Z2 =', Z2, ' X =', X
RETURN
END

7.6.3 BETAGN Subprogram

7.6.3.1 Description of Variables
A list of variables used in the Beta(a, b, p, θ) random number generator code, BETAGN, is given in Table 7-19. The variable names are indicated by BOLD UPPERCASE letters; the variable “type” is specified as follows: INT is a standard integer variable, RE is a standard real variable, and DRE is a double precision real variable. The subprogram names are indicated by UPPERCASE letters.
Table 7-19 List of Variables for Subprogram BETAGN

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>RE</td>
<td>Lower bound a of the Beta distribution.</td>
</tr>
<tr>
<td>B</td>
<td>RE</td>
<td>Upper bound b of the Beta distribution.</td>
</tr>
<tr>
<td>GAM</td>
<td>RE</td>
<td>The Gamma(α, 1) random variate generator GAM.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>RHO</td>
<td>RE</td>
<td>Beta distribution parameter ρ.</td>
</tr>
<tr>
<td>THETA</td>
<td>RE</td>
<td>Beta distribution parameter θ.</td>
</tr>
<tr>
<td>W</td>
<td>RE</td>
<td>A Beta(0, 1, ρ, θ) random variate.</td>
</tr>
<tr>
<td>X</td>
<td>RE</td>
<td>The Beta(a, b, ρ, θ) random variate generated by BETAGN.</td>
</tr>
<tr>
<td>Y1</td>
<td>RE</td>
<td>Gamma(α, 1) random variate.</td>
</tr>
<tr>
<td>Y2</td>
<td>RE</td>
<td>Gamma(α, 1) random variate.</td>
</tr>
</tbody>
</table>

7.6.3.2 Subprogram BETAGN Listing

```
C******************************************************************************
C THIS SUBROUTINE GENERATES A BETA RANDOM VARIABLE
C PROGRAMMER: L. GRONDASKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: GAM
C
C The random variates are generated using the method described in:
C Johnson, N. L. and Kotz, S., Distribution in Statistics: Continuous
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,*) '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 - 481
7.6.4 GAM Subprogram

7.6.4.1 Description of Variables

A list of variables used in the Gamma(\(\alpha, 1\)) random number generator code, GAM, is given in Table 7-20. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" is specified as follows: INT is a standard integer variable, RE is a standard real variable, and DRE is a double precision real variable. The subprogram names are indicated by **UPPERCASE** letters.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>RE</td>
<td>Intermediate calculation variable.</td>
</tr>
<tr>
<td>ALPHA</td>
<td>RE</td>
<td>Gamma distribution parameter (\alpha).</td>
</tr>
<tr>
<td>ARG</td>
<td>RE</td>
<td>Intermediate calculation variable.</td>
</tr>
<tr>
<td>GAM</td>
<td>RE</td>
<td>The Gamma((\alpha, 1)) random variate generated by GAM.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>U1</td>
<td>RE</td>
<td>(U(0,1)) random variate.</td>
</tr>
<tr>
<td>U2</td>
<td>RE</td>
<td>(U(0,1)) random variate.</td>
</tr>
<tr>
<td>V1</td>
<td>RE</td>
<td>Exponential(1) random variate.</td>
</tr>
<tr>
<td>V2</td>
<td>RE</td>
<td>Exponential(1) random variate.</td>
</tr>
</tbody>
</table>

7.6.4.2 Subprogram GAM Listing

```
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
```

7 - 482
CALL RANDOM (U1, RAND)
CALL RANDOM (U2, RAND)
V1 = - ALOG(U1)
V2 = - ALOG(U2)
C
IF (IOUT .EQ. 15) WRITE(8,*), 'U1 =', U1, ' U2 =', U2, ' V1 =', V1,
   ' V2 =', V2
   ARG = A * (V1 - ALOG(V1) - 1.)
   IF (V2 .LT. ARG) GOTO 10
   GAM = ALPHA * V1
   IF (IOUT .EQ. 15) WRITE(8,*), 'GAMMA =', GAM
RETURN
END

7.6.5 WEIBGN Subprogram

7.6.5.1 Description of Variables
A list of variables used in the Weibull($\beta, \eta(\beta)$) random number generator code, WEIBGN, is given in Table 7-21. The variable names are indicated by BOLD UPPERCASE letters; the variable “type” is specified as follows: INT is a standard integer variable, RE is a standard real variable, and DRE is a double precision real variable. The subprogram names are indicated by UPPERCASE letters.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG</td>
<td>RE</td>
<td>Intermediate calculation variable.</td>
</tr>
<tr>
<td>BETA</td>
<td>RE</td>
<td>Weibull distribution shape parameter $\beta$.</td>
</tr>
<tr>
<td>ETA</td>
<td>RE</td>
<td>Weibull distribution location parameter $\eta$.</td>
</tr>
<tr>
<td>FRAC</td>
<td>RE</td>
<td>A Uniform(0,1) random variate generated by RANDOM.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>WEIB</td>
<td>RE</td>
<td>The Weibull($\beta, \eta(\beta)$) random variate generated by WEIBGN.</td>
</tr>
</tbody>
</table>

7.6.5.2 Subprogram WEIBGN Listing

C THIS SUBROUTINE GENERATES WEIBULL($\beta, \eta(\beta)$) 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
SUBROUTINE WEIBGN (BETA, RAND, WEIB)

INPUTS: BETA, RAND
OUTPUTS: WEIB
SUBPROGRAMS: RANDOM

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,*)'BETA = ', BETA, ', ETA = ', ETA,
& ' FRAC = ', FRAC, ', ARG = ', ARG, ', WEIB = ', WEIB
RETURN
END

7.6.6 PRYRV Subprogram

7.6.6.1 Description of Variables
A list of variables used in the random number generator code PRYRV, which generates a pair of independent Uniform variates \( U(a, b) \) and \( U(c, d) \), is given in Table 7-22. The variable names are indicated by **BOLD UPPERCASE** letters; the variable "type" is specified as follows: INT is a standard integer variable, RE is a standard real variable, and DRE is a double precision real variable. The subprogram names are indicated by **UPPERCASE** letters.
Table 7-22 List of Variables for Subprogram PRYRV

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAC</td>
<td>RE</td>
<td>A Uniform(0,1) random variate generated by RANDOM.</td>
</tr>
<tr>
<td>IOUT</td>
<td>INT</td>
<td>Output dump controller.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>RH01</td>
<td>RE</td>
<td>Lower bound a of the U(a,b) distribution.</td>
</tr>
<tr>
<td>RH02</td>
<td>RE</td>
<td>Upper bound b of the U(a,b) distribution.</td>
</tr>
<tr>
<td>THE1</td>
<td>RE</td>
<td>Lower bound c of the U(c,d) distribution.</td>
</tr>
<tr>
<td>THE2</td>
<td>RE</td>
<td>Upper bound d of the U(c,d) distribution.</td>
</tr>
<tr>
<td>X</td>
<td>RE</td>
<td>The U(a,b) random variate generated by PRYRV.</td>
</tr>
<tr>
<td>Y</td>
<td>RE</td>
<td>The U(c,d) random variate generated by PRYRV.</td>
</tr>
</tbody>
</table>

7.6.6.2 Subprogram PRYRV Listing

```
C******************************************************************************
C SUBROUTINE PRYRV GENERATES A PAIR OF U(RH01,RH02) AND U(THE1,THE2)
C INDEPENDENT RANDOM VARIATES
C PROGRAMMER: L. GRONDALSKI, L. NEWLIN
C DATE: 9MAR87
C SUBPROGRAM: RANDOM
C******************************************************************************

SUBROUTINE PRYRV (RAND, RH01, RH02, THE1, THE2, X, Y)
COMMON IOUT
DOUBLE PRECISION RAND
REAL FRAC, RH01, RH02, THE1, THE2, X, Y
INTEGER IOUT

CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE(8,*),'FRAC = ', FRAC
X = FRAC * (RH02 - RH01) + RH01

CALL RANDOM (FRAC, RAND)
IF (IOUT .EQ. 15) WRITE(8,*),'FRAC = ', FRAC
Y = FRAC * (THE2 - THE1) + THE1
IF (IOUT .EQ. 15) WRITE(8,*),'RH01 = ', RH01, ' RH02 = ', RH02,
RETURN
END
```
Section 7.7
Reference Time History Generation Program

The program tree structure, list of subprograms, description of the key variables, and the FORTRAN source listing for the reference time history generation code NBSIN are given here. The pertinent methodology is given in Section 2.1.4. The overall description of the program and the flowchart are given in Section 4.5. The user's guide for running NBSIN is given in Section 6.6.

7.7.1 NBSIN Program

7.7.1.1 Program Tree Structure
The tree structure gives the layout of the program in terms of the subprogram hierarchy. The tree structure for NBSIN is given in Figure 7-13. Subprograms NORMGN and RANDOM are described in Section 4.4. The program, subprogram, and file names are indicated by UPPERCASE letters.

![Figure 7-13 Tree Structure For Program NBSIN](image)

7.7.1.2 List of Subprograms
A list of subprograms and their purposes is given in Table 7-23. The section number where each subprogram is described is given next to the name.

<table>
<thead>
<tr>
<th>NAME</th>
<th>SECTION</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMGN1</td>
<td>4.4.3</td>
<td>Generates two independent Normal(μ, σ²) random variates.</td>
</tr>
<tr>
<td>RANDOM2</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>NBSIN</td>
<td>4.5.2</td>
<td>The main routine that controls the logical flow and performs the calculations of the time history generation.</td>
</tr>
</tbody>
</table>

1 The Normal distribution is discussed on Page 2-23.
2 The Uniform distribution is discussed on Page 2-23.

PRECEDING PAGE BLANK NOT FILMED
7.7.1.3 Description of Variables

A list of variables used in the time history generation code, NBSIN, is given in Table 7-24 below. 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; and DRE is a double precision real variable. The various array dimensions are defined by using the following parameters: MAXHIS, MAXRAN, MAXSIN, and MAXTIM.

<table>
<thead>
<tr>
<th>VARIABLE NAME</th>
<th>TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(MAXSIN)</td>
<td>DRE</td>
<td>1-D array containing the amplitudes of the sinusoidal processes.</td>
</tr>
<tr>
<td>CLIP</td>
<td>DRE</td>
<td>Peak-clipping level for the narrow-band processes.</td>
</tr>
<tr>
<td>DELTAT</td>
<td>DRE</td>
<td>The time increment ( \Delta t ) of Equation 2-60.</td>
</tr>
<tr>
<td>F</td>
<td>DRE</td>
<td>( f (\text{Hz}) ) of Equation 2-60, the frequency used to determine the time increment, usually the value of ( \max(f_0, f_c) ).</td>
</tr>
<tr>
<td>FC(MAXSIN)</td>
<td>DRE</td>
<td>1-D array containing values of the frequencies ( f_c ) (Hz) of the carrier (sinusoidal) processes.</td>
</tr>
<tr>
<td>FILNUM(MAXHIS)</td>
<td>INT</td>
<td>1-D array containing the file unit numbers for the reference time history storage files.</td>
</tr>
<tr>
<td>FO(MAXRAN)</td>
<td>INT</td>
<td>( f_0 ) (Hz) of Equation 2-58, the 1-D array containing values of the modal frequencies of the component. These are the frequencies of the narrow-band processes.</td>
</tr>
<tr>
<td>HISNAM(MAXHIS)</td>
<td>CH6</td>
<td>1-D array containing the HIStory NAMes for the time history files.</td>
</tr>
<tr>
<td>I</td>
<td>INT</td>
<td>Controls DO loop for each history and time increment.</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 sinusoidal process.</td>
</tr>
<tr>
<td>K</td>
<td>INT</td>
<td>Controls DO loop for each narrow-band process.</td>
</tr>
<tr>
<td>LASTT</td>
<td>DRE</td>
<td>Length ( T ) of the time histories to be generated.</td>
</tr>
<tr>
<td>MAXHIS</td>
<td>INT</td>
<td>Maximum total number of time histories allowed. The maximum number of time histories allowed is 29.</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>MAXRAN</td>
<td>INT</td>
<td>Maximum number of narrow-band time histories allowed. The maximum number of narrow-band histories allowed is 19.</td>
</tr>
<tr>
<td>MAXSIN</td>
<td>INT</td>
<td>Maximum number of sinusoidal time histories allowed. The maximum number of sinusoidal histories allowed is 10.</td>
</tr>
<tr>
<td>MAXTIM</td>
<td>INT</td>
<td>Maximum number of time increments allowed. The maximum number of increments is 25,000.</td>
</tr>
<tr>
<td>MU</td>
<td>DRE</td>
<td>Mean $\mu$ of the Normal distribution used to generate the narrow-band process. The value of $\mu$ must be zero.</td>
</tr>
<tr>
<td>N</td>
<td>DRE</td>
<td>$N$ of Equation 2-60, the number of points per cycle of frequency $f$.</td>
</tr>
<tr>
<td>N1, N2</td>
<td>DRE</td>
<td>Normal(0, 1) random variates.</td>
</tr>
<tr>
<td>NCI(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $N_c(t_i)$, Equation 2-57, the values of the cosine components of the narrow-band processes.</td>
</tr>
<tr>
<td>NCIOLD(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $N_c(t_{i-1})$.</td>
</tr>
<tr>
<td>NI(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $N(t_i)$, Equation 2-57, the narrow-band processes.</td>
</tr>
<tr>
<td>NRAND(0:4)</td>
<td>INT</td>
<td>1-D array containing the number of narrow-band time histories for each correlation group with NRAND(0) being the total number of narrow-band time histories.</td>
</tr>
<tr>
<td>NSI(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $N_s(t_i)$, Equation 2-57, the values of the sine components of the narrow-band processes.</td>
</tr>
<tr>
<td>NSIN</td>
<td>INT</td>
<td>The number of sinusoidal time histories.</td>
</tr>
<tr>
<td>NSIOLD(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $N_s(t_{i-1})$.</td>
</tr>
<tr>
<td>NTIM</td>
<td>INT</td>
<td>The number of time increments.</td>
</tr>
<tr>
<td>PHASE(MAXSIN)</td>
<td>DRE</td>
<td>1-D array containing the relative phase angle shifts (rad) of the sinusoidal processes.</td>
</tr>
<tr>
<td>RAND</td>
<td>DRE</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>RHO(MAXRAN)</td>
<td>DRE</td>
<td>$\rho$ of Equation 2-60, the 1-D array containing a value of the autocorrelation coefficient for each narrow-band process.</td>
</tr>
<tr>
<td>SHIFT</td>
<td>DRE</td>
<td>User-requested amount to shift all sinusoidal process phase angles (degrees).</td>
</tr>
<tr>
<td>VARIABLE NAME</td>
<td>TYPE</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SI(MAXSIN)</td>
<td>DRE</td>
<td>1-D array containing values of $S(t_i)$, the sinusoidal processes.</td>
</tr>
<tr>
<td>SIGMAC(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $\sigma_c$, the standard deviation of the cosine components of the narrow-band processes.</td>
</tr>
<tr>
<td>SIGMAN(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $\sigma_N$, Equation 2-60, the standard deviation of the narrow-band processes.</td>
</tr>
<tr>
<td>TI(0:MAXTIM)</td>
<td>DRE</td>
<td>1-D array containing the time values, $t_i$.</td>
</tr>
<tr>
<td>TWOPI</td>
<td>DRE</td>
<td>Value of $2\pi$.</td>
</tr>
<tr>
<td>UCI(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $u_c(t_i)$ the Normal($0, \sigma_c^2$) generated random variates for the cosine components of the narrow-band processes.</td>
</tr>
<tr>
<td>USI(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing values of $u_s(t_i)$ the Normal($0, \sigma_s^2$) generated random variates for the sine components of the narrow-band processes.</td>
</tr>
<tr>
<td>Note: $\sigma_s = \sigma_c$.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WC(MAXSIN)</td>
<td>DRE</td>
<td>1-D array containing values of the angular frecuencies $\omega_c$ of the sinusoidal processes.</td>
</tr>
<tr>
<td>WO(MAXRAN)</td>
<td>DRE</td>
<td>$\omega_o$, of Equation 2-57, the 1-D array containing values of the modal angular frequencies of the component. These are the angular frequencies of the narrow-band processes.</td>
</tr>
<tr>
<td>XC</td>
<td>DRE</td>
<td>The damping coefficient $\xi$ of Equation 2-58.</td>
</tr>
<tr>
<td>Z1(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing Normal($0, 1$) random variates for the cosine component of the narrow-band process.</td>
</tr>
<tr>
<td>Z2(MAXRAN)</td>
<td>DRE</td>
<td>1-D array containing Normal($0, 1$) random variates for the sine component of the narrow-band process.</td>
</tr>
</tbody>
</table>

---

1 $f_c$ is discussed on Page 2-30.
2 $T$ is discussed on Page 2-30.
3 See variable NC1().
4 See variable NS1().
5 See $u(t)$ in Equation 2-60.
7.7.1.4 Program NBSIN Listing

PROGRAM NBSIN CONTROLS THE GENERATION OF MULTIPLE SINUSOID AND NARROW-BAND PROCESSES
PROGRAMMER: L. NEWLIN
DATE: CODE: 28JAN91 COMMENTS: 11SEP91
VERSION: V5.4

Copyright (C) 1990, California Institute of Technology.
U.S. Government Sponsorship under NASA Contract NAS7-818 is acknowledged.

PROGRAM NBSIN

SUBPROGRAMS: NORMGN
FILES: 1:NBSIN-OLD; 8:IOUTPR-NEW; 11-39:user named-NEW

IMPLICIT NONE
COMMON IOUT
INTEGER MAXHIS, MAXRAN, MAXSIN, MAXTIM
DOUBLE PRECISION RAND, TWOPI
PARAMETER (MAXHIS = 29, MAXRAN = 19, MAXSIN = 10, MAXTIM = 25000,
& TWOPI = 6.283185308)

INTEGER FILNUM(MAXHIS), I, IOUT, J, K, N, NRAND(0:4), NSIN, NTIM
DOUBLE PRECISION A(MAXSIN), CLIP, DELTAT, F, FC(MAXSIN),
& FO(MAXRAN), LASTT, MU, N1, N2, NCI(MAXRAN),
& NCI0LD(MAXRAN), NI(MAXRAN), NSI(MAXRAN), NSI0LD(MAXRAN),
& PHASE(MAXSIN), RHO(MAXRAN), SHIFT, SI(MAXSIN), SIGMA,
& SIGMAC(MAXRAN), SIGNAM(MAXRAN), TI(O:MAXTIM),
& UCI(MAXRAN), USI(MAXRAN), WC(MAXSIN), WO(SAXRAN), XC,
& z1(MAXRAN), z2 (MAXRAN)

CHARACTER*6 HISNAM(MAXHIS)

DATA (FILNUM(I), I = 1, MAXHIS) /
& 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
& 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
& 31, 32, 33, 34, 35, 36, 37, 38, 39 /

LIST OF VARIABLES

A() 1-D ARRAY CONTAINING THE AMPLITUDES OF SINUSOIDAL PROCESSES
CLIP PEAK-CLIPPING LEVEL
DELTAT TIME INCREMENT
F FREQUENCY USED TO DETERMINE TIME INCREMENT
FC() 1-D ARRAY CONTAINING FREQUENCIES OF SINUSOIDAL PROCESSES
FILNUM() 1-D ARRAY CONTAINING UNIT NUMBERS FOR TIME HISTORY FILES
FO() 1-D ARRAY CONTAINING FREQUENCIES OF NARROW-BAND PROCESSES
HISNAM() 1-D ARRAY CONTAINING HISTORY NAMES FOR THE TIME HISTORIES
I CONTROLS DO LOOP FOR EACH TIME INCREMENT
IOUT OUTPUT DUMP CONTROLLER
J CONTROLS DO LOOP FOR EACH SINUSOIDAL PROCESS
K CONTROLS DO LOOP FOR EACH NARROW-BAND PROCESS
LASTT LARGEST TIME VALUE TO BE CONSIDERED (LENGTH OF GENERATED TIME HISTORY)
MAXHIS MAXIMUM TOTAL NUMBER OF TIME HISTORIES ALLOWED
MAXRAN MAXIMUM NUMBER OF NARROW-BAND RANDOM HISTORIES ALLOWED
MAXSIN MAXIMUM NUMBER OF SINUSOIDAL HISTORIES ALLOWED
MAXTIM MAXIMUM NUMBER OF TIME INCREMENTS ALLOWED
MU MEAN OF NARROW-BAND PROCESS -- EQUAL TO ZERO
N NUMBER OF POINTS PER CYCLE OF FREQUENCY F
N1 NORMAL(0,1) RANDOM VARIATE
N2 NORMAL(0,1) RANDOM VARIATE
NCI() 1-D ARRAY CONTAINING VALUES OF Nc(i), THE MAGNITUDES OF
THE COSINE COMPONENTS OF THE NARROW-BAND PROCESSES
NCIOLD() 1-D ARRAY CONTAINING VALUES OF Nc(i-l)
NC() 1-D ARRAY CONTAINING VALUES OF Nc(i), THE NARROW-BAND PROCESSES
NRAND() 1-D ARRAY CONTAINING THE NUMBER OF NARROW-BAND TIME HISTORIES
FOR EACH CORRELATION GROUP WITH NRAND(0) BEING THE TOTAL
NUMBER OF NARROW-BAND TIME HISTORIES
NSI() 1-D ARRAY CONTAINING VALUES OF Ns(i), THE MAGNITUDES OF
THE SINE COMPONENTS OF THE NARROW-BAND PROCESSES
NSIN() NUMBER OF SINUSOIDAL HISTORIES
NSIOD() 1-D ARRAY CONTAINING VALUES OF Ns(i-l)
NTIM NUMBER OF TIME INCREMENTS
PHASE() 1-D ARRAY CONTAINING THE RELATIVE PHASE ANGLE SHIFTS OF THE
SINUSOIDAL PROCESSES (RADIANS)
RAND RANDOM NUMBER SEED
RHO() 1-D ARRAY CONTAINING VALUES OF THE AUTOCORRELATION COEFFICIENT
FOR EACH NARROW-BAND PROCESS
SHIFT AMOUNT TO SHIFT ALL SINE PHASE ANGLES (DEGREES)
SI() 1-D ARRAY CONTAINING VALUES OF S(i), THE SINUSOIDAL
PROCESSES
SIGMA STANDARD DEVIATION OF NORMAL DISTRIBUTION -- EQUAL (CONSTRAINED)
TO ZERO
SIGMAC() 1-D ARRAY CONTAINING VALUES OF STANDARD DEVIATION OF COSINE
COMPONENTS
SIGMAN() 1-D ARRAY CONTAINING VALUES OF STANDARD DEVIATION OF
NARROW-BAND PROCESSES
TI() 1-D ARRAY CONTAINING TIME VALUES, Ti
TWOPI 2 * PI
UCI() 1-D ARRAY CONTAINING NORMAL(0,SIGMAC) GENERATED RANDOM VARIATE
FOR COSINE COMPONENTS
USI() 1-D ARRAY CONTAINING NORMAL(0,SIGMAS) GENERATED RANDOM VARIATE
FOR SINE COMPONENTS -- SIGMAS = SIGMAC
WC() 1-D ARRAY CONTAINING ANGULAR FREQUENCIES OF THE SINUSOIDAL
PROCESSES
WO() 1-D ARRAY CONTAINING ANGULAR FREQUENCIES OF NARROW-BAND
PROCESSES
XC DAMPING COEFFICIENT
ZI() 1-D ARRAY CONTAINING NORMAL(0,1) RANDOM VARIATES FOR THE COSINE
COMPONENT OF THE NARROW-BAND PROCESS
Z2() 1-D ARRAY CONTAINING NORMAL(0,1) RANDOM VARIATES FOR THE SINE
COMPONENT OF THE NARROW-BAND PROCESS

C ** READ DATA FROM FILE NBSIN
OPEN (1, FILE = 'NBSIN', STATUS = 'OLD')
OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
READ(1,*) RAND, IOUT, P, XC, N, LASTT, CLIP, SHIFT, NRAND(1),
& NRAND(2), NRAND(3), NRAND(4), NSIN
NRAND(0) = NRAND(1) + NRAND(2) + NRAND(3) + NRAND(4)
DO 10 K = 1, NRAND(0)
READ(1,*) HISMAM(K), SIGMAN(K), FO(K)
10 CONTINUE
DO 20 J = 1, NSIN
READ(1,*) HISMAM(J+NRAND(0)), A(J), FC(J), PHASE(J)
PHASE(J) = (PHASE(J) + SHIFT) * TWOPI / 360.D0
WC(J) = TWOPI * FC(J)
20 CONTINUE
CLOSE (1)

C ** PERFORM PRELIMINARY CALCULATIONS
DELTAT = 1.D0 / (F * DFLOAT (N))
TI(0) = - DELTAT
MU = 0.D0
SIGMA = 1.D0

DO 30 K = 1, NRAND(0)
RHO(K) = DEXP ((( - TWOPI * XC / DFLOAT(N)) * (FO(K) / F))
SIGMAC(K) = SIGMAN(K) * DSQRT (1.D0 - RHO(K) ** 2)
WO(K) = TWOPI * FO(K)
30 CONTINUE

C ** INITIALIZE NARROW-BAND PROCESSES

IF (NRAND(1) .NE. 0) THEN
  CALL NORMGN (RAND, MU, SIGMA, N1, N2)
  DO 31 K = 1, NRAND(1)
    Z1(K) = N1
    Z2(K) = N2
  31 CONTINUE
ENDIF

IF (NRAND(2) .NE. 0) THEN
  CALL NORMGN (RAND, MU, SIGMA, N1, N2)
  DO 32 K = (1+NRAND(1)), (NRAND(1)+NRAND(2))
    Z1(K) = N1
    Z2(K) = N2
  32 CONTINUE
ENDIF

IF (NRAND(3) .NE. 0) THEN
  CALL NORMGN (RAND, MU, SIGMA, N1, N2)
  DO 33 K = (1+NRAND(1)+NRAND(2)), NRAND(0)
    Z1(K) = N1
    Z2(K) = N2
  33 CONTINUE
ENDIF

DO 40 K = 1, NRAND(0)
  NCIOLD(K) = SIGMAN(K) * Z1(K)
  NSIOLD(K) = SIGMAN(K) * Z2(K)
40 CONTINUE

IF (IOUT .EQ. 10) THEN
  WRITE(8,*), 'TI = ', TI(0)
  WRITE(8,*), 'NRAND0 = ', NRAND(0)
  WRITE(8,*), 'NRAND1 = ', NRAND(1), ' NRAND2 = ', NRAND(2)
  WRITE(8,*), 'NRAND3 = ', NRAND(3), ' NRAND4 = ', NRAND(4)
  DO 50 J = 1, NSIN
    WRITE(8,*), 'FC = ', FC(J), ' WC = ', WC(J), ' PHASE = ',
    & PHASE(J)
  50 CONTINUE
ENDIF

C ** OPEN FILES FOR NARROW-BAND RANDOM HISTORY STORAGE

DO 70 K = 1, NRAND(0)
  OPEN (FILNUM(K), FILE = HISNAM(K), STATUS = 'NEW')
70 CONTINUE

C ** GENERATE TIME HISTORIES
C ** FIRST GENERATE NARROW-BAND PROCESSES

DO 100 I = 1, MAXTIM
  TI(I) = TI(I-1) + DELTAT
  IF (TI(I) .GT. LASTT) GOTO 150
  IF (NRAND(1) .NE. 0) THEN
    CALL NORMGN (RAND, MU, SIGMA, N1, N2)
  100 CONTINUE
DO 101 K = 1, NRAND(1)
   Z1(K) = N1
   Z2(K) = N2
101 CONTINUE
ENDIF

IF (NRAND(2) .NE. 0) THEN
   CALL NORMGN (RAND, MU, SIGMA, N1, N2)
   DO 102 K = (1+NRAND(1)), (NRAND(1)+NRAND(2))
      Z1(K) = N1
      Z2(K) = N2
102 CONTINUE
ENDIF

IF (NRAND(3) .NE. 0) THEN
   CALL NORMGN (RAND, MU, SIGMA, N1, N2)
   DO 103 K = (1+NRAND(1)+NRAND(2)), NRAND(0)
      Z1(K) = N1
      Z2(K) = N2
103 CONTINUE
ENDIF

IF (NRAND(4) .NE. 0) THEN
   CALL NORMGN (RAND, MU, SIGMA, N1, N2)
   DO 104 K = (1+NRAND(1)+NRAND(2)+NRAND(3)), NRAND(0)
      Z1(K) = N1
      Z2(K) = N2
104 CONTINUE
ENDIF

DO 110 K = 1, NRAND(0)
   UCI(K) = SIGMAC(K) * Z1(K)
   USI(K) = SIGMAC(K) * Z2(K)
   NCI(K) = RHO(K) * NCIOLD(K) + UCI(K)
   NSI(K) = RHO(K) * NSIOLD(K) + USI(K)
   NI(K) = NCI(K) * DCOS (WO(K) * TI(I))
   & NI(K) = NCI(K) * DSIN (WO(K) * TI(I))
110 CONTINUE
C ** CLIP PEAKS
DO 125 K = 1, NRAND(0)
   IF (NI(K) .LT. 0.0) THEN
      NI(K) = MAX (NI(K), - CLIP)
   ELSE
      NI(K) = MIN (NI(K), CLIP)
   ENDIF
125 CONTINUE
C ** WRITE TO FILES
DO 130 K = 1, NRAND(0)
   WRITE(FILNUM(K), *) NI(K)
130 CONTINUE
C ** UPDATE RECURSIVE PARAMETERS
DO 140 K = 1, NRAND(0)
   NCIOLD(K) = NCI(K)
   NSIOLD(K) = NSI(K)
140 CONTINUE
CONTINUE
CONTINUE
CONTINUE
  IF (IOUT .EQ. 10) WRITE(8,*) 'TI = ', TI(I)
  NTIM = I - 1
C ** CLOSE FILES FOR NARROW-BAND RANDOM HISTORY STORAGE
  DO 160 K = 1, NRAND(0)
    CLOSE (FILNUM(K))
  CONTINUE
C ** OPEN FILES FOR SINUSOIDAL HISTORY STORAGE
  DO 170 J = 1, NSIN
    OPEN (FILNUM(J+NRAND(0)), FILE = HISNAME(J+NRAND(0)), &
         STATUS = 'NEW')
  CONTINUE
C ** NOW GENERATE SINUSOIDAL PROCESSES
  DO 200 I = 1, NTIM
    DO 210 J = 1, NSIN
      SI(J) = A(J) * DCOS (WC(J) * TI(I) + PHASE(J))
    CONTINUE
C ** WRITE TO FILES
  DO 220 J = 1, NSIN
    WRITE(FILNUM(J+NRAND(0)),*) SI(J)
  CONTINUE
  IF (IOUT .EQ. 10) THEN
    DO 230 J = 1, NSIN
      WRITE(8,*) 'J = ', J, ' SI = ', SI(J)
    CONTINUE
  ENDIF
200 CONTINUE
STOP
END

C SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C PROGRAMMER: L. GRONDAISLKI, L. NEWLIN
C DATE: 14APR88
C VERSION: NBSIN VI, V1.1, V2, V3, V4, V5, V5.1, V5.3, V5.4
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, X1, X2)
C IMPLICIT NONE
COMMON IOUT
DOUBLE PRECISION RAND, MU, PI, SIGMA, X1, X2, U1, U2, Z1, Z2

7 - 495
REAL FRAC

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
X1 NORMAL RANDOM VARIATE NUMBER ONE
X2 NORMAL RANDOM VARIATE NUMBER TWO
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 = DBLE (FRAC)
CALL RANDOM (FRAC, RAND)
U2 = DBLE (FRAC)
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'U1 =', U1, ' U2 =', U2
Z1 = DSQRT (-2.D0 * DLOG(U1)) * DSIN(2.D0 * PI * U1)
Z2 = DSQRT (-2.D0 * DLOG(U1)) * DCOS(2.D0 * PI * U2)
X1 = SIGMA * Z1 + MU
X2 = SIGMA * Z2 + MU
IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
& WRITE(8,*) 'X1 =', X1, ' X2 =', X2
RETURN
END
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  RANDSUB  INTERNAL CALCULATION
C  RANT   INTERNAL CALCULATION
C  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 = INT(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