JPL Publication 92-15, Vol. III

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.

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,[1] 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[2] 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

VOLUME II – Software Documentation

List of Figures

List of Tables

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 7-1 List of Subprograms For Program DCTHCF (Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
ADDREG[1]	4.1.3.9	Adds the *m* ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.
ADDRGN[1]	4.1.3.15	Adds the *m* ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.
BETAGN[2]	4.4.5	Generates Beta(a, b, ρ, θ) random variates.
CALCS	5.1.2.4	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.
CONCAV[3]	4.1.3.10	Adjusts the upper bound of the posterior ranges on *m* to be consistent with concavity constraints.
CONVRT[4]	4.1.3.3	Transforms stress data to equivalent zero-mean stresses with stress ratio of -1.0.

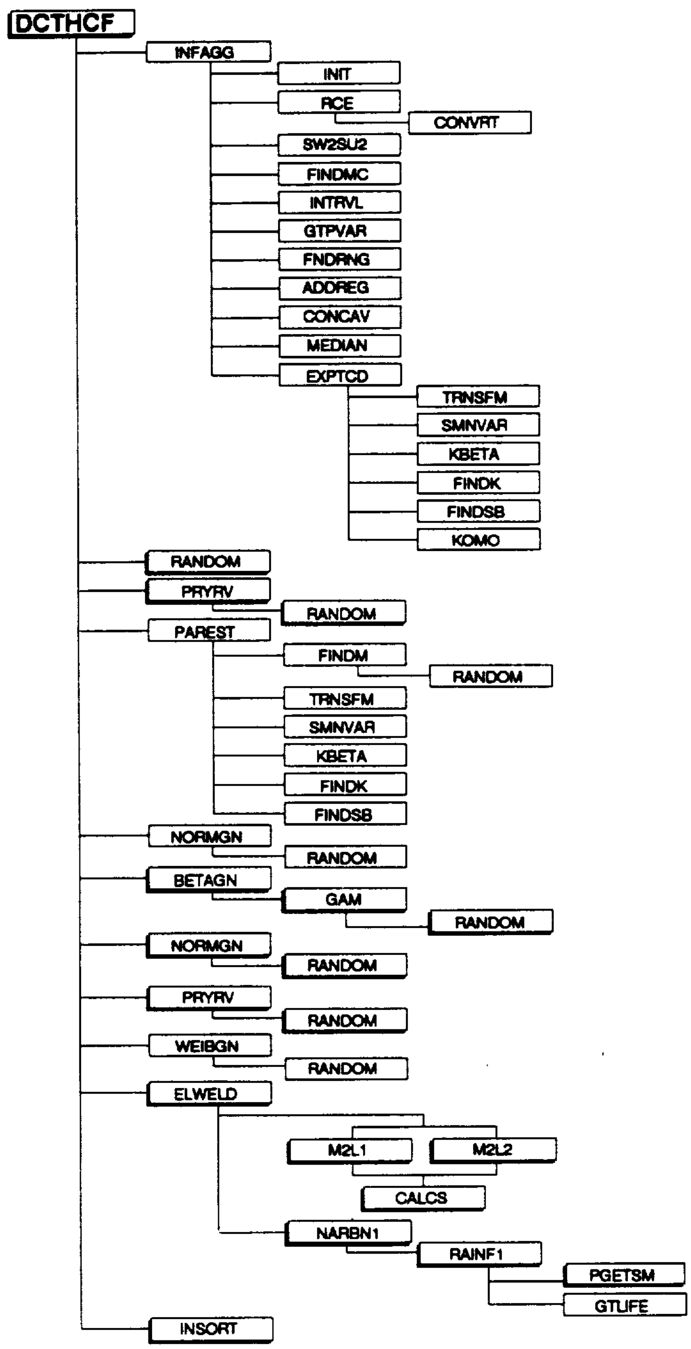

Figure 7-1 Tree Structure for Program DCTHCF for the Uniform Variation in the Materials Shape Parameter *m*

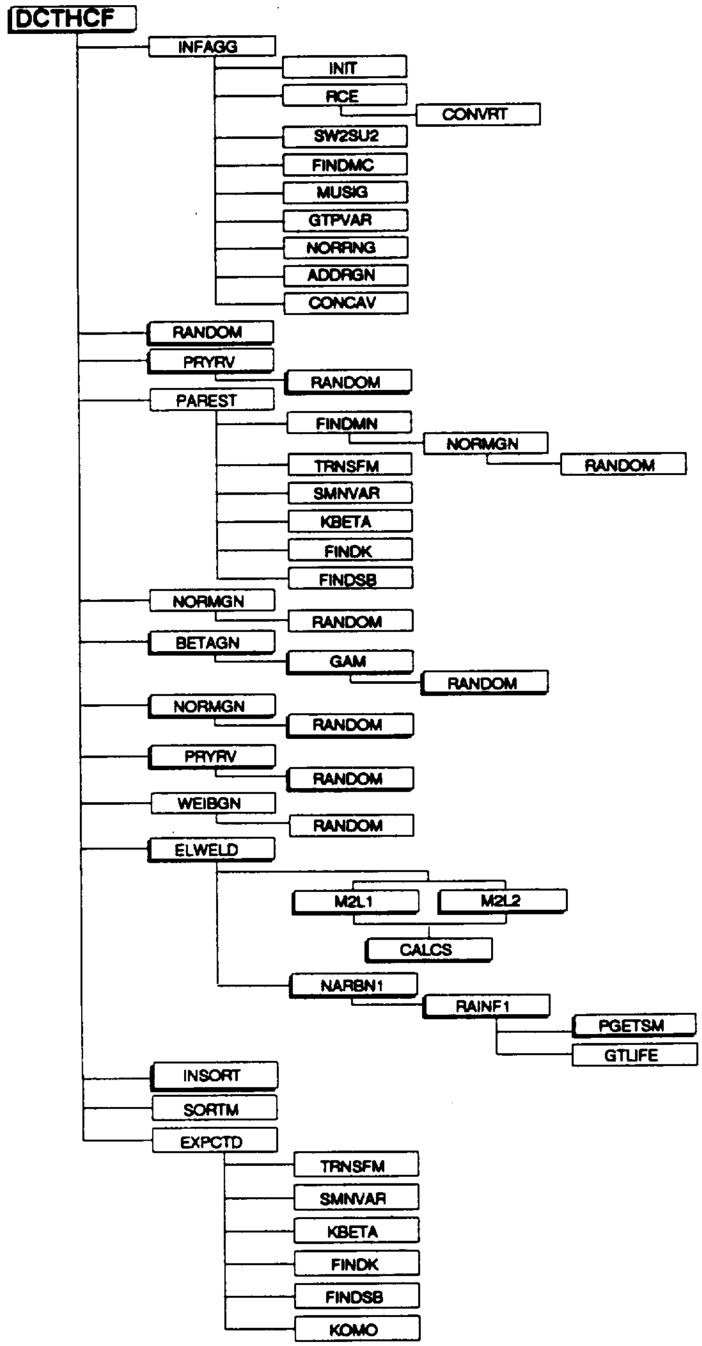

Figure 7-2 Tree Structure for Program DCTHCF for the Truncated Normal Variation in the Materials Shape Parameter *m*

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

NAME	SECTION	PURPOSE
DCTHCF	5.1.2.1	The main routine that controls the logical flow of the high cycle fatigue elbow welded duct program.
ELWELD	5.1.2.2	Controls the logical flow for the driver transformation and fatigue life calculations.
EXPCTD[5]	4.1.3.12	Calculates the median S/N curve parameters from the results of the information aggregation calculations.
FINDK	4.1.5.6	Calculates the value of the location parameter K (where $A = K^m$) for each life region by using *Equations 2-37* and *2-41*.
FINDM[6]	4.1.5.1	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.
FINDMC	4.1.3.5	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*.
FINDMN[6]	4.1.5.2	Obtains the value of m for each life region by sampling from the appropriate truncated Normal distribution on m.
FINDSB	4.1.5.7	Calculates the life region "tie-points" or stress values which correspond to the "life boundaries" conditional on the randomly selected m for each region. Also calculates K, characterizing the specific material S/N data set, which is a function of β_o and k.
FNDRNG[7]	4.1.3.8	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.
GAM	4.4.4	Generates Gamma(α, 1) random variates.
GTLIFE	4.1.8	Calculates the cycles to failure for a particular stress based upon the materials characterization model S/N curve of *Equation 2-48*.
GTPVAR	4.1.3.7	Calculates σ^2, *Equation 2-49*, the extent of departures from the multiple heat median S/N curve warranted by the information available.
INFAGG[8]	4.1.3	Controls the logical flow for the information aggregation portion of the materials characterization model.
INIT	4.1.3.1	Initializes the entries of the arrays used in the information aggregation subroutine, INFAGG, to zero.
INSORT	5.B	Performs an insertion sort for the lowest fifty percent of the lives calculated.
INTRVL	4.1.3.6	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*.

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

NAME	SECTION	PURPOSE
KBETA	4.1.5.5	Calculates *k* and β_0 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.*
KOMO[9]	4.1.6	Calculates K_0 and m_0 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_0. Disabled for this application.
M2L1	5.1.2.3	Performs the calculations, *Equations 2-73* through *2-80,* necessary to find the stress at location 1, the exterior surface of the duct.
M2L2	5.1.2.3	Performs the calculations, *Equations 2-73* through *2-80,* necessary to find the stress at location 2, the interior surface of the duct.
MEDIAN	4.1.3.11	Calculates the median values of *m* based on the posterior credibility ranges of *m* by using *Equation 2-34.*
MUSIG[10]	4.1.3.13	Calculates the posterior Normal distribution parameters: mean m_* and standard deviation σ_*, for each life region of the S/N curve.
NARBN1	5.1.2.5	Calculates the composite stress-time history by using *Equations 2-82* and *2-84* and then calls RAINF1 to calculate the fatigue life.
NORMGN[11]	4.4.3	Generates Normal(μ, σ^2) random variates.
NORRNG[7]	4.1.3.14	Combines the implicit and explicit constraints on *m* to obtain the posterior credibility ranges of *m* for each life region.
PAREST[12]	4.1.5	Controls the logical flow for the parameter estimation model portion of the materials characterization model.
PGETSM	5.1.2.7	Calculates the equivalent mean stress from the maximum stress by using *Equation 2-87.*
PRYRV[13]	7.6.6	Generates the Uniform(*a*, *b*) and Uniform(*c*, *d*) pair of independent random variates.
RAINF1[14]	5.1.2.6	Performs rainflow cycle counting, Miner's rule damage accumulation, and calls GTLIFE to calculate the fatigue life.
RANDOM[13]	4.4.2	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.
RCE	4.1.3.2	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.
SMNVAR	4.1.5.4	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.*
SORTM[15]	4.1.10	Sorts the *m* values in increasing order for each life region for the truncated Normal distribution case.

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

NAME	SECTION	PURPOSE
SW2SU2	4.1.3.4	Calculates the residual variances from the *Y* on *X* and *X* on 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.
TRMNAT	4.1.11	Performs premature program termination, when required.
TRNSFM[16]	4.1.5.3	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.
WEIBGN	4.4.6	Generates Weibull(β, $\eta(\beta)$) random variates.

---

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 7-2 List of Variables for Program DCTHCF
(Footnotes are at the end of the table)

VARIABLE NAME	TYPE	DESCRIPTION
ALLM(MAXMM, MAXREG)	RE	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]
ANGLE	RE	ϕ (rad) in *Equation 2-68*, the angle measured counterclockwise from Z-direction to the critical circumferential location.
AREA	RE	A (in.^{2}) in *Equation 2-68*, the cross-sectional area of the duct wall.
B	RE	β in *Equation 2-79*, the stress increase due to the torus effect.
BIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter K, *Equation 2-12*, where $A = K^m$.
BIGK1	RE	Dummy variable used during calls to subroutine EXPCTD, equal to **BIGK(1)**.
BLFPER(MAXBLF)	RE	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%.
BLFPOS(MAXBLF)	INT	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( )**.
BNRD	RE	R_B (in.) in *Equation 2-74*, the elbow BeNd RaDius. (see **RB** in ELWELD)

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
BZERO	RE	Estimate of Weibull distribution shape parameter β_o, *Equation 2-11*, which characterizes the intrinsic variation of the S/N data set.
CCY	RE	C_{cy} in *Equation 2-77*, the out-of-plane circumferential stress carryover factor.
CCYA	RE	C_{cy} Uniform distribution lower bound.
CCYB	RE	C_{cy} Uniform distribution upper bound.
CCZ	RE	C_{cz} in *Equation 2-75*, the in-plane circumferential stress carryover factor.
CCZA	RE	C_{cz} Uniform distribution lower bound.
CCZB	RE	C_{cz} Uniform distribution upper bound.
CLY	RE	C_{ly} in *Equation 2-76*, the out-of-plane axial stress carryover factor.
CLYA	RE	C_{ly} Uniform distribution lower bound.
CLYB	RE	C_{ly} Uniform distribution upper bound.
CLZ	RE	C_{lz} in *Equation 2-74*, the in-plane axial stress carryover factor.
CLZA	RE	C_{lz} Uniform distribution lower bound.
CLZB	RE	C_{lz} Uniform distribution upper bound.
CULPRT	INT	Location about duct circumference responsible for failure. See variable **LOCAT** for the possible locations.
DI	RE	D_i (in.) the duct inner diameter at the weld, used to calculate R_i in *Equation 2-68*. (see **IDWE** in DCTHCF)
DSTR	RE	$\lambda_{DYN_{str}}$ in *Equation 2-81*, the randomly selected dynamic stress analysis accuracy factor.
DSTRA	RE	Dynamic stress analysis accuracy factor Uniform distribution lower bound.
DSTRB	RE	Dynamic stress analysis accuracy factor Uniform distribution upper bound.
ELWELD	RE	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).

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
EM	RE	E (psi) in *Equation 2-70*, Young's modulus of elasticity for the material. (see **EMOD** in DCTHCF)
EMOD	RE	E (psi) in *Equation 2-70*, Young's modulus of elasticity for the material. (see **EM** in ELWELD)
FATLIF	RE	Value of FATigue LIFe calculated (sec).
FIFTY	RE	Variable used to access the fifty-percent point in the **LIFE( )** array.
FILNUM(MAXLD)	INT	1-D array containing the file unit numbers for the reference time history files.
FK(10)	RE	1-D array containing values of F_k, *Equation 2-73*, used to find stress concentration due to weld eccentricity, K_{OFF}.
FTEST	LOG	File TEST. Used to test for the existence of a reference time history file before attempting to open it.
FTU	RE	Material ultimate strength (psi).
FTY	RE	Material yield strength (psi).
GAM	RE	λ_{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.
GAMA	RE	Damage accumulation model accuracy factor Uniform distribution lower bound.
GAMB	RE	Damage accumulation model accuracy factor Uniform distribution upper bound.
GCY	RE	γ_{cy} in *Equation 2-77*, the out-of-plane circumferential ovality effect coefficient.
GCZ	RE	γ_{cz} in *Equation 2-75*, the in-plane circumferential ovality effect coefficient.
GLY	RE	γ_{ly} in *Equation 2-76*, the out-of-plane axial ovality effect coefficient.
GLZ	RE	γ_{lz} in *Equation 2-74*, the in-plane axial ovality effect coefficient.
GNBI	RE	γ_{nbi} in *Equations 2-74* through *2-77*.
GNBO	RE	γ_{nbo} in *Equations 2-74* through *2-77*.
GTMI	RE	γ_{tmi} in *Equations 2-74* through *2-77*.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
GTMO	RE	γ_{tmo} in *Equations 2-74* through *2-77.*
I	INT	Controls inner DO loop.
IDWE	RE	D_i (in.) the duct Inner Diameter at the WEld, used to calculate R_i in *Equation 2-68.* (see **DI** in ELWELD)
II	INT	Controls DO loop for narrow-band random and superimposed sinusoidal loads.
IOUT	INT	Output dump controller.
J	INT	Controls DO loop for each B-life.[2]
K	INT	Controls outer DO loop.
K(2, 2)	RE	2-D array containing the fatigue stress concentration factors required for the stress analysis. **K(1,*)** is K_{T1} in *Equation 2-68* and **K(2,*)** is K_{T2} in *Equation 2-69*. **K(1,1)** is the outer diameter axial stress concentration factor, the value of **KGOD * KWOD**; **K(1,2)** is the inner diameter axial stress concentration factor, the value of **KGID * 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)
KGID	RE	Axial stress concentration factor due to geometry for the duct inner diameter used to calculate K_{T1} in *Equation 2-68.*
KGOD	RE	Randomly selected axial stress concentration factor due to geometry for the duct outer diameter used to calculate K_{T1} in *Equation 2-68.*
KGODA	RE	Outer diameter geometric axial stress concentration factor lower bound of Beta distribution.
KGODB	RE	Outer diameter geometric axial stress concentration factor upper bound of Beta distribution.
KGODR	RE	Randomly selected Beta distribution location parameter ρ for the outer diameter geometric axial stress concentration factor.
KGODR1	RE	ρ Uniform distribution lower bound of Beta distribution of the outer diameter geometric axial stress concentration factor.
KGODR2	RE	ρ Uniform distribution upper bound of Beta distribution of the outer diameter geometric axial stress concentration factor.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
KGODT	RE	Randomly selected Beta distribution shape parameter θ for the outer diameter geometric axial stress concentration factor.
KGODT1	RE	θ Uniform distribution lower bound of Beta distribution of the outer diameter geometric axial stress concentration factor.
KGODT2	RE	θ Uniform distribution upper bound of Beta distribution of the outer diameter geometric axial stress concentration factor.
KOFF	RE	K_{OFF} in *Equation 2-73,* the stress concentration factor due to eccentricity of the weld.
KRATIO	RE	Ratio of *MED K*/MED K* in *Equation 2-48.* **KRATIO** is constant over life regions for the materials model.
KT(2, 2)	RE	2-D array containing the fatigue stress concentration factors required for the stress analysis. **KT(1,*)** is K_{T_1} in *Equation 2-68* and **KT(2,*)** is K_{T_2} 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)
KT1	RE	K_{T_1} in *Equation 2-68,* the stress concentration factor for the axial stress.
KT2	RE	K_{T_2} in *Equation 2-69,* the stress concentration factor for the hoop stress.
KWID	RE	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.*
KWIDA	RE	Inner diameter weld axial stress concentration factor lower bound of Beta distribution.
KWIDB	RE	Inner diameter weld axial stress concentration factor upper bound of Beta distribution.
KWIDR	RE	Randomly selected Beta distribution location parameter ρ for the inner diameter weld axial stress concentration factor.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
KWIDR1	RE	ρ Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWIDR2	RE	ρ Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWIDT	RE	Randomly selected Beta distribution shape parameter θ for the inner diameter weld axial stress concentration factor.
KWIDT1	RE	θ Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWIDT2	RE	θ Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWOD	RE	Randomly selected axial stress concentration factor due to the weld for the duct outer diameter used to calculate K_{T1} in *Equation 2-68*.
KWODA	RE	Outer diameter weld axial stress concentration factor lower bound of Beta distribution.
KWODB	RE	Outer diameter weld axial stress concentration factor upper bound of Beta distribution.
KWODR	RE	Randomly selected Beta distribution location parameter ρ for the outer diameter weld axial stress concentration factor.
KWODR1	RE	ρ Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.
KWODR2	RE	ρ Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.
KWODT	RE	Randomly selected Beta distribution shape parameter θ of the outer diameter weld axial stress concentration factor.
KWODT1	RE	θ Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
KWODT2	RE	θ Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.
L	INT	Controls DO loop for each life region of the S/N curve.
L	RE	λ in *Equations 2-74* through *2-77.*
LAMN	RE	λD_{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.
LAMNA	RE	Lower bound of the Uniform distribution of k for the narrow-band random load scale factor.
LAMNB	RE	Upper bound of the Uniform distribution of k for the narrow-band random load scale factor.
LAMNC	RE	Coefficient of variation C for the narrow-band random load scale factor.
LAMND	RE	Strain gage correction factor d for the narrow-band random load scale factor.
LAMNK	RE	Randomly selected k for the narrow-band random load scale factor.
LAMNMU	RE	The resulting mean μ of the Normal distribution for the narrow-band random load scale factor, where $\mu = d/(1 + kC)$.
LAMNSG	RE	The resulting standard deviation σ of the Normal distribution for the narrow-band random load scale factor, where $\sigma = C/(1 + kC)$.
LAMS	RE	$\lambda D_{SINUSOIDAL}$ 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.
LAMSA	RE	Lower bound of the Uniform distribution of k for the superimposed sinusoidal load scale factor.
LAMSB	RE	Upper bound of the Uniform distribution of k for the superimposed sinusoidal load scale factor.
LAMSC	RE	Coefficient of variation C for the superimposed sinusoidal load scale factor.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LAMSD	RE	Strain gage correction factor *d* for the superimposed sinusoidal load scale factor.
LAMSK	RE	Randomly selected *k* for the superimposed sinusoidal load scale factor.
LAMSMU	RE	The resulting mean μ of the Normal distribution for the superimposed sinusoidal load scale factor, where $\mu = d/(1 + kC)$.
LAMSSG	RE	The resulting standard deviation σ of the Normal distribution for the superimposed sinusoidal load scale factor, where $\sigma = C/(1 + kC)$.
LAMST	RE	λ_{ST} in *Equation 2-81,* the randomly selected load scale factor for the static loads.
LAMSTA	RE	Uniform distribution lower bound for the static load scale factor.
LAMSTB	RE	Uniform distribution upper bound for the static load scale factor.
LAMW	RE	LAMbda Weld offset, the randomly selected λ_{OFF} in *Equation 2-73*, the accuracy factor for the weld offset eccentricity stress concentration factor, K_{OFF}.
LAMWA	RE	λ_{OFF} Uniform distribution lower bound.
LAMWB	RE	λ_{OFF} Uniform distribution upper bound.
LDNAME(MAXLD)	CH6	1-D array containing LoaD NAMEs for the dynamic or time-varying loads. These are the names of the reference time history files.
LIFE(MAXLIF)	RE	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.
LIMPR	RE	p_i (psi) in *Equation 2-68,* the LIMit or internal PRessure. (see **PSUBI** in ELWELD)
LNA(0:MAXREG)	RE	1-D array containing values of ln(*A*) = ln(**BIGK**) * **MM** for each life region of the S/N curve.
LNZ	RE	ln(*Z*) in *Equation 2-48,* the Normal(0, **PVAR**) random variate for the materials process variation aspect of the materials model.
LOCAT	INT	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.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LPHIM(0:MAXREG)	RE	1-D array containing values of ln(PHI) * **MM** for each life region of the S/N curve.
M(2, MAXLD)	RE	2-D array containing the dynamic or time-varying moment load components. **M(1,*)** is M_y (in.-lbs) in *Equation 2-68*, the moment load components about the y axis; and **M(2,*)** is M_z (in.-lbs) in *Equation 2-68*, the moment load components about the z axis.
MAXBLF	INT	Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.[2]
MAXDAT	INT	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.
MAXLD	INT	Maximum number of dynamic or time-varying loads allowed. The maximum number of loads is 16.
MAXLIF	INT	Maximum number of fatigue lives allowed for the simulated failure distribution. The maximum number of fatigue lives to be saved is 10,000.
MAXM	INT	Maximum number of points allowed in the time history arrays. The maximum number of points is 24,000.
MAXMM	INT	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.
MAXREG	INT	Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.
MCOUNT	INT	Counts number of *m*'s to be used to calculate median S/N curve for the truncated Normal distribution case.[1]
MEDM(MAXMM)	RE	1-D array containing the empirical median *m* for each life region of the S/N curve.[3]
MI	RE	*I* (in.^{4}) in *Equation 2-68*, the cross-sectional Moment of Inertia.
MID	INT	Pointer to the median *m* values in array **SORTM( )** for the truncated Normal median S/N curve. Value of half of **MCOUNT**.
MIIB	RE	*I* (in.^{4}) in *Equation 2-68*, the cross-sectional moment of inertia calculated by using the wall thickness at the inner bend.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
MIOB	RE	I (in.^{4}) in *Equation 2-68*, the cross-sectional moment of inertia calculated by using the wall thickness at the outer bend.
MLAM(2, MAXLD)	RE	2-D array containing the dynamic or time-varying moment load components scaled by **DSTR** and **LAMS** or **LAMN**, as appropriate, according to variable **TYPE( )**. **MLAM(1,*)** is M_y (in.-lbs) in *Equation 2-68*, the moment load components about the y axis; and **MLAM(2,*)** is M_z (in.-lbs) in *Equation 2-68*, the moment load components about the z axis.
MM(0:MAXREG)	RE	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.
MNWT	RE	tw_1 (in.) the duct MINimum Wall Thickness at the weld outer diameter. (see **TOB** in ELWELD)
MPROC	INT	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]
MSLAM(2)	RE	1-D array containing static moment load components scaled by **SSTR** and **LAMST**. **MSLAM(1)** is M_y (in.-lbs) in *Equation 2-68*, the moment load component about the y axis; and **MSLAM(2)** is M_z (in.-lbs) in *Equation 2-68*, the moment load component about the z axis.
MSTAT(2)	RE	1-D array containing the static moment load components. **MSTAT(1)** is M_y (in.-lbs) in *Equation 2-68*, the moment load component about the y axis; and **MSTAT(2)** is M_z (in.-lbs) in *Equation 2-68*, the moment load component about the z axis.
MU(MAXREG)	RE	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.
NBLIFE	INT	Number of B-lives to be obtained from the simulated failure distribution.[2]
NBND(0:MAXREG)	RE	$N^*_{i,i+1}$ 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.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NEWLIF	RE	Fatigue life value (sec) returned from call to function ELWELD.
NF(MAXDAT, MAXREG)	RE	2-D array containing values from the array **RAWNF( )** for the specific material S/N data set partitioned into life regions.
NHYPER	INT	The outer loop size.
NLIFE	INT	The inner loop size.
NLIFET	INT	Total number of lives calculated by program DCTHCF. Value of **NHYPER * NLIFE**.
NLOAD	INT	NLOAD in *Equation 2-81*, the number of dynamic or time-varying loads.
NMED	INT	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.
NPTS(MAXREG)	INT	1-D array containing the number of points per life region for the specific material S/N data set.
NRAN	INT	Number of RANdom points. Number of points in the reference time history.
NU	RE	ν in *Equations 2-74* through *2-77*, the Poisson's ratio for the duct material.
NUMREG	INT	R in *Equation 2-11*, the number of life regions of interest in the S/N curve.
OVAL	RE	λ_{oval} in *Equations 2-74* through *2-77*, the randomly selected ovality effect analysis accuracy factor.
OVALA	RE	Uniform distribution lower bound for the ovality effect analysis accuracy factor.
OVALB	RE	Uniform distribution upper bound for the ovality effect analysis accuracy factor.
P(MAXLD)	RE	1-D array containing P (lbs) in *Equation 2-68*, the dynamic or time-varying axial load components.
PERIOD	RE	T (sec) in *Equation 2-91*, the length of time in seconds of the reference time history.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
PHI	RE	φ in *Equation 2-11,* the materials intrinsic variation, or scatter, given by a Weibull(β_o, $\eta_o(\beta_o)$) random variate.
PI	RE	π, constant equal to 3.1415926536.
PLAM(MAXLD)	RE	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( )**.
PSI	RE	ψ in *Equations 2-74* through *2-77.*
PSIG	RE	σ in *Equation 2-48,* the value of SQRT(**PVAR**).
PSLAM	RE	*P* (lbs) in *Equation 2-68,* the static axial load component scaled by **SSTR** and **LAMST**.
PSTAT	RE	*P* (lbs) in *Equation 2-68,* the static axial load component.
PSUBI	RE	p_i (psi) in *Equation 2-68,* the limit or internal pressure. (see **LIMPR** in DCTHCF)
PVAR	RE	σ^2 in *Equation 2-48,* characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.
QO	RE	Q_o in *Equation 2-78,* the decay factor for the ovality effect.
QT	RE	Q_T in *Equation 2-80,* the decay factor for the torus effect.
R	RE	*R* in *Equation 2-68,* the radius where the stress is to be found.
RAINF1	RE	Real function which performs rainflow cycle counting, Miner's Rule damage accumulation, and calls GTLIFE to calculate the fatigue life.
RAND	DRE	Random number seed.
RANGEM(2, MAXREG)	RE	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.[6]
RB	RE	R_B (in.) in *Equation 2-74,* the elbow bend radius. (see **BNRD** in DCTHCF)
RI	RE	R_i (in.) in *Equation 2-68,* the duct inner radius.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
RM	RE	R_m (in.) in *Equations 2-74* through *2-77*, the mean duct radius.
RO	RE	R_o (in.) in *Equation 2-68*, the duct outer radius.
ROIB	RE	R_o (in.) in *Equation 2-68*, the duct outer radius calculated by using the wall thickness at the inner bend.
ROOB	RE	R_o (in.) in *Equation 2-68*, the duct outer radius calculated by using the wall thickness at the outer bend.
ROT	RE	*R* Over *T*, the value of the ratio *R/t*.
ROVERI	RE	*R* OVER *I*, the value of the ratio *R/I*.
RT(10)	RE	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}.
S(4, MAXM)	RE	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,*)** is the axial stress-time history $\sigma_1(t)$; **S(2,*)** is the hoop stress-time history $\sigma_2(t)$; **S(3,*)** is the radial stress-time history $\sigma_3(t)$; and **S(4,*)** is the shear stress-time history $\sigma_4(t)$.
SBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each life region of the S/N curve contained in array **NBND( )**.
SEFF(MAXM)	RE	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.
SIG(MAXREG)	RE	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.
SSTR	RE	$\lambda_{ST_{str}}$ in *Equation 2-81*, the randomly selected static stress analysis accuracy factor.
SSTRA	RE	Static stress analysis accuracy factor Uniform distribution lower bound.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
SSTRB	RE	Static stress analysis accuracy factor Uniform distribution upper bound.
STATIC(4)	RE	1-D array containing values of the static stresses σ_{STk} (psi), *Equation 2-82*. **STATIC(1)** is the axial stress σ_{ST1}; **STATIC(2)** is the hoop stress σ_{ST2}; **STATIC(3)** is the radial stress σ_{ST3}; and **STATIC(4)** is the shear stress σ_{ST4}.
STR(MAXDAT, MAXREG)	RE	2-D array containing stress points with stress ratio $= -1.0$, for the specific material S/N data set partitioned into life regions.
STRAMP(4, MAXLD)	RE	2-D array containing values of the amplitudes of the dynamic or time-varying stresses $\overline{\sigma_{Dki}}$ (psi), *Equation 2-82*. **STRAMP(1,I)** is $\overline{\sigma_{D1i}}$, the amplitude of the i^{th} axial stress; **STRAMP(2,I)** is $\overline{\sigma_{D2i}}$, the amplitude of the i^{th} hoop stress; **STRAMP(3,I)** is $\overline{\sigma_{D3i}}$, the amplitude of the i^{th} radial stress; and **STRAMP(4,I)** is $\overline{\sigma_{D4i}}$, the amplitude of the i^{th} shear stress.
STRHIS(MAXLD, MAXM)	RE	2-D array containing $\sigma_i(t)$, *Equation 2-82*, the reference time histories for the dynamic or time-varying load components.
SZERO	RE	Stress tensile test point, S_o (psi).[8]
T(MAXLD)	RE	1-D array containing M_x (in.-lbs) in *Equation 2-72*, the dynamic or time-varying torsional load components.
TEST	RE	Uniform(0, 1) random variate used to determine Beta distribution for W_{OFF}.
TIB	RE	tw_2 (in.) the wall thickness at the bend inner diameter at the weld. (see **WTID** in DCTHCF)
TLAM(MAXLD)	RE	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( )**.
TM	RE	t_m (in.) in *Equations 2-74* through *2-77*, the mean wall thickness at the weld.
TOB	RE	tw_1 (in.) the duct minimum wall thickness at the weld outer diameter. (see **MNWT** in DCTHCF)

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
TRSBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each region of the S/N curve contained in array **NBND( )** for each **PHI** draw consistent with the tensile point S_o.[8]
TRUNC	RE	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.
TSLAM	RE	M_x (in.-lbs) in *Equation 2-72*, the static torsional load component scaled by **SSTR** and **LAMST**.
TSTAT	RE	M_x (in.-lbs) in *Equation 2-72*, the static torsional load component.
TYPE(MAXLD)	INT	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.
V(2, MAXLD)	RE	2-D array containing the dynamic or time-varying shear load components. **V(1,*)** is V_y (lbs) in *Equation 2-72*, the shear load components along the y axis; and **V(2,*)** is V_z (lbs) in *Equation 2-72*, the shear load components along the z axis.
VARY	INT	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.
VLAM(2, MAXLD)	RE	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,*)** is V_y (lbs) in *Equation 2-72*, the shear load components along the y axis; and **VLAM(2,*)** is V_z (lbs) in *Equation 2-72*, the shear load components along the z axis.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
VSLAM(2)	RE	1-D array containing the static shear load components scaled by **SSTR** and **LAMST**. **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.
VSTAT(2)	RE	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.
WD	RE	W_D (in.) in *Equation 2-78,* the Weld Distance from elbow tangency line. (see **WEDS** in DCTHCF)
WEDS	RE	W_D (in.) in *Equation 2-78,* the WEld DiStance from elbow tangency line. (see **WD** in ELWELD)
WEOF	RE	W_{OFF} in *Equation 2-73,* the randomly selected WEld OFfset (%). (see **WOFF** in ELWELD)
WEOFA	RE	W_{OFF} lower bound of Beta distribution 1.
WEOFB	RE	W_{OFF} upper bound of Beta distribution 1.
WEOFC	RE	W_{OFF} lower bound of Beta distribution 2.
WEOFD	RE	W_{OFF} upper bound of Beta distribution 2.
WEOFE	RE	Decimal equivalent percentage weight occurring in Beta distribution 1 of the weld offset W_{OFF}.
WEOFHI	RE	Upper bound of the randomly selected Beta distribution for the weld offset W_{OFF}.
WEOFLO	RE	Lower bound of the randomly selected Beta distribution for the weld offset W_{OFF}.
WEOFR	RE	Randomly selected Beta distribution location parameter ρ for the weld offset W_{OFF}.
WEOFR1	RE	ρ Uniform distribution lower bound of Beta distribution 1 of W_{OFF}.
WEOFR2	RE	ρ Uniform distribution upper bound of Beta distribution 1 of W_{OFF}.
WEOFR3	RE	ρ Uniform distribution lower bound of Beta distribution 2 of W_{OFF}.

Table 7-2 List of Variables for Program DCTHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
WEOFR4	RE	ρ Uniform distribution upper bound of Beta distribution 2 of W_{OFF}.
WEOFT	RE	Randomly selected Beta distribution shape parameter θ for the weld offset W_{OFF}.
WEOFT1	RE	θ Uniform distribution lower bound of Beta distribution 1 of W_{OFF}.
WEOFT2	RE	θ Uniform distribution upper bound of Beta distribution 1 of W_{OFF}.
WEOFT3	RE	θ Uniform distribution lower bound of Beta distribution 2 of W_{OFF}.
WEOFT4	RE	θ Uniform distribution upper bound of Beta distribution 2 of W_{OFF}.
WOFF	RE	W_{OFF} in *Equation 2-73*, the randomly selected Weld OFFset (%). (see **WEOF** in DCTHCF)
WTID	RE	tw_2 (in.) the Wall Thickness at the bend Inner Diameter at the weld. (see **TIB** in ELWELD)
X1	RE	X_1 in *Equations 2-74* through *2-77*.
X2	RE	X_2 in *Equations 2-74* through *2-77*.
X3	RE	X_3 in *Equations 2-74* through *2-77*.
X4	RE	X_4 in *Equations 2-74* through *2-77*.
Z	RE	Z in *Equation 2-48*, the randomly selected process variation shift factor given by a Lognormal(0,**PVAR**) random variate.
ZROREG	INT	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**.[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 σ_* 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

DCTHCF Version 3.4

Program DCTHCF Listing Temporal Order, Uniform Distribution

Routine	Page

Program DCTHCF Listing Temporal Order, Truncated Normal Distribution

```
C***********************************************************************
C  PROGRAM DCTHCF CONTROLS THE FLOW OF LOGIC OF THE HIGH CYCLE
C  FATIGUE ELBOW WELDED DUCT PROBLEM
C  PROGRAMMER:  L. NEWLIN
C        DATE:  4SEP91
C     VERSION:  3.4 (MATCHR V8.5, PIPE V8.3, INSORT V2.1)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      PROGRAM DCTHCF

C  SUBPROGRAMS:  INFAGG, PAREST, PRYRV, BETAGN, NORMGN, WEIBGN, ELWELD,
C                TRMNAT, INSORT, SORTM, EXPTCD
C         FILES: 1:DCTHCD-OLD; 3:DCTHCO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
C                7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
C                11-26:user named-OLD;
C                NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

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, MID, MPROC, NBLIFE, NHYPER, NLIFE,
     &         NLIFET, NLOAD, NMED, NPTS(MAXREG), NRAN, 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, KGODR1,
     &         KGODR2, KGODT, KGODT1, KGODT2, KRATIO, KT(2,2), KWID,
     &         KWIDA, KWIDB, KWIDR, KWIDR1, KWIDR2, KWIDT, KWIDT1,
     &         KWIDT2, KWOD, KWODA, KWODB, KWODR, KWODR1, KWODR2,
     &         KWODT, KWODT1, KWODT2, LAMN, LAMNA, LAMNB, LAMNC, LAMND,
     &         LAMNK, LAMNMU, LAMNSG, LAMS, LAMSA, LAMSB, LAMSC, LAMSD,
     &         LAMSK, LAMSMU, LAMSSG, LAMST, LAMSTA, LAMSTB, LAMW,
     &         LAMWA, LAMWB, LIFE(MAXLIF), LIMPR, LNA(0:MAXREG), LNZ,
     &         LPHIM(0:MAXREG)

      REAL     M(2, MAXLD), MEDM(MAXREG), MLAM (2, MAXLD),
     &         MM(0:MAXREG), MNWT, MSLAM(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, RANGEM(2, MAXREG), RT(10), SBND(0:MAXREG),
     &         SIG(MAXREG), SSTR, SSTRA, SSTRB, STR(MAXDAT, MAXREG),
     &         STRHIS(MAXLD, MAXM), SZERO, T(MAXLD), TEST, TLAM(MAXLD),
     &         TRSBND(0:MAXREG), TRUNC, TSLAM, TSTAT, V(2, MAXLD),
     &         VLAM(2, MAXLD), VSLAM(2), VSTAT(2), WEDS, WEOF, WEOFA,
     &         WEOFB, WEOFC, WEOFD, WEOFE, WEOFHI, WEOFLO, WEOFR,
     &         WEOFR1, WEOFR2, WEOFR3, WEOFR4, WEOFT, WEOFT1, WEOFT2,
     &         WEOFT3, WEOFT4, WTID, Z

      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'
      WRITE(8,*) '                             (0 - NO, 1 - YES) =', MPROC

      IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF S/N VARIATION DESIRED'
         CALL TRMNAT
      ENDIF
      IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID RESPONSE TO NORMAL MEDIAN ',
     &              'CURVE QUESTION'
         CALL TRMNAT
      ENDIF

      IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF MATERIALS PROCESS ',
     &              'VARIATION DESIRED'
         CALL TRMNAT
      ENDIF

      READ(1,*) NBLIFE
```

```
      DO 15 J = 1, NBLIFE
         READ(1,*) BLFPER(J)
   15 CONTINUE

C ** READ DATA FROM DCTHCD

      READ(1,*) WEOFA, WEOFB, WEOFR1, WEOFR2, WEOFT1, WEOFT2,
     &          WEOFC, WEOFD, WEOFR3, WEOFR4, WEOFT3, WEOFT4,
     &          WEOFE,
     &          KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2,
     &          KWIDA, KWIDB, KWIDR1, KWIDR2, KWIDT1, KWIDT2,
     &          KGODA, KGODB, KGODR1, KGODR2, KGODT1, KGODT2,
     &          LAMNA, LAMNB, LAMNC,  LAMND,
     &          LAMSA, LAMSB, LAMSC,  LAMSD,
     &          LAMSTA,LAMSTB,DSTRA,  DSTRB,  SSTRA,  SSTRB,
     &          CLZA,  CLZB,  CLYA,   CLYB,
     &          CCZA,  CCZB,  CCYA,   CCYB,
     &          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(1,*) LDNAME(I), TYPE(I), P(I), T(I), M(1,I), M(2,I),
     &             V(1,I), V(2,I)
         IF ((TYPE(I) .LT. 1) .OR. (TYPE(I) .GT. 2)) THEN
            WRITE(8,*) 'ERROR:  LOAD INCORRECTLY TYPED'
            CALL TRMNAT
         ENDIF
   20 CONTINUE
      READ(1,*) KGID, KT(2,1), KT(2,2), LIMPR, BNRD, WEDS, IDWE,
     &          MNWT, WTID, 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, WEOFR1, WEOFR2, WEOFT1, WEOFT2,
     &             WEOFC, WEOFD, WEOFR3, WEOFR4, WEOFT3, WEOFT4,
     &             WEOFE
      WRITE(3,902) KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2
      WRITE(3,903) KWIDA, KWIDB, KWIDR1, KWIDR2, KWIDT1, KWIDT2
      WRITE(3,904) KGODA, KGODB, KGODR1, KGODR2, KGODT1, KGODT2
      WRITE(3,905) LAMNA, LAMNB, LAMNC,  LAMND
      WRITE(3,906) LAMSA, LAMSB, LAMSC,  LAMSD
      WRITE(3,907) LAMSTA,LAMSTB,DSTRA,  DSTRB,  SSTRA,  SSTRB,
     &             CLZA,  CLZB,  CLYA,   CLYB,
     &             CCZA,  CCZB,  CCYA,   CCYB,
     &             OVALA, OVALB, LAMWA,  LAMWB,EXP(GAMA),EXP(GAMB)
      WRITE(3,910) PSTAT, TSTAT, MSTAT(1), MSTAT(2), VSTAT(1), VSTAT(2)
      DO 25 I = 1, NLOAD
         WRITE(3,911) LDNAME(I), P(I), T(I), M(1,I), M(2,I), V(1,I),
     &                V(2,I)
   25 CONTINUE
      WRITE(3,920) KGID, KT(2,1), KT(2,2), LIMPR, BNRD, WEDS, IDWE,
     &             MNWT, WTID, EMOD, LOCAT, ANGLE, PERIOD, TRUNC,
     &             NLOAD, NRAN

      IF (NRAN .GT. MAXM) THEN
         WRITE(8,*) 'ERROR:  STRESS-TIME HISTORY TOO LARGE'
         CALL TRMNAT
      ENDIF

      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 TRMNAT
```

```
         ENDIF
   30 CONTINUE

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

      CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZROREG, NUMREG,
     &             NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)

      ZROREG = 1
      SZERO = 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, MAXBLF
         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 PRYRV TO OBTAIN RHO,THETA PAIRS FOR INNER LOOP CALCULATIONS

         CALL RANDOM (TEST, RAND)
         IF (TEST .LE. WEOFE) THEN
            CALL PRYRV (RAND,WEOFR1,WEOFR2,WEOFT1, WEOFT2, WEOFR, WEOFT)
            WEOFLO = WEOFA
            WEOFHI = WEOFB
         ELSE
            CALL PRYRV (RAND,WEOFR3,WEOFR4,WEOFT3, WEOFT4, WEOFR, WEOFT)
            WEOFLO = WEOFC
            WEOFHI = WEOFD
         ENDIF
         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'TEST =', TEST, ' WEOFE =', WEOFE
            WRITE(8,*) 'WEOFLO =', WEOFLO, ' WEOFHI =', WEOFHI
         ENDIF

         CALL PRYRV (RAND, KWIDR1, KWIDR2, KWIDT1, KWIDT2, KWIDR, KWIDT)
         CALL PRYRV (RAND, KWODR1, KWODR2, KWODT1, KWODT2, KWODR, KWODT)
         CALL PRYRV (RAND, KGODR1, KGODR2, KGODT1, KGODT2, KGODR, KGODT)
         CALL PRYRV (RAND, LAMNA, LAMNB,  LAMSA,  LAMSB,  LAMNK, LAMSK)
         LAMNMU = LAMND / (1.0 + LAMNK * LAMNC)
         LAMNSG = LAMNC / (1.0 + LAMNK * LAMNC)
         LAMSMU = LAMSD / (1.0 + LAMSK * LAMSC)
         LAMSSG = LAMSC / (1.0 + LAMSK * LAMSC)

         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'LAMNK = ', LAMNK, ' LAMNMU = ', LAMNMU,
     &                 ' LAMNSG = ', LAMNSG
            WRITE(8,*) 'LAMSK = ', LAMSK, ' LAMSMU = ', LAMSMU,
     &                 ' LAMSSG = ', LAMSSG
         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)

C ** OBTAIN MATERIALS PROCESS VARIATION PARAMETERS IF DESIRED
```

```
      CALL NORMGN (RAND, 0.0, PSIG, LNZ)

      IF (MPROC .EQ. 1) THEN
         Z = EXP (LNZ)
      ELSE
         KRATIO = 1.0
         Z = 1.0
         LNZ = 0.0
      ENDIF

      MCOUNT = MCOUNT + 1
      DO 175 L = 1, NUMREG
         ALLM(MCOUNT, L) = MM(L)
175   CONTINUE

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

      DO 200 I = 1, NLIFE

C ** INITILIZE S/N CURVE PARAMETERS

         DO 225 L = 0, MAXREG
            LNA(L) = 0.0
            LPHIM(L) = 0.0
            TRSBND(L) = 0.0
225      CONTINUE

C ** SELECT DRIVERS FOR CALCULATING LIFE

         CALL BETAGN (RAND, WEOFR, WEOFT, WEOFLO, WEOFHI, WEOF)
         CALL BETAGN (RAND, KWIDR, KWIDT, KWIDA, KWIDB, KWID)
         CALL BETAGN (RAND, KWODR, KWODT, KWODA, KWODB, KWOD)
         CALL BETAGN (RAND, KGODR, KGODT, KGODA, KGODB, KGOD)
         CALL NORMGN (RAND, LAMNMU, LAMNSG, LAMN)
         CALL NORMGN (RAND, LAMSMU, LAMSSG, LAMS)
         CALL PRYRV  (RAND, OVALA, OVALB, LAMSTA,LAMSTB, OVAL,LAMST)
         CALL PRYRV  (RAND, DSTRA, DSTRB, SSTRA, SSTRB,  DSTR, SSTR)
         CALL PRYRV  (RAND, CLZA,  CLZB,  CLYA,  CLYB,   CLZ,  CLY)
         CALL PRYRV  (RAND, CCZA,  CCZB,  CCYA,  CCYB,   CCZ,  CCY)
         CALL PRYRV  (RAND, LAMWA, LAMWB, GAMA,  GAMB,   LAMW, GAM)
         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,' KWOD =',KWOD
            WRITE(8,*) 'KGOD =',KGOD,' LAMN =',LAMN,' LAMS =',LAMS
            WRITE(8,*) 'LAMST =',LAMST,' DSTR =',DSTR,' SSTR =',SSTR
            WRITE(8,*) 'CLZ = ', CLZ, ' CLY = ', CLY
            WRITE(8,*) 'CCZ = ', CCZ, ' CCY = ', CCY
            WRITE(8,*) 'OVAL =',OVAL,' LAMW =',LAMW
            WRITE(8,*) 'GAM =',GAM,' 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
```

```
                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
  235     CONTINUE

          IF (IOUT .EQ. 15) THEN
             WRITE(8,*) 'STATIC LOADS'
             WRITE(8,*) '  P = ', PSLAM, ' T = ', TSLAM,
     &                  ' M2 = ', 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,*) '  P = ', PLAM(II), ' T = ', TLAM(II),
     &                     ' M2 = ', MLAM(1,II)
                WRITE(8,*) '  M3 = ', MLAM(2,II), ' V2 = ',
     &                VLAM(1,II), ' V3 = ', VLAM(2,II)
  240        CONTINUE
          ENDIF

C ** CALCULATE AXIAL Kt's

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

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

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 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, IDWE, WEOF, LAMW, BNRD,
     &                    WEDS, FK, RT, CCY, CCZ, CLY, CLZ, OVAL,
     &                    LOCAT, MM, LNA, LPHIM, KRATIO, LNZ, TRSBND,
     &                    SZERO, ZROREG, NUMREG, STRHIS, NRAN,
     &                    PERIOD, TRUNC, ANGLE, CULPRT)

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

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

  200    CONTINUE

  150 CONTINUE

      IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES

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

C ** PRINT EMPIRICAL BLIVES
```

```
         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)
  400    CONTINUE

         CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGK1, BZERO)

      ENDIF


C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO DCTHCO

  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &       'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &       'NASA Contract NAS7-918 is acknowledged.',////,
     &       33X,'INPUT DATA',
     &       ///,14X,'DRIVERS',25X,'PARAMETER DISTRIBUTIONS',
     &       //,48X,'RHO',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,
     &       'Be(',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,')')

  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,
     &       /,20X,'STRAIN GAGE FACTOR: ',F9.7)

  906 FORMAT(/,2X,'LAMBDA SINE',7X,'k: U(',F7.5,',',F8.5,')',
     &       /,20X,'COEFFICIENT OF VARIATION: ',F5.3,
     &       /,20X,'STRAIN GAGE FACTOR: ',F9.7)

  907 FORMAT(//,2X,'LAMBDA STATIC',23X,'U(',F8.5,',',F9.5,')',
     &       //,2X,'DYNAMIC STRESS ANALYSIS',13X,'U(',F8.5,',',F9.5,')',
     &       //,2X,'STATIC STRESS ANALYSIS',14X,'U(',F8.5,',',F9.5,')',
     &       //,2X,'STRESS CARRYOVER FACTORS',
     &       /,5X,'IN-PLANE AXIAL',19X,'U(',F8.5,',',F9.5,')',
     &       /,5X,'OUT-OF-PLANE AXIAL',15X,'U(',F8.5,',',F9.5,')',
     &       /,5X,'IN-PLANE CIRCUMFERENTIAL',9X,'U(',F8.5,',',F9.5,')',
     &       /,5X,'OUT-OF-PLANE CIRCUMFERENTIAL',
     &       5X,'U(',F8.5,',',F9.5,')',//,
     &       2X,'OVALITY ANALYSIS FACTOR',13X,'U(',F8.5,',',F9.5,')',
     &       //,2X,'LAMBDA KOFF',25X,'U(',F8.5,',',F9.5,')',
     &       //,2X,'DAMAGE MODEL ACCURACY',15X,'U(ln',F8.5,
     &       ', ln',F8.5,')')
```

```
910 FORMAT(////,28X,'LOADS INPUT',///,5X,'P LOADS',5X,'T LOADS',5X,
     &        'M2 LOADS',4X,'M3 LOADS',4X,'V2 LOADS',4X,'V3 LOADS',
     &        /,6X,'(LBS)',5X,'(IN.-LBS)',4X,'(IN.-LBS)',
     &        3X,'(IN.-LBS)',5X,'(LBS)',6X,'(LBS)',
     &        //,2X,'STATIC',
     &        /,2X,F10.4,2X,F10.4,2X,F10.4,2X,F10.4,2X,F10.4,2X,F10.4)

911 FORMAT(2X,A6,/,2X,F10.4,2X,F10.4,2X,F10.4,2X,F10.4,2X,F10.4,
     &        2X,F10.4)

920 FORMAT(////,20X,'GEOMETRIC AND OTHER INPUT',
     &        ///,2X,'K GEOM (ID)',41X,F4.2,
     &        //,2X,'K HOOP (OD)',41X,F4.2,
     &        //,2X,'K HOOP (ID)',41X,F4.2,
     &        //,2X,'LIMIT PRESSURE, PSI',29X,F6.0,
     &        //,2X,'BEND RADIUS, IN.',35X,F5.2,//,
     &        2X,'WELD DISTANCE FROM ELBOW TANGENCY LINE, IN.',8X,F6.3,
     &        //,2X,'DUCT INSIDE DIAMETER, IN.',26X,F5.2,
     &        //,2X,'MINIMUM WALL THICKNESS, IN.',25X,F6.4,
     &        //,2X,'WALL THICKNESS AT BEND (ID), IN.',20X,F6.4,
     &        //,2X,'ELASTIC MODULUS, PSI',32X, E9.3,
     &        //,2X,'ANALYSIS LOCATION',35X,I1,
     &        //,2X,'ANGLE PHI (DEG)',34X,F6.1,
     &        //,2X,'STRESS-TIME HISTORY PERIOD, SEC',20X,F5.2,
     &        //,2X,'STRESS-TIME HISTORY NOISE FILTER, PSI',11X,F7.1,
     &        //,2X,'NUMBER OF TIME-VARYING LOADS',23X,I2,
     &        //,2X,'NUMBER OF POINTS IN HISTORIES',19X,I5)

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

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


    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 3................................S/N VARIATION
C 1................................NORMAL MEDIAN REQUIRED
C 0................................MATERIALS PROC. VAR. NOT REQUIRED
C 3................................NUMBER OF BLIVES TO BE PROVIDED
C 0.0001...........................B.01 LIFE
C 0.001............................B.1 LIFE
C 0.01.............................B1 LIFE
C 0.00  0.20  0.50  0.8   0.5   5...WELD OFFSET      (A,B) (R1,R2) (T1,T2)
C 0.20  0.50  0.00  0.2   5.   10.................. (C,D) (R3,R4) (T3,T4)
C 0.90...............................................TEST FOR HYPER-DIST.
C 1.20  1.80  0.30  0.70  0.5  10...K WELD (OD)     (A,B) (R1,R2) (T1,T2)
C 1.04  1.43  0.30  0.70  0.5  10...K WELD (ID)     (A,B) (R1,R2) (T1,T2)
C 1.20  1.34  0.30  0.70  0.5  10...K GEOM (OD)     (A,B) (R1,R2) (T1,T2)
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 0.90  1.10........................LAMBDA STATIC
C 0.80  1.20........................DYNAMIC STRESS ANALYSIS ACCURACY FACTOR
C 0.90  1.10........................STATIC STRESS ANALYSIS ACCURACY FACTOR
C
C                                   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
C
C 0.85  1.15........................OVALITY ANALYSIS ACCURACY FACTOR
```

```
C 0.95  1.05.........................LAMBDA K WELD OFFSET ACCURACY FACTOR
C -1.38629  0.95166..................MODEL ACCURACY
C
C 16.................................NUMBER OF LOADS
C
C PSTAT     TSTAT     MSTAT(1)  MSTAT(2)  VSTAT(1)  VSTAT(2).....STATIC
C 8130.00  20900.00  42010.00  42010.00  3805.00   3805.00......LOADS
C
C FILE   TYPE  P()     T()      M(1,*)   M(2,*)   V(1,*)   V(2,*).....LOADS
C
C 'NBP'    1 237.675  0.00     0.00     0.00     0.00     0.00......RANDOM1
C 'NBT'    1 0.00     103.41   0.00     0.00     0.00     0.00......RANDOM2
C 'NBM2'   1 0.00     0.00     181.21   0.00     0.00     0.00......RANDOM3
C 'NBM3'   1 0.00     0.00     0.00     626.175  0.00     0.00......RANDOM4
C 'NBV2'   1 0.00     0.00     0.00     0.00     147.49   0.00......RANDOM5
C 'NBV3'   1 0.00     0.00     0.00     0.00     0.00     34.075....RANDOM6
C 'SIN1'   2 4.889461 1.88731  3.002265 8.618995 13.91015 0.829459..SINE1
C 'SIN2'   2 17.2329  12.6415  0.182346 38.4677  54.89455 2.90558...SINE2
C 'SIN3'   2 3.117695 2.764815 4.45821  29.7981  4.905385 0.691592..SINE3
C 'SIN4'   2 1.107417 0.856604 1.17435  3.663675 1.350412 0.414575..SINE4
C 'SIN5'   2 10.23887 11.81905 137.38   28.5843  6.0813   13.24209..SINE5
C 'SIN6'   2 2.151205 1.62707  0.430078 5.991475 7.077595 0.395232..SINE6
C 'SIN7'   2 4.13738  8.509805 5.235795 71.06695 15.61234 1.242015..SINE7
C 'SIN8'   2 9.1491   0.904076 5.953345 0.934805 5.04324  0.843876..SINE8
C 'SIN9'   2 32.10965 0.084774 1.236315 23.9187  16.7327  0.162597..SINE9
C 'SIN10'  2 79.7046  7.056975 2.48936  35.04565 36.66045 4.07806...SINE10
C
C 1.0................................K GEOM (ID)
C 1.0................................K HOOP (OD)
C 1.0................................K HOOP (ID)
C 4902...............................LIMIT PRESSURE, PSI
C 6.0................................BEND RADIUS, IN
C .112...............................WELD DIST. FROM ELBOW TANG. LINE, IN
C 4.0................................DUCT INSIDE DIAMETER, IN
C 0.113..............................MINIMUM WALL THICKNESS, IN
C 0.134..............................WALL THICKNESS AT BEND (ID), IN
C 3.01E+07...........................ELASTIC MODULUS
C 1..................................LOCATION OF INTEREST
C 90.0...............................ANGLE OF INTEREST (DEG)
C 1.0................................STRESS-TIME HISTORY PERIOD (SEC)
C 500.0..............................STRESS-TIME HISTORY NOISE FILTER (PSI)
C 1000...............................NUMBER OF POINTS IN STRESS-TIME HISTORY
C 0.615     2.00.....................FK(1), RT(1)
C 0.693     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 0.873    12.50.....................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   200.00.....................FK(10), RT(10)
C '-320 HOURGLASS + STRAIGHT'........MATERIAL DESCRIPTION
C 178600.   220400.   1   20.........YIELD & ULTIMATE STRENGTHES, NDIV, NPTS
C 20   0.05   1......................# PTS IN DIV, STRESS RATIO, REGION
C 150000.    65000...................S(1) N(1)
C 140000.   261000...................S(2) N(2)
C 120000.   265000...................S(3) N(3)          RAW
C 160000.   377000...................S(4) N(4)         STRESS
C 130000.   694000...................S(5) N(5)          LIFE
C 110000.  2175000...................S(6) N(6)         (S/N)
C 100000.  4198000...................S(7) N(7)          DATA
C 105000.  5053000...................S(8) N(8)         POINTS
C  92000.  9210000...................S(9) N(9)           FOR
C  95000.  9667000...................S(10) N(10)        THE
C 150000.   418000...................S(11) N(11)     SPECIFIC
C 140000.   732000...................S(12) N(12)     MATERIAL
C 130000.   740000...................S(13) N(13)
C 120000.   859000...................S(14) N(14)
C 110000.  1181000...................S(15) N(15)
C 100000.  4020000...................S(16) N(16)
C  92000.  5917000...................S(17) N(17)
C  94000.  6522000...................S(18) N(18)
C  90000.  6891000...................S(19) N(19)
C  86000.  4460000...................S(20) N(20)
```

```
C 0.0...............................NO VALUE OF So SUPPLIED
C 1  0..............................NUMBER OF REGIONS:W/DATA W/O DATA
C 1.0E+36...........................LIFE BOUNDARIES REGION 1
C 0.00..............................CONSTRAINT ON COEFF. OF VARIATION
C 2    3.596    5.874...............2 PTS IN RANGE, LOWER BOUND, UPPER BOUND
C 0.0    0.0    0.0.................NORMAL DIST. PRIORS: DELTA, Mo, SIGMA2
C
C*************************************************************************
C                           LIST OF VARIABLES
C*************************************************************************
C
C ALLM()          2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C ANGLE           ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --
C                   GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
C BIGK()          1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                   EACH REGION
C BIGK1           EQUAL TO BIGK(1) -- DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C                   EXPCTD
C BLFPER()        1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS()        1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPIRICAL BLIVES
C BNRD            BEND RADIUS, IN.
C BZERO           VALUE OF BETAo RANDOMLY SELECTED FROM BETA() INTERVAL
C CCY             SELECTED OUT-OF-PLANE CIRCUMFERENTIAL STRESS CARRYOVER FACTOR
C CCYA            CCY LOWER BOUND
C CCYB            CCY UPPER BOUND
C CCZ             SELECTED IN-PLANE CIRCUMFERENTIAL STRESS CARRYOVER FACTOR
C CCZA            CCZ LOWER BOUND
C CCZB            CCZ UPPER BOUND
C CLY             SELECTED OUT-OF-PLANE AXIAL STRESS CARRYOVER FACTOR
C CLYA            CLY LOWER BOUND
C CLYB            CLY UPPER BOUND
C CLZ             SELECTED IN-PLANE AXIAL STRESS CARRYOVER FACTOR
C CLZA            CLZ LOWER BOUND
C CLZB            CLZ UPPER BOUND
C CULPRT          LOCATION CAUSING FAILURE
C DSTR            SELECTED DYNAMIC STRESS ANALYSIS ACCURACY FACTOR
C DSTRA           DYNAMIC STRESS ANALYSIS ACCURACY FACTOR LOWER BOUND
C DSTRB           DYNAMIC STRESS ANALYSIS ACCURACY FACTOR UPPER BOUND
C DUM             DUMMY VARIABLE
C ELWELD          REAL FUNCTION WHICH CALCULATES THE DUCT LIFE (IN SECONDS)
C                   AT A WELD NEAR AN ELBOW
C EMOD            MATERIAL ELASTIC MODULUS
C FIFTY           EQUAL TO .5 -- USED TO ACCESS 50% POINT IN LIFE()
C FILNUM()        1-D ARRAY CONTAINING UNIT NUMBERS FOR STRESS-TIME HISTORIES
C                   FILES
C FK()            1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C                   CONCENTRATION DUE TO WELD ECCENTRICITY
C FTEST           File TEST -- USED TO TEST EXISTENCE OF FILE
C FTU             MATERIAL ULTIMATE STRENGTH
C FTY             MATERIAL YIELD STRENGTH
C GAM             SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, LAMBDAdam
C GAMA            GAM LOWER BOUND
C GAMB            GAM UPPER BOUND
C I               CONTROLS DO LOOP FOR EACH LIFE CALCULATION
C IDWE            WELD INSIDE DIAMETER, IN
C II              CONTROLS DO LOOP FOR RANDOM AND SUPERIMPOSED SINE LOADS
C IOUT            CONTROLS DUMP TO SCREEN/PRINTER
C J               CONTROLS DO LOOP FOR EACH B-LIFE
C K               CONTROLS DO LOOP FOR EACH HYPER-PARAMETER SET
C KGID            K GEOM (ID)
C KGOD            SELECTED K GEOM (OD)
C KGODA           K GEOM (OD) LOWER BOUND
C KGODB           K GEOM (OD) UPPER BOUND
C KGODR           SELECTED RHO FOR K GEOM (OD)
C KGODR1          K GEOM (OD) - RHO LOWER BOUND
C KGODR2          K GEOM (OD) - RHO UPPER BOUND
C KGODT           SELECTED THETA FOR K GEOM (OD)
C KGODT1          K GEOM (OD) - THETA LOWER BOUND
C KGODT2          K GEOM (OD) - THETA UPPER BOUND
C KRATIO          RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C KT()            FATIGUE STRESS CONCENTRATION FACTORS -- KT(1,1) =
C                   Kt AXIAL (OD) (= KGOD * KWOD); KT(1,2) = Kt AXIAL (ID)
C                   (= KGID * KWID); KT(2,1) = Kt HOOP (OD); KT(2,2) =
C                   Kt HOOP (ID)
```

```
C KWID          SELECTED K WELD (ID)
C KWIDA         K WELD (ID) LOWER BOUND
C KWIDB         K WELD (ID) UPPER BOUND
C KWIDR         SELECTED RHO FOR K WELD (ID)
C KWIDR1        K WELD (ID) - RHO LOWER BOUND
C KWIDR2        K WELD (ID) - RHO UPPER BOUND
C KWIDT         SELECTED THETA FOR K WELD (ID)
C KWIDT1        K WELD (ID) - THETA LOWER BOUND
C KWIDT2        K WELD (ID) - THETA UPPER BOUND
C KWOD          SELECTED K WELD (OD)
C KWODA         K WELD (OD) LOWER BOUND
C KWODB         K WELD (OD) UPPER BOUND
C KWODR         SELECTED RHO FOR K WELD (OD)
C KWODR1        K WELD (OD) - RHO LOWER BOUND
C KWODR2        K WELD (OD) - RHO UPPER BOUND
C KWODT         SELECTED THETA FOR K WELD (OD)
C KWODT1        K WELD (OD) - THETA LOWER BOUND
C KWODT2        K WELD (OD) - THETA UPPER BOUND
C L             CONTROLS DO LOOP FOR EACH REGION
C LAMN          SELECTED LAMBDA FOR ONE SIGMA NARROW-BAND RANDOM LOADS
C LAMNA         LAMBDA FOR NARROW-BAND RANDOM LOADS -- LOWER BOUND OF k
C LAMNB         LAMBDA FOR NARROW-BAND RANDOM LOADS -- UPPER BOUND OF k
C LAMNC         LAMBDA FOR NARROW-BAND RANDOM LOADS COEFFICIENT OF VARIATION
C LAMND         NARROW-BAND RANDOM LOADS STRAIN GAGE ACCURACY FACTOR
C LAMNK         LAMBDA FOR NARROW-BAND RANDOM LOADS k -- INDICATES VARIATION
C                 DUE TO SAMPLE SIZE
C LAMNMU        MEAN OF LAMBDA FOR NARROW-BAND RANDOM LOADS (MU, NORMAL
C                 DISTRIBUTION)
C LAMNSG        STANDARD DEVIATION OF LAMBDA FOR NARROW-BAND RANDOM LOADS
C                 (SIGMA, NORMAL DISTRIBUTION)
C LAMS          SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS
C LAMSA         LAMBDA FOR SUPERIMPOSED SINE LOADS -- LOWER BOUND OF k
C LAMSB         LAMBDA FOR SUPERIMPOSED SINE LOADS -- UPPER BOUND OF k
C LAMSC         LAMBDA FOR SUPERIMPOSED SINE LOADS COEFFICIENT OF VARIATION
C LAMSD         SUPERIMPOSED SINE LOADS STRAIN GAGE ACCURACY FACTOR
C LAMSK         LAMBDA FOR SUPERIMPOSED SINE LOADS k -- INDICATES VARIATION
C                 DUE TO SAMPLE SIZE
C LAMSMU        MEAN OF LAMBDA FOR SUPERIMPOSED SINE LOADS (MU, NORMAL
C                 DISTRIBUTION)
C LAMSSG        STANDARD DEVIATION OF LAMBDA FOR SUPERIMPOSED SINE LOADS
C                 (SIGMA, NORMAL DISTRIBUTION)
C LAMST         SELECTED LAMBDA FOR STATIC LOADS
C LAMSTA        LAMBDA STATIC LOADS LOWER BOUND
C LAMSTB        LAMBDA STATIC LOADS UPPER BOUND
C LAMW          SELECTED ACCURACY FACTOR FOR WELD ECCENTRICITY STRESS
C                 CONCENTRATION FACTOR, Koff
C LAMWA         LAMW LOWER BOUND
C LAMWB         LAMW UPPER BOUND
C LDNAME()      1-D ARRAY CONTAINING LoaD NAMEs FOR THE TIME-VARYING LOADS
C LIFE()        1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE
C                 PFM -- SORTED VALUES OF THE LEFT-HAND TAIL
C LIMPR         LIMIT PRESSURE, PSI (INTERNAL PRESSURE)
C LNA()         1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNZ           NORMAL(0,PVAR) GENERATED RANDOM VARIABLE
C LOCAT         LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE
C                 DUCT AND 2 IS THE INTERIOR SURFACE OF THE DUCT
C LPHIM()       1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C M()           2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                 ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C MAXBLF        MAXIMUM NUMBER OF BLIVES TO BE CALCULATED
C MAXDAT        MAXIMUM NUMBER OF POINTS PER DATA SET PER REGION ALLOWED
C MAXLD         MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C MAXLIF        MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA,
C                 ALPHA CALCULATION
C MAXM          MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C MAXMM         MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
C MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C MCOUNT        NUMBER OF M's TO BE USED TO CALCULATE MEDIAN S/N CURVE
C MEDM()        1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C MID           POINTER TO THE MEDIAN M VALUES -- EQUAL TO HALF OF MCOUNT
C MLAM()        2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS SCALED
C                 BY DSTR AND LAMS OR LAMN AS APPROPRIATE (INDICATED BY
C                 TYPE()) -- MLAM(1,*) ARE THE M2 LOADS; MLAM(2,*) ARE
C                 THE M3 LOADS
```

```
C MM()          1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MNWT          MINIMUM WALL THICKNESS AT WELD, IN
C MPROC         Materials PROCess variation -- CONTROLS MATERIALS PROCESS
C                 VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C MSLAM()       1-D ARRAY CONTAINING THE STATIC MOMENT LOADS SCALED BY LAMST
C                 AND SSTR -- MSLAM(1) IS THE M2 LOAD; MSLAM(2) IS THE M3 LOAD
C MSTAT()       1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- MSTAT(1) IS
C                 THE M2 LOAD; MSTAT(2) IS THE M3 LOAD
C MU()          1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION MEAN FOR EACH REGION
C NBLIFE        NUMBER OF BLIVES TO BE CALCULATED
C NBND()        1-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE
C                 REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE)
C                 MATERIAL S/N DATA SET
C NEWLIF        LIFE VALUE RETURNED FROM CALL TO ELWELD
C NF()          2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL
C                 S/N DATA SET BROKEN INTO LIFE REGIONS
C NHYPER        SIZE OF OUTER LOOP
C NLIFE         SIZE OF INNER LOOP
C NLIFET        TOTAL NUMBER OF LIVES CALCULATED BY PFM
C NLOAD         NUMBER OF TIME-VARYING LOADS
C NMED          CONTROLS MEDIAN CALCULATION FOR THE NORMAL DISTRIBUTION CASE
C                 -- 0 - NO MEDIAN CALCULATION;  1 - MEDIAN CALCULATION DESIRED
C NPTS()        1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION
C                 FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C NRAN          NUMBER OF POINTS IN STRESS-TIME HISTORY (Number of RANdom
C                 points)
C NUMREG        NUMBER OF REGIONS OF INTEREST
C OVAL          SELECTED OVALITY ANALYSIS ACCURACY FACTOR
C OVALA         OVALITY ANALYSIS ACCURACY FACTOR LOWER BOUND
C OVALB         OVALITY ANALYSIS ACCURACY FACTOR UPPER BOUND
C P()           1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C PERIOD        LENGTH OF TIME IN SECONDS OF RANDOM STRESS-TIME HISTORY
C PHI           WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C PI            CONSTANT EQUAL TO 3.1415926536
C PLAM()        1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS SCALED BY
C                 DSTR AND LAMN OR LAMS AS APPROPRIATE (INDICATED BY TYPE())
C PSIG          EQUAL TO SQRT(PVAR) -- MATERIALS PROCESS STANDARD DEVIATION
C PSLAM         STATIC AXIAL LOAD SCALED BY LAMST AND SSTR
C PSTAT         STATIC AXIAL LOAD
C PVAR          MATERIALS PROCESS VARIATION
C RAND          RANDOM NUMBER SEED
C RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C RT()          1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C                 CONCENTRATION DUE TO WELD ECCENTRICITY
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()
C SIG()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C SSTR          SELECTED STATIC STRESS ANALYSIS ACCURACY FACTOR
C SSTRA         STATIC STRESS ANALYSIS ACCURACY FACTOR LOWER BOUND
C SSTRB         STATIC STRESS ANALYSIS ACCURACY FACTOR UPPER BOUND
C STR()         2-D ARRAY CONTAINING STRESS POINTS (STRESS RATIO = -1.0)
C                 FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO
C                 LIFE REGIONS
C STRHIS()      2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING
C                 STRESS-TIME HISTORIES
C SZERO         STRESS TENSILE TEST POINT, So
C T()           1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C TEST          UNIFORM(0,1) RANDOM VARIATE USED TO DETERMINE HYPER-
C                 DISTRIBUTION TO SELECT FROM FOR DUAL-BETA DISTRIBUTIONS
C TLAM()        1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS SCALED
C                 BY DSTR AND LAMN OR LAMS AS APPROPRIATE (INDICATED BY
C                 TYPE())
C TRSBND()      1-D ARRAY CONTAINING VALUES OF PHI * KRATIO * Z * SBND FOR
C                 EACH REGION CALCULATED FOR EACH TRIAL
C TRUNC         VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
C TSLAM         STATIC TORQUE LOAD SCALED BY LAMST AND SSTR
C TSTAT         STATIC TORQUE LOAD
C TYPE()        1-D ARRAY CONTAINING THE TYPE OF TIME-VARYING LOAD, USED FOR
C                 LOAD FACTORS -- TYPE(*) = 1 INDICATES NARROW-BAND RANDOM;
C                 TYPE(*) = 2 INDICATES SUPERIMPOSED SINUSOID
```

```
C V()           2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                 ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C VARY          CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
C                 1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;  3 -
C                 TRUNCATED NORMAL VARIATION
C VLAM()        2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS SCALED
C                 BY DSTR AND LAMN OR LAMS AS APPROPRIATE (INDICATED BY
C                 TYPE()) -- VLAM(1,*) ARE THE V2 LOADS; VLAM(2,*) ARE THE
C                 V3 LOADS
C VSLAM()       1-D ARRAY CONTAINING THE STATIC SHEAR LOADS SCALED BY LAMST
C                 AND SSTR -- VSLAM(1) IS THE V2 LOAD; VSLAM(2) IS THE V3 LOAD
C VSTAT()       1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- VSTAT(1) IS
C                 THE V2 LOAD; VSTAT(2) IS THE V3 LOAD
C WEDS          WELD DISTANCE FROM ELBOW TANGENCY LINE, IN
C WEOF          SELECTED WELD OFFSET (%)
C WEOFA         WELD OFFSET LOWER BOUND - HYPER-DISTRIBUTION 1
C WEOFB         WELD OFFSET UPPER BOUND - HYPER-DISTRIBUTION 1
C WEOFC         WELD OFFSET LOWER BOUND - HYPER-DISTRIBUTION 2
C WEOFD         WELD OFFSET UPPER BOUND - HYPER-DISTRIBUTION 2
C WEOFE         PERCENTAGE OCCURRING IN HYPER-DISTRIBUTION 1
C WEOFHI        SELECTED WELD OFFSET UPPER BOUND
C WEOFLO        SELECTED WELD OFFSET LOWER BOUND
C WEOFR         SELECTED RHO FOR WELD OFFSET
C WEOFR1        WELD OFFSET - RHO LOWER BOUND - HYPER-DISTRIBUTION 1
C WEOFR2        WELD OFFSET - RHO UPPER BOUND - HYPER-DISTRIBUTION 1
C WEOFR3        WELD OFFSET - RHO LOWER BOUND - HYPER-DISTRIBUTION 2
C WEOFR4        WELD OFFSET - RHO UPPER BOUND - HYPER-DISTRIBUTION 2
C WEOFT         SELECTED THETA FOR WELD OFFSET
C WEOFT1        WELD OFFSET - THETA LOWER BOUND - HYPER-DISTRIBUTION 1
C WEOFT2        WELD OFFSET - THETA UPPER BOUND - HYPER-DISTRIBUTION 1
C WEOFT3        WELD OFFSET - THETA LOWER BOUND - HYPER-DISTRIBUTION 2
C WEOFT4        WELD OFFSET - THETA UPPER BOUND - HYPER-DISTRIBUTION 2
C WTID          SELECTED WALL THICKNESS AT BEND (ID) AT WELD, IN
C Z             LOG-NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C ZROREG        ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE -- 0 ZERO REGION EXISTS, 1 - NO ZERO
C                 REGION


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

C  SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20JUN90
C     VERSION:  2.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C   INPUTS:  NEWLIF, LIFE, NLIFET
C  OUTPUTS:  LIFE

C     IMPLICIT NONE

      INTEGER  MAXLIF

      PARAMETER (MAXLIF = 10000)

      COMMON   IOUT

      INTEGER  I, IOUT, NLIFET, NUM, PLACE

      REAL     LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

C                 LIST OF VARIABLES
C
C  I        CONTROLS DO LOOP FOR INSERTION
C  IOUT     OUTPUT DUMP CONTROLLER
```

```
C  LIFE()     1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
C               PFM TO BE SORTED
C  MAXLIF     MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
C               CALCULATION
C  NEWLIF     LIFE VALUE TO BE INSERTED INTO LIFE()
C  NLIFET     TOTAL NUMBER OF LIVES CALCULATED BY PFM
C  NUM        NUMBER OF LIFE VALUES IN LIFE()
C  PLACE      POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C  TEMP()     1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
C               INSERTION OF NEWLIF


      NUM = NLIFET / 2

C     FIND POSITION IN LIFE() FOR NEWLIF

      IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

      DO 100 I = 1, NUM
         IF (NEWLIF .LT. LIFE(I)) THEN
            PLACE = I
            GOTO 110
         ENDIF
  100 CONTINUE
  110 CONTINUE

C     STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

      DO 200 I = (PLACE + 1), NUM
         TEMP(I) = LIFE(I-1)
  200 CONTINUE

C     INSERT NEWLIF

      LIFE(PLACE) = NEWLIF

C     SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

      DO 300 I = (PLACE + 1), NUM
         LIFE(I) = TEMP(I)
  300 CONTINUE

C     IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN

  400 CONTINUE


      RETURN
      END


C*************************************************************************
C   SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C   INDEPENDENT RANDOM VARIATES
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  9MAR87
C   SUBPROGRAM:  RANDOM
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C*************************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, RHO1, RHO2, THE1, THE2, X, Y

      INTEGER IOUT
```

```
      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

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

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &      ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      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

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

C  TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

      X = W * (B - A) + A
      IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X

      RETURN
      END
C***********************************************************************

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    pp. 407-409.

      REAL FUNCTION GAM (ALPHA, RAND)

C     SUBPROGRAM:  RANDOM

      COMMON  IOUT

      INTEGER IOUT

      REAL    A, ALPHA, ARG, U1, U2, V1, V2

      DOUBLE PRECISION RAND

      A = ALPHA - 1.
```

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

  10  CALL RANDOM (U1, RAND)
      CALL RANDOM (U2, RAND)
      V1 = - ALOG(U1)
      V2 = - ALOG(U2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =',
C    &                  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
C  is acknowledged.

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

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

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

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

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
     &        NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
     &        NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG

      REAL    BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
     &        FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
     &        KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
     &        MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET),
     &        RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &        SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT, MAXREG),
     &        SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(MAXREG),
     &        SXY(MAXREG), SY2(MAXREG), SZERO


C                    LIST OF VARIABLES
C
C  BIGKHT       EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
```

```
C  BZERO       VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N
C                DATA SET
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
C                AND SIG() CALCULATION
C  FTUZ        ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  IOUT        OUTPUT DUMP CONTROLLER
C  IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                FOR EACH REGION
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LAMN        LAMBDA-N -- RATIO OF Var(Ln N given S) / (m**2 C**2),
C                CONSTANT OVER REGIONS AND COMPONENTS
C  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 IN 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  MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                BOUND
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MC() FOR EACH REGION
C  MEDM()      1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C  MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MPROC       Materials PROCess variation --CONTROLS MATERIALS PROCESS
C                VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MU()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                SET IN EACH REGION
C  NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                ALL DATA SETS IN A REGION (Number of Points Per Region)
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  PVAR        MATERIALS PROCESS VARIATION
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C  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) OR 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  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  STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
```

```
C  SUHAT2()      1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                  REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT2()      1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
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  SXY()         1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                  REGION (X = Ln S, Y = Ln N)
C  SY2()         1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                  (Y = Ln N)
C  SZERO         STRESS TENSILE TEST POINT, So
C  VARY          CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                  VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                  VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  ZROREG        ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                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
C  INFORMATION

C  PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION

C  INITIALIZE PRIMARY ARRAYS

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

C  READ, CONVERT, ECHO INFORMATION

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

C  CALCULATE RESIDUAL VARIANCES

      CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, 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, MCHAT)

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C  OBTAIN POSTERIOR RANGES ON M

         CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
     &                RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

         CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)
```

```
C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

         CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

         WRITE(7,900)

         DO 25 L = 1, NUMREG
            WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
     &                      JZERO(1, L), JZERO(2, L)
   25    CONTINUE

         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,
     &                NBND, BIGKHT, BZERO)

C  CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED

         IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
            DO 200 L = 1, NUMREG
               RANGEM(1,L) = MEDM(L)
               RANGEM(2,L) = MEDM(L)
  200       CONTINUE
         ENDIF

      ELSE

C  NORMAL VARIATION IS DESIRED

C  CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION

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

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M
```

```
      CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
      CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO,
     &             MPNT, MO, SIGMA2)
C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
      CALL CONCAV (NUMREG, RANGEM)
C  WRITE RESULTS TO FILE DUMP
      WRITE(7,975)
      DO 350 L = 1, NUMREG
         WRITE(7,980) L, MCHAT(1,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(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)
      DO 380 L = 1, NUMREG
         WRITE(7,990) L, MU(L), SIG(L)
380 CONTINUE
    ENDIF
C  PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS
    IF (MPROC .EQ. 1) THEN
       WRITE(7,995) PVAR
    ENDIF
C  FORMAT STATEMENTS
900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
  &         'Technology.  U.S. Government',/,2X,'Sponsorship under ',
  &         'NASA Contract NAS7-918 is acknowledged.',////,
  &         2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
  &         ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
  &         'FOR EACH REGION',/)
905 FORMAT(7X,'REGION:  ',I1,7X,'Io = (',F12.9,',',F12.9,')',
  &         /,24X,'Jo = (',F12.9,',',F12.9,')')
910 FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
  &         //,7X,'REGION',8X,'E(C)',12X,'E(m)',/)
915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)
920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH '
  &         'REGION')
930 FORMAT(//,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)
```

```
 940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

 950 FORMAT(///)

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

 965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')

 970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

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

 980 FORMAT(9X,I1,11X,F10.6)

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

 990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)

 995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',
     &        'MEDIAN S/N CURVE',/,2X,'WARRANTED BY THE AVAILABLE ',
     &        'INFORMATION',//,7X,E11.5)

      RETURN
      END

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

C   SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C   ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C   PROGRAMMER:  L. NEWLIN
C         DATE:  5OCT87
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRMNAT

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

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

C   SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C   USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

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

C   INPUTS:    ---
C   OUTPUTS:   NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
C              NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2
```

```
C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

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

      COMMON  IOUT

      INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
     &        NPTS(0:MAXSET), REFNP(MAXREG)

      REAL    DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
     &        RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG)

C                    LIST OF VARIABLES
C
C  DELTA()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C               SIG() CALCULATION
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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 IN 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  MO()       1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C               MEAN FOR EACH REGION
C  MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C               MZERO() FOR EACH REGION
C  MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C               EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C               IS THE UPPER BOUND
C  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
C               S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

      DO 100 J = 0, MAXSET
         NPTS(J) = 0.0
  100 CONTINUE

      DO 200 L = 1, MAXREG
         DO 250 J = 0, MAXSET
            NP(J, L) = 0.0
  250    CONTINUE
  200 CONTINUE

      DO 300 J = 0, MAXSET
         DO 350 I = 1, MAXDAT
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            RATSTR(I,J) = 0.0
  350    CONTINUE
  300 CONTINUE
```

```
      DO 400 L = 1, MAXREG
         DO 425 K = 1, MAXDAT
            DO 450 J = 0, MAXSET
               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"
C   THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C   SPECFO AND RELATO.  RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C   SPECIFIED BY USER
C   PROGRAMMER:  L. NEWLIN
C         DATE:  21JUN88      FORMAT/COMMENTS:  12AUG91
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
     &                LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
     &                NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
     &                FTUZ, FTYZ, DELTA, MO, SIGMA2, 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,
C              CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2,
C              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),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RATIO, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
```

```
     &        STR(MAXDAT, MAXREG), SZERO
      CHARACTER*40  DESCRP(0:MAXSET)

C                          LIST OF VARIABLES
C
C   COUNT      INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO,
C                CONVERSION, AND BREAK UP
C   CZERO      EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C   DELTA()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                SIG() CALCULATION
C   DESCRP()   1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C   FTU        ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
C   FTUZ       ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   FTY        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   FTYZ       YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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   KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C   L          CONTROLS DO LOOP FOR EACH REGION
C   LAMN       LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                CONSTANT OVER ALL REGIONS AND COMPONENTS
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   M          CONTROLS DO LOOP FOR EACH DATA DIVISION
C   MAXDAT     MAXIMUM NUMBER OF POINTS IN 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   MO()       1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C   MPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C   MPROC      Materials PROcess variation -- CONTROLS MATERIALS PROCESS
C                VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C   MZERO()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C   NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C   NDIV       NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
C                REGION PAIRS DURING INPUT
C   NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NNODAT     Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NP()       2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                IN EACH REGION
C   NPTS()     1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   NSETS      NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUM        NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   RATIO      STRESS RATIO (R = -1.0 IS DESIRED)
C   RATSTR()   2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C                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) OR 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   REG        REGION OF INTEREST IN A PARTICULAR DIVISION
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
C                S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZERO      STRESS TENSILE TEST POINT, So
C   VARY       CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                VARIATION;  3 - TRUNCATED NORMAL VARIATION
C   ZROREG     ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                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), FTY, FTU, NDIV, NPTS(0)

      IF (NPTS(0) .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF

      WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
      IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)

      WRITE(3,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

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

      FTUZ = FTU
      FTYZ = FTY

C     INPUT STRESS/LIFE INFORMATION -- INCLUDING STRESS RATIO AND REGION
C     INFORMATION FROM SPECFD AND ECHO TO SPECFO

      DO 100 M = 1, NDIV

         READ (1,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE FOR RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

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

         DO 110 I = (COUNT + 1), (COUNT + NUM)
            READ(1,*) RAWSTR(I,0), RAWNF(I,0)
  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, FTY)

         ENDIF

C        ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL

         DO 130 I = (COUNT + 1), (COUNT + NUM)
```

```
            WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
     &                   RATSTR(I,0), RAWNF(I,0)

            IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
     &                   RATIO, REG, RATSTR(I,0), RAWNF(I,0)

  130     CONTINUE

C         BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C         EXPCTD, AND PAREST

          K = NP(0,REG)

          DO 140 I = (COUNT + 1), (COUNT + NUM)

             K = K + 1
             LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
             LNNF(K,0,REG) = ALOG(RAWNF(I,0))
             STR(K,REG) = RATSTR(I,0)
             NF(K,REG) = RAWNF(I,0)

  140     CONTINUE

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

          NP(0,REG) = K
          REFNP(REG) = K
          COUNT = COUNT + NUM

  100 CONTINUE

      IF (NPTS(0) .NE. COUNT) THEN
         WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &              'INCORRECTLY SPECIFIED'
         WRITE(8,*) 'IN SPECIFIC DATA SET'
         CALL TRMNAT
      ENDIF

      READ(1,*) SZERO
      IF (NINT (SZERO) .GT. 0) THEN
         ZROREG = 0
      ELSE
         ZROREG = 1
      ENDIF
      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG

C     INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION

      READ(1,*) NUMREG, NNODAT

      IF ((NUMREG + NNODAT) .GT. MAXREG)  THEN
         WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON NUMBER OF REGIONS'
         CALL TRMNAT
      ENDIF

      DO 150 L = ZROREG, (NUMREG + NNODAT)
         READ(1,*) NBND(L)
  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
         ENDIF
  180 CONTINUE

      IF (VARY .EQ. 3) THEN
C        READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
         WRITE(3,945)
         IF (IOUT .EQ. 10) WRITE(8,945)
         DO 190 L = 1, (NUMREG + NNODAT)
            READ(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
            ENDIF
  190    CONTINUE
      ENDIF

      IF (MPROC .EQ. 1) THEN
         READ(1,*) KRATIO, LAMN
         WRITE(3,955) KRATIO, LAMN
         IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
      ENDIF

C  BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATD
C  AND THEN ECHO TO RELATO

      READ(5,*) NSETS

      IF (NSETS .GT. MAXSET) THEN
         WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF RELATED DATA SETS'
         CALL TRMNAT
      ENDIF

      WRITE(6,935) NSETS

      DO 200 J = 1, NSETS

         COUNT = 0

         IF (IOUT. EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS

         READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)

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

```
      ENDIF

      WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
      IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)

      WRITE(6,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

      DO 300 M = 1, NDIV

         READ(5,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE OF RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

         IF (REG .GT. MAXREG) THEN
            WRITE(8,*)
     &         'ERROR:  OVER REGION LIMIT IN RELATED MATERIAL ', J
            CALL TRMNAT
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
            WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
         ENDIF

         DO 310 I = (COUNT + 1), (COUNT + NUM)
            READ(5,*) RAWSTR(I,J), RAWNF(I,J)
  310    CONTINUE

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,I2)

 905 FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
     &         /,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
     &         'STRESS',7X,'LIFE'/)

 910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

 913 FORMAT(//)

 914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN So OF',
     &         5X,E11.5)

 915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA  ',
     &         /,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
     &         /,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
     &         '(CYCLES): ',/)

 920 FORMAT(10X,E9.3)

 925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
     &         'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
     &         //,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
     &         //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
     &         5X,'UPPER BOUND',/)

 930 FORMAT(6X,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****************************************************************************
```

```
C  THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C  STRESS RATIO, R, IS NOT -1.0
C  PROGRAMMER:  L. NEWLIN
C       DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C    VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                     V8.3, V8.4, V8.5
C              MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
C
C  FTU        ULTIMATE STRENGTH OF MATERIAL (PSI)
C  FTY        YIELD STRENGTH OF MATERIAL (PSI)
C  I          CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  J          DATA SET OF INTEREST
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXSET     MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  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 * (1 - R)/2 , TO BE COMPARED WITH FTY

C  Kt IS ASSUMED TO BE ONE

      DO 100 I = NUM1, NUM2

         TEST = STR(I,J) * (1.0 - R)/2.0

         IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST


         IF (TEST .GE. FTY) THEN

            RSTR(I,J) = TEST

            IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() =',RSTR(I,J)

         ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN

            RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))

            IF (IOUT.EQ.10) WRITE(8,*)'2:RSTR() =',RSTR(I,J)

         ELSE

            RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
     &                  /(2.0 * FTU)))

            IF (IOUT.EQ.10) WRITE(8,*)'3:RSTR() =',RSTR(I,J)

         END IF
```

```
 100 CONTINUE

     RETURN
     END


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


C  SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C  AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C  X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C  PROGRAMMER:  L. NEWLIN
C       DATE:  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              MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   NUMREG, NSETS, NP, LNSTR, LNNF
C  OUTPUTS:  SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR

C     IMPLICIT NONE

      INTEGER 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)
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                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
C               L AND DATA SET J (X = Ln S)
C  MEANY()    1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
C               L AND DATA SET J (Y = Ln N)
C  NP()       2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C               SET IN EACH REGION
C  NPPR()     1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C               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
C               REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SWHAT2()   1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
```

```
C              REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
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  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

         DO 200 J = 0, NSETS
C           FIRST CALCULATE SAMPLE X AND Y MEANS
C           FOR DATA SET J IN REGION L
            MEANY(J) = 0.0
            MEANX(J) = 0.0
            IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
     &                        ' NP =', NP(J,L)

            DO 250 K = 1, NP(J,L)
               MEANY(J) = MEANY(J) + LNNF(K,J,L)
               MEANX(J) = MEANX(J) + LNSTR(K,J,L)
               IF (IOUT .EQ. 10) WRITE(8,*) 'LNNF =', LNNF(K,J,L),
     &               ' LNSTR =', LNSTR(K,J,L)
  250       CONTINUE

            MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
            MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
            IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY(J) =', MEANY(J),
     &         ' MEANX(J) =', MEANX(J)

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

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

            NPPR(L) = NPPR(L) + NP(J,L) - 1
```

```
            IF (IOUT .EQ. 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
C            OUR MODEL
             WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
             CALL TRMNAT
          ENDIF

          NPPR(L) = NPPR(L) - 1

          IF (NPPR(L) .LE. 0) THEN
             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))

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,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
     &                  ' SX2(L) =', SX2(L)
             WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
     &                  ' BB(L) =', BB(L)
          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) =',
     &             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 INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, Io, ON
C  C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5OCT87   COMMENTS:  15SEP89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                   JZERO, MCHAT)
```

```
C  INPUTS:    NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C  OUTPUTS:  IZERO, JZERO, MCHAT
C  SUBPROGRAMS:  TRMNAT

C      IMPLICIT NONE

       INTEGER CHITAB, MAXREG, TTAB

       PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)

       COMMON  IOUT

       INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG

       REAL    ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
     &         IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
     &         SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
     &         SX2(MAXREG), T, T025(TTAB)

       DATA (CHI025(I), I = 1, 75) /
     &                   0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
     &                   1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
     &                   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,
     &                   29.15,    29.95,   30.75,   31.55,   32.3574,
     &                   33.16,    33.96,   34.77,   35.58,   36.39,
     &                   37.21,    38.02,   38.84,   39.66,   40.4817,
     &                   41.30,    42.12,   42.95,   43.77,   44.60,
     &                   45.43,    46.26,   47.09,   47.92,   48.7576,
     &                   49.59,    50.42,   51.26,   52.10,   52.94 /
       DATA (CHI025(I), I = 76, 150) /
     &                   53.78,    54.62,   55.46,   56.30,   57.1532,
     &                   57.80,    58.84,   59.69,   60.54,   61.39,
     &                   62.24,    63.09,   63.94,   64.79,   65.6466,
     &                   66.50,    67.35,   68.21,   69.07,   69.92,
     &                   70.78,    71.64,   72.50,   73.36,   74.2219,
     &                   75.08,    75.94,   76.80,   77.67,   78.53,
     &                   79.40,    80.27,   81.13,   82.00,   82.87,
     &                   83.73,    84.60,   85.47,   86.34,   87.21,
     &                   88.08,    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 /

       DATA (CHI975(I), I = 1, 75) /
     &                   5.02389,  7.37776, 9.34840, 11.1433, 12.8325,
     &                   14.4494,  16.0128, 17.5346, 19.0228, 20.4831,
     &                   21.9200,  23.3367, 24.7356, 26.1190, 27.4884,
     &                   28.8454,  30.1910, 31.5264, 32.8523, 34.1696,
     &                   35.4789,  36.7807, 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,
     &                   54.44,    55.67,   56.89,   58.12,   59.3417,
     &                   60.56,    61.77,   62.99,   64.20,   65.41,
     &                   66.62,    67.82,   69.02,   70.22,   71.4202,
     &                   72.61,    73.81,   75.00,   76.19,   77.38,
     &                   78.57,    79.75,   80.93,   82.12,   83.2976,
     &                   84.48,    85.65,   86.83,   88.00,   89.18,
     &                   90.35,    91.52,   92.69,   93.86,   95.0231,
     &                   96.19,    97.35,   98.52,   99.68,   100.84 /
       DATA (CHI975(I), I = 76, 150) /
     &                   102.00,   103.16,  104.31,  105.47,  106.629,
     &                   107.78,   108.94,  110.09,  111.24,  112.39,
     &                   113.54,   114.69,  115.84,  116.99,  118.136,
     &                   119.28,   120.43,  121.57,  122.72,  123.86,
     &                   125.00,   126.14,  127.28,  128.42,  129.561,
     &                   130.70,   131.84,  132.98,  134.11,  135.25,
```

```
     &                 136.38,    137.52,   138.65,   139.79,   140.92,
     &                 142.05,    143.18,   144.31,   145.44,   146.57,
     &                 147.70,    148.83,   149.96,   151.09,   152.21,
     &                 153.34,    154.47,   155.59,   156.72,   157.84,
     &                 158.97,    160.09,   161.21,   162.33,   163.46,
     &                 164.58,    165.70,   166.82,   167.94,   169.06,
     &                 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 /

C  VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:
C
C     1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719
C
C     31-39, 41-49, 51-59, 61-69, 71-79, 81-89, 91-99, 101-150
C         -- CALCULATED USING CUBE RULE APPROXIMATION

      DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
     &             2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
     &             2.160, 2.145, 2.131, 2.120, 2.110, 2.101,
     &             2.093, 2.086, 2.080, 2.074, 2.069, 2.064,
     &             2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /

C                         LIST OF VARIABLES
C
C  ARG         INTERMEDIATE CALCULATION VARIABLE
C  CHI025()    TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHI975()    TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHITAB      MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  I           CONTROLS LOOP FOR CHI025() AND CHI975()
C  IOUT        OUTPUT DUMP CONTROLLER
C  IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                FOR EACH REGION
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
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  NUM         EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  SUHAT       EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SUHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT       EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SWHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SX          EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
C  SX2()       1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                (X = Ln S)
C  T           VALUE OF T025() USED IN CALCULATIONS
C  T025()      TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
C  TTAB        MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C  INITIALIZE IZERO, JZERO AND MCHAT

      DO 50 L = 1, MAXREG
         IZERO(1,L) = 0.0
         IZERO(2,L) = 0.0
         JZERO(1,L) = 0.0
         JZERO(2,L) = 0.0
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
   50 CONTINUE
```

```
C  CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

      DO 75 L = 1, NUMREG
         IF (NPPR(L) .GT. CHITAB) THEN
            WRITE(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,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
     &                  IZERO(2,L)
            WRITE(8,*) 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
     &                  JZERO(2,L)
            WRITE(8,*) 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
     &                  MCHAT(2,L)
         ENDIF
  100 CONTINUE

      RETURN
      END


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


C  SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C  THE Co GIVEN BY THE USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  8OCT87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
```

```
      SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

C  INPUTS:   NUMREG, CZERO, SX2, SXY, SY2
C  OUTPUTS:  MCPNT, MC

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, MCPNT(MAXREG), NUMREG

      REAL     ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
     &         SXY(MAXREG), SY2(MAXREG)


C                     LIST OF VARIABLES
C
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     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         MCPNT(L) = 0
         MC(1,L) = 0.0
         MC(2,L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATIONS

      CZERO2 = CZERO ** 2

      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

      DO 100 L = 1, NUMREG

         ARG1 = SX2(L) - CZERO2
         ARG2 = 0.0

         IF (CZERO .EQ. 0.0) THEN

C           THEN NO M CONSTRAINT IS REQUIRED

            MCPNT(L) = 0

         ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

C           THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M
```

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

         ELSE

C           THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C           COMMON CALCULATIONS

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

            IF (SX2(L) .LT. CZERO2) THEN

C              AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

               MCPNT(L) = 1
               MC(1,L) = (- SXY(L) - ARG2) / ARG1

            ELSE

C              SX2(L) .GT. CZERO2 -- THIS TIME THE M CONSTRAINT IS A RANGE

               MCPNT(L) = 2
               MC(1,L) = (- SXY(L) - ARG2) / ARG1
               MC(2,L) = (- SXY(L) + ARG2) / ARG1

            ENDIF

         ENDIF
  100 CONTINUE

      IF (IOUT .EQ. 10) THEN

         DO 200 L = 1, NUMREG

            WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
            WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

  200    CONTINUE

      ENDIF

      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
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM 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
```

```
      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


C                    LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  J            CONTROLS DO LOOP FOR EACH DATA SET
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LAMN         LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                 CONSTANT OVER REGIONS AND COMPONENTS
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET       MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                 MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  NP()         2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                 SET IN EACH REGION
C  NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM()        EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
C                 NUMBER OF POINTS IN EACH DATA SET
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  PSIG2()      1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C                 VARIATION IN EACH REGION
C  PVAR         THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                 CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  SUM          WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
C                 AVERAGE
C  TOTAL        SUM OF NUM() OVER ALL REGIONS

C     INITIALIZE VARIABLES

      SUM = 0.0
      TOTAL = 0.0

      DO 50 L = 1, MAXREG
         PSIG2(L) = 0.0
         NUM(L) = 0
   50 CONTINUE

      DO 100 L = 1, NUMREG
         DO 150 J = 0, NSETS
            NUM(L) = NUM(L) + NP(J,L)
  150    CONTINUE
         NUM(L) = NUM(L) - 1
         TOTAL = TOTAL + NUM(L)
  100 CONTINUE

      DO 200 L = 1, NUMREG
         PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
         SUM = SUM + PSIG2(L) * NUM(L)
  200 CONTINUE

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'LAMN = ', LAMN
         DO 300 L = 1, NUMREG
            WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
            WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
  300    CONTINUE
         WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
      ENDIF

      PVAR = SUM / FLOAT (TOTAL)
```

```
      RETURN
      END

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

C  SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C  M AND Co WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C  TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:   2FEB88       FORMAT/COMMENTS:   12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
     &                   MCHAT, RANGEM)

C  INPUTS:    NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
     &        MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER


C                          LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  JZERO()      2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LOWER        LOWER BOUND OF INTERSECTION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()         2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                 REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                 -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                 BOUND
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION -- MCHAT(1,L) = - DD(L), THE ESTIMATE
C                 FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  UPPER        UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

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

```
C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

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

         IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS NO EXOGENOUS INFORMATION
C           ASSUME RANGE TO BE Jo

            RANGEM(1,L) = JZERO(1,L)
            RANGEM(2,L) = JZERO(2,L)

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' JZERO(1,L) = ', JZERO(1,L)
               WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

C           NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
C           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

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

C           THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
C           CORRESPONDING TO THE Co CONSTRAINT, ADJUST Jo ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

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

```
C           THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER
C           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
C           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)
               WRITE(8,*) '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

C           THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C           CONSTRAINT, INTERSECT Jo AND Mo, ADJUSTING THE LOWER BOUND
C           BY Mc ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANEGM(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
C           INTERSECT THESE TWO RANGES WITH Jo

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
```

```
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSE

         WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
         CALL TRMNAT

      ENDIF

C     RESTRICT RANGE TO BE NON-NEGATIVE

      RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

      IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END


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


C  SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C  WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88     FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  INPUTS:    RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C  OUTPUTS:   RANGEM, MCHAT, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT
```

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

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

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           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

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            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

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
C        VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                         V8.4, V8.5
C                  MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONCAV (NUMREG, RANGEM)

C     INPUTS:    NUMREG, RANGEM
C     OUTPUTS:   RANGEM
C     SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    RANGEM(2, MAXREG), TESTM


C                              LIST OF VARIABLES
C
C     IOUT        OUTPUT DUMP CONTROLLER
C     L           CONTROLS DO LOOP FOR EACH REGION
C     MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C     NUMREG      NUMBER OF REGIONS OF INTEREST
C     RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND
C     TESTM      UPPER BOUND OF RANGE ON M IN REGION L-1 -- USED DURING
C                   CONCAVITY ADJUSTMENT


C     ADJUST RANGE TO INSURE CONCAVITY

      DO 100 L = NUMREG, 2, -1

         IF (RANGEM(2,L-1) .EQ. 0.0) THEN
C           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
C           RANGE IS AN INTERVAL IN REGION L-1
            TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
            IF (TESTM .LT. RANGEM(1,L-1)) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &            ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
     &            'REGION ', L-1
               CALL TRMNAT
            ELSE
               RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
            ENDIF
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L-1) =', RANGEM(1,L-1),
     &                 ' RANGEM(2,L-1) =', RANGEM(2,L-1)
            WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                 ' RANGEM(2,L) =', RANGEM(2,L)
            WRITE(8,*) 'TESTM =', TESTM, ' L = ', L
         ENDIF

  100 CONTINUE

      RETURN
```

```
      END

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

C  SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C  BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5OCT87     COMMENTS:  1DEC87
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C  INPUTS:   NUMREG, RANGEM
C  IOUTPUT:  MEDM

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)


C                   LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LOWERM       LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
C                 TO BE USED IN MEDIAN CALCULATION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEDM()       1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND

C     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
C           -- 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)
            WRITE(8,*) '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
C    VERSION:  MATCHR V8.3, V8.4, V8.5   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 EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
     &                   ZROREG, NBND, BIGK1, BZHAT)

C  INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C  OUTPUTS:  BIGK1, BZHAT
C  SUBPROGRAMS:  TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL    BIGK(0:MAXREG), BIGK1, 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)

C                      LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  BIGK1      EQUAL TO BIGK(1)
C  BZHAT      E(BETAo)
C  FACTR      A SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  KHAT       E(k)
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MEDM()     1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  MM()       1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
```

```
C                REGIONS OF INTEREST
C   NCOMPS      Number of COMPonents -- 1 FOR STRESS AND STRAIN WHEN DECOMPOSED
C                DATA UNAVAILABLE -- 2 FOR DECOMPOSED STRAIN DATA
C   NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                 DATA SET
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR
C                 THE SPECIFIC MATERIAL S/N DATA SET
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 REGION
C                CONTAINED IN NBND()
C   STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C                DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZ2         SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C   SZERO       STRESS TENSILE TEST POINT, So
C   TRBIGK()    1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C                TRBIGK(i) = BIGK(i)
C   ZROREG      ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                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  INITIALIZE VARIABLES

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

C  CREATE MM() ARRAY FROM MEDM() ARRAY

      DO 100 L = 1, NUMREG
         MM(L) = MEDM(L)
  100 CONTINUE

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

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

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE BETAo AND k

      CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C  CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

      CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

      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), SBND(L)
            IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
     &                                     NBND(L), SBND(L)
  200    CONTINUE

         WRITE(7,920)

      ELSE

         WRITE(7,930) MM(1), BIGK(1), KHAT

      ENDIF

C  FORMAT STATEMENTS

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

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

  920 FORMAT(///)

  930 FORMAT(//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &       //,11X,'m',14X,'K',13X,'E(k)',
     &       //,7X,F8.5,5X,E12.5,6X,F7.4,/)

      RETURN
      END


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


C  SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C  MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      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
C  ARG         INTERMEDIATE CALCULATION VARIABLE
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
```

```
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                  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  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 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  SUMX2         EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
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     INITALIZE ARRAYS

      DO 50 L = 1, MAXREG
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
         MU(L) = 0.0
         SIG(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATION FOR EACH REGION

      DO 100 L = 1 , NUMREG

         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SQRT (SUHAT2(L))
         SUMX2 = NPPR(L) * SX2(L)
         ARG = SUMX2 + DELTA(L)

         IF (DELTA(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C           USE THE ESTIMATE OF M
            MU(L) = MCHAT(1,L)
         ELSE
C           UPDATE THE ESTIMATE OF M WITH Mo USING DELTA
            MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
         ENDIF

         IF (SIGMA2(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
C           USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
            SIG(L) = SQRT (SWHAT2(L) / ARG)
         ELSE
            SIG(L) = SQRT (SIGMA2(L) / ARG)
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
     &                 MCHAT(1,L)
            WRITE(8,*) 'SUHAT2 = ', SUHAT2(L),  ' MCHAT2 = ',
     &                 MCHAT(2,L)
            WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
     &                ' SUMX2 = ', SUMX2
            WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
            WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
            WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
     &                ' SIG = ', SIG(L)
         ENDIF
```

```
  100 CONTINUE

      RETURN
      END

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      VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C   INPUTS:     NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C   OUTPUTS:    RANGEM
C   SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
     &        RANGEM(2, MAXREG), UPPER

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

C     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
C           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 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. 1)) THEN

C           THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C           CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc

            LOWER = AMAX1(MZERO(1,L), MC(1,L))
            UPPER = MZERO(2,L)
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANEGM(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
C           INTERSECT THESE TWO RANGES

            LOWER = AMAX1(MZERO(1,L), MC(1,L))
            UPPER = AMIN1(MZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF
```

```
         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSE

         WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
         CALL TRMNAT

      ENDIF

C        RESTRICT RANGE TO BE NON-NEGATIVE

      RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

      IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END

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


C  SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C  DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88     FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C            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)
```

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

      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
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            MU(LL) = MO(LL)
            SIG(LL) = SQRT(SIGMA2(LL))
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
               WRITE(8,*) 'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
               WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
     &                    SIGMA2(LL)
            ENDIF

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
               MU(LL) = RANGEM(1,LL)
               SIG(LL) = 0.0
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
     &         ' MU = ', MU(LL), ' SIG = ', SIG(LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE
```

```
      RETURN
      END

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

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                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   INPUTS:    VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
C              NBND, STR
C   OUTPUTS:   BIGK, BZERO, MM, SBND
C   SUBPROGRAMS:  FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG

      REAL     BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG),
     &         MU(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
     &         STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

      DOUBLE PRECISION RAND

C                           LIST OF VARIABLES
C
C   BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                EACH REGION
C   BZERO      VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C   IOUT       OUTPUT DUMP CONTROLLER
C   K          VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C   L          CONTROLS DO LOOP FOR EACH REGION
C   MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C   MM()       1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C   MU()       1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C   NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C   NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C   NPTS()     1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C                SPECIFIC MATERIAL S/N DATA SET
C   NUMREG     NUMBER OF REGIONS OF INTEREST
C   RANGEM()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C   RAND       RANDOM NUMBER SEED
```

```
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()
C  SIG()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
C               STANDARD DEVIATION FOR EACH REGION
C  STR()      2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C               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;
C               1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;
C               3 - TRUNCATED NORMAL VARIATION
C  ZROREG     ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C               BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C  ZZ()       1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C               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, 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,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. 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

      SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C   INPUTS:    RAND, NUMREG, RANGEM
C   OUTPUTS:   MM
C   SUBPROGRAMS:  RANDOM, TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X

      DOUBLE PRECISION RAND

C                         LIST OF VARIABLES
C
C   IOUT        OUTPUT DUMP CONTROLLER
C   L           CONTROLS DO LOOP FOR EACH REGION
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()        1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   PICK()      1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C   RAND        RANDOM NUMBER SEED
C   RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C   X           UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                 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)
            WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
          ENDIF
        ELSE
C         ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C         AND THEN SAMPLE
          PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
          PICK(2) = RANGEM(2,L)
          IF (PICK(1) .GT. PICK(2)) THEN
C            NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C            STOP PROGRAM
             WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
             CALL TRMNAT
          ELSE
C            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),
     &                   ' RANGEM(1,L) =', RANGEM(1,L)
             WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
             WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                   ' MM(L) =', MM(L)
          ENDIF
        ENDIF

  100 CONTINUE

      RETURN
      END


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


C***********************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C***********************************************************************

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

C                 LIST OF VARIABLES
C
C  FRAC        UNIFORM (0,1) RANDOM VARIATE
C  IOUT        OUTPUT DUMP CONTROLLER
C  RANA        CONSTANT FOR LCG
C  RANC        CONSTANT FOR LCG
C  RAND        RANDOM NUMBER SEED
C  RANDIV      INTERNAL CALCULATION
C  RANM        CONSTANT FOR LCG
C  RANSUB      INTERNAL CALCULATION
```

```
C  RANT        INTERNAL CALCULATION
C  RANX        INTERNAL CALCULATION


C          USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

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

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

      RETURN
      END

C     NOTES:  IOUT=2 DUMPS TO SCREEN


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


C  SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:   7JUN88      COMMENTS:  13FEB89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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


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

```
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C  SIG()       1-D ARRAY CONTAINING THE STANDARD DEVIATION OF M FOR EACH
C                REGION
C  X            NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                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
   10       CALL NORMGN (RAND, MU(L), SIG(L), X)
            IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
            MM(L) = X
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                    ' RANGEM(2,L) =', RANGEM(2,L)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              SAMPLE ON ADJUSTED RANGE
   20          CALL NORMGN (RAND, MU(L), SIG(L), X)
               IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
               MM(L) = X
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF

  100 CONTINUE

      RETURN
      END


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


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

```
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  3FEB88
C     VERSION:  MATCHR V7, V7.1, 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
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C*************************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT

C             LIST OF VARIABLES
C
C  FRAC    UNIFORM(0,1) RANDOM VARIATE
C  IOUT    OUTPUT DUMP CONTROLLER
C  MU      MEAN OF NORMAL DISTRIBUTION
C  RAND    RANDOM NUMBER SEED
C  SIGMA   STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X       NORMAL RANDOM VARIATE
C  U1      UNIFORM RANDOM NUMBER U(0,1)
C  U2      UNIFORM RANDOM NUMBER U(0,1)
C  Z1      NORMAL RANDOM NUMBER ON N(0,1)
C  Z2      NORMAL RANDOM NUMBER ON N(0,1)


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

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

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

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

      X = SIGMA * Z1 + MU
      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 V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   NPTS, STR, NF, NUMREG, MM, NBND
C  OUTPUTS:  NP, ZZ

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON IOUT

      INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG

      REAL     MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         STR(MAXDAT, MAXREG), ZZ(MAXDAT)


C                        LIST OF VARIABLES
C
C  I           CONTROLS DO LOOP FOR EACH DATA POINT
C  IOUT        OUTPUT DUMP CONTROLLER
C  K           CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LL          CONTROLS INNER DO LOOP FOR EACH REGION
C  MAXDAT      MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()        1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C  MML         EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()      1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C                SPECIFIC MATERIAL S/N DATA SET
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  ZZ()        1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
C                Z = F(STR,NF,NBND,MM)


C  INITIALIZE VARIABLES

      NP = 0

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


C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG
         MML = MM(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =',
     &         MML, ' NPTS =', NPTS(L)

         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))
     &                  * ((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) =',
```

```
     &            MM(LL), ' ZZ =', ZZ(NP)
 300        CONTINUE

 200     CONTINUE

 100 CONTINUE

      RETURN
      END

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

C  SUBROUTINE SMNVAR CALCULATES THE Sample MeaN and VARiance OF
C  Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  24AUG87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V3.3, V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
C  I         CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT      OUTPUT DUMP CONTROLLER
C  MAXDAT    MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MEANZ     SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  NP        TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C               DATA SET
C  SZ2       SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZZ()      1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)

C  INITIALIZE VARIABLES

      MEANZ = 0.0
      SZ2 = 0.0

C  CALCULATE THE MEAN OF ZZ(), MEANZ

      DO 100 I = 1, NP
         MEANZ = MEANZ + ZZ(I)
         IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I,
     &      ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
 100  CONTINUE
      MEANZ = MEANZ / FLOAT(NP)
      IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ

C  CALCULATE THE VARIANCE OF ZZ(), SZ2

      DO 200 I = 1, NP
```

```
          SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
          IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
  200 CONTINUE
      SZ2 = SZ2 / FLOAT(NP - 1)
      IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

      RETURN
      END


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


C   SUBROUTINE KBETA CALCULATES k AND BETAo FROM THE SAMPLE MEAN AND
C   VARIANCE OF Z = F(STR, NF, NBND, MM)
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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


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


C   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
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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)


C                    LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
C              MATERIAL DATA BASE
C  GAMMA     EULER'S CONSTANT
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C              DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST


C  INITIALIZE VARIABLES

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

C  CALCULATE K FOR REGION ONE

      BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C      WRITE(7,*) 'REGION:  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)))
C         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
C         DATE:  22DEC88
C      VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.2, V4.3, V4.4, V4.5

      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)


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


C   INITIALIZE SBND()

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

C    CALCULATE SBND(0) IF ZROREG = 0

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

C   CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES

      DO 100 L = 1, NUMREG
         IF (NBND(L) .GE. 1.0E+36) THEN
            SBND(L) = 0.0
```

```
         ELSE
            SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
         ENDIF
  100 CONTINUE

      RETURN
      END


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


C  THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
C  MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C  TRANSFORM METHOD"
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  18MAR87     COMMENTS:  15SEP89
C     VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                      V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                      V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE WEIBGN (BETA, RAND, WEIB)

C  INPUTS:    BETA, RAND
C  OUTPUTS:   WEIB
C  SUBPROGRAMS:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER IOUT

      REAL    ARG, BETA, ETA, FRAC, WEIB

      DOUBLE PRECISION RAND


C                  LIST OF VARIABLES
C
C  ARG      INTERMEDIATE CALCULATION VARIABLE
C  BETA     WEIBULL DISTRIBUTION SHAPE PARAMETER
C  ETA      WEIBULL DISTRIBUTION LOCATION PARAMETER
C  FRAC     UNIFORM (0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  RAND     RANDOM NUMBER SEED
C  WEIB     WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE


C     CALCULATE CONSTRAINED ETA

      ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))

C     GENERATE WEIBULL RANDOM VARIATE

      CALL RANDOM(FRAC, RAND)
      ARG = -ALOG(1.0 - FRAC)
      WEIB = ETA * ARG**(1.0/BETA)
      IF (IOUT .EQ. 10) WRITE(8,*)'BETA = ', BETA, ' ETA =', ETA,
     &   ' FRAC =', FRAC, ' ARG =', ARG, ' WEIB =', WEIB

      RETURN
      END
```

```
C***********************************************************************

C  SUBROUTINE KOMO CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA
C  REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
C  AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
C  PROGRAMMER :  L. NEWLIN
C          DATE:  1AUG91
C       VERSION:  MATCHR V8.5   MATGRM V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
     &                  FACTR, NUMREG)

C  INPUTS:    SZERO, BIGK, MM, NBND, TRSBND, FACTR
C  OUTPUTS:   TRBIGK, MM, TRSBND

C  .   IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1         SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)


C                        LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  FACTR      SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NUMREG     NUMBER OF REGIONS
C  SCLK       ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
C  SZERO      STRESS TENSILE TEST POINT, So
C  TRBIGK()   1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
C               SBND(0) < So FOR EACH TRIAL
C  TRSBND()   1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
C               LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
C               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, ' BIGK1 = ', TRBIGK(1)
```

```
         WRITE(8,*) 'MM1 = ', MM(1), ' MMo = ', MM(0)
      ENDIF

      RETURN
      END

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


C   FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C   BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C   PROGRAMMER:  L. NEWLIN
C         DATE:  10FEB89
C      VERSION:  MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND,
     &                      ZROREG, NUMREG, SZERO)

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

C     IMPLICIT NONE

      INTEGER IOUT, L, MAXREG, NUMREG, ZROREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      REAL     GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
     &         MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

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

      GETLIF = 0.0

C   CALCULATE CYCLES TO FAILURE

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

```
            IF (S .GT. SBND(L)) THEN
               TEMP = LNA(L) + LPHIM(L) + MM(L) * ( - ALOG(S)
     &                    +  ALOG (KRATIO) + LNZ)
               IF (TEMP .GT. 86.0) THEN
                  TEMP = 86.0
               ENDIF
               GETLIF = EXP (TEMP)
               GOTO 150
            ENDIF
  100    CONTINUE
      ENDIF
  150 CONTINUE

      GTLIFE = GETLIF

      RETURN
      END


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


C  SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C  M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB88
C     VERSION:  MATCHR V7, V7.1, 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
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, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

      PARAMETER (MAXMM = 20001, MAXREG = 3)

      LOGICAL INORDR

      REAL    ALLM(MAXMM, MAXREG), TEMP

C                   LIST OF VARIABLES
C
C  ALLM()     2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
C  I          CONTROLS INSERTION POINTER
C  INC        SORT INCREMENT VARIABLE
C  INORDR     FLAG TO INDICATE WHETHER SORT IS FINISHED
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXMM      MAXIMUM NUMBER OF M'S TO BE SORTED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUM        NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  TEMP       TEMPORARY SORTING VARIABLE


      DO 400 L = 1, NUMREG

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

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

          IF (.NOT. INORDR) GOTO 20
          GOTO 10
       ENDIF

400 CONTINUE

      RETURN
      END


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
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  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, CCY, CCZ, CLY, CLZ, OVAL,
     &                 LOCAT, MM, LNA, LPHIM, KRATIO, LNZ, SBND,
     &                 SZERO, ZROREG, NUMREG, STRHIS, NRAN, PERIOD,
     &                 TRUNC, ANGLE, CULPRT)

C       INPUTS:  PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, FTY, FTU,
C                EM, K, PSUBI, TIB, TOB, DI, WOFF, LAMW, RB, WD, FK, RT,
C                CCY, CCZ, CLY, CLZ, OVAL, LOCAT, MM, LNA, LPHIM, KRATIO,
C                LNZ, SBND, SZERO, ZROREG, NUMREG, STRHIS, NRAN, PERIOD,
C                TRUNC, ANGLE
C      OUTPUTS:  ELWELD, CULPRT
C  SUBPROGRAMS:  M2L1, M2L2, NARBN1, TRMNAT

C     IMPLICIT NONE

      COMMON    IOUT

      INTEGER   CULPRT, IOUT, LOCAT, MAXLD, MAXM, MAXREG, NLOAD, NRAN,
     &          NUMREG, ZROREG

      REAL      PI

      PARAMETER (MAXLD = 16, MAXM = 24000, MAXREG = 3,
     &           PI = 3.1415926536)

      REAL      ANGLE, CCY, CCZ, CLY, CLZ, DI, ELWELD, EM, FATLIF,
     &          FK(10), FTY, FTU, K(2, 2), KRATIO, LAMW, LIFE, LIFE1,
     &          LIFE2, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), M(2, MAXLD),
     &          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
```

```
C
C  ANGLE     ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --
C              GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
C  CCY       OUT-OF-PLANE CIRCUMFERENTIAL STRESS CARRYOVER FACTOR
C  CCZ       IN-PLANE CIRCUMFERENTIAL STRESS CARRYOVER FACTOR
C  CLY       OUT-OF-PLANE AXIAL STRESS CARRYOVER FACTOR
C  CLZ       IN-PLANE AXIAL STRESS CARRYOVER FACTOR
C  CULPRT    LOCATION CAUSING FAILURE
C  DI        INTERIOR DIAMETER
C  EM        ELASTIC MODULUS
C  FATLIF    VALUE OF LIFE CALCULATED
C  FK()      1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C              CONCENTRATION DUE TO WELD ECCENTRICITY
C  FTY       YIELD STRENGTH
C  FTU       ULTIMATE STRENGTH
C  IOUT      OUTPUT DUMP CONTROLLER
C  K()       FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR PIPE
C              EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR PIPE EXTERIOR
C              FOR HOOP DIRECTION; K(1,2) IS FOR PIPE INTERIOR FOR AXIAL
C              DIRECTION; K(2,2) IS FOR PIPE INTERIOR FOR HOOP DIRECTION
C  KRATIO    RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  LAMW      ACCURACY FACTOR OF Fk - r/t CURVE
C  LIFE      VALUE OF LIFE CAUSING FAILURE
C  LIFE1     VALUE OF LIFE AT LOCATION 1
C  LIFE2     VALUE OF LIFE AT LOCATION 2
C  LNA()     1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C  LNZ       NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  LPHIM()   1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C              PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C  LOCAT     LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE
C              PIPE AND 2 IS THE INTERIOR SURFACE OF THE PIPE
C  M()       2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C              ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C  MAXLD     MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C  MAXM      MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING VALUES FOR m FOR EACH REGION
C  MSTAT()   1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE
C              M2 LOAD; M(2) IS THE M3 LOAD
C  NLOAD     NUMBER OF TIME-VARYING LOADS
C  NRAN      NUMBER OF POINTS IN STRESS-TIME HISTORY
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  OVAL      OVALITY ANALYSIS ACCURACY FACTOR
C  P()       1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C  PSUBI     INTERNAL PRESSURE
C  PERIOD    LENGTH OF TIME IN SECONDS FOR RANDOM STRESS-TIME HISTORY
C  PI        CONSTANT EQUAL TO 3.14.....
C  PSTAT     STATIC AXIAL LOAD
C  RB        BEND RADIUS MEASURED FROM CENTER OF PIPE
C  RT()      1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C              CONCENTRATION DUE TO WELD ECCENTRICITY
C  SBND()    1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CORRECTED FOR PHI, KRATIO, AND LNZ
C  STATIC()  1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES -- STATIC(1)
C              IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS; STATIC(3)
C              IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C  STRAMP()  2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C              -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C              THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C              STRAMP(4,*) ARE THE SHEAR STRESSES
C  STRHIS()  2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING
C              STRESS-TIME HISTORIES
C  SZERO     STRESS TENSILE POINT (PSI)
C  T()       1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C  TIB       WALL THICKNESS AT INNER BEND
C  TOB       WALL THICKNESS AT PIPE OUTER BEND
C  TRUNC     VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
C  TSTAT     STATIC TORQUE LOAD
C  V()       2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS
C  VSTAT()   1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE
C              V2 LOAD; V(2) IS THE V3 LOAD
C  WD        WELD DISTANCE FROM ELBOW TANGENCY
C  WOFF      WELD OFFSET
C  ZROREG    Zero REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
```

```
C              BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
C              REGION

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

C        EXTERIOR SURFACE OF THE PIPE

         CALL 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)

         CALL NARBN1 (STRHIS, NRAN, PERIOD, TRUNC, STATIC, STRAMP,
     &                NLOAD, FTY, FTU, 1.0, MM, LNA, LPHIM, KRATIO,
     &                LNZ, SBND, SZERO, ZROREG, NUMREG, FATLIF)

         LIFE1 = FATLIF

C         IF (IOUT .EQ. 25) WRITE(8,*) 'FATLIF = ', FATLIF

      ENDIF

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

C        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 NARBN1 (STRHIS, NRAN, PERIOD, TRUNC, STATIC, STRAMP,
     &                NLOAD, FTY, FTU, 1.0, MM, LNA, LPHIM, KRATIO,
     &                LNZ, SBND, SZERO, ZROREG, NUMREG, FATLIF)

         LIFE2 = FATLIF

C         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
            CULPRT = 1
         ELSE IF (LIFE .EQ. LIFE2) THEN
            CULPRT = 2
         ELSE
            WRITE(8,*) 'ERROR:  CANNOT FIND CULPRIT LOCATION'
            CALL TRMNAT
         ENDIF

         ELWELD = LIFE

C         IF (IOUT .EQ. 25)
C    &       WRITE(8,*) 'LIFE = ', LIFE, ' CULPRT = ', CULPRT

      ELSE

         ELWELD = FATLIF
         CULPRT = LOCAT

      ENDIF
```

```
      RETURN
      END

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

C  SUBROUTINE M2L1 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
C  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)

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

C     IMPLICIT NONE

      COMMON    IOUT

      INTEGER   IOUT, J, MAXLD, NLOAD

      REAL      NU, PI

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

      REAL      ANGLE, AREA, ARGNB, ARGNB2, ARGTM3, B, C1PHI, CCY, CCZ,
     &          CLY, CLZ, DI, EM, FK(10), GCY, GCZ, GLY, GLZ, GNBI,
     &          GNBO, 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, ROVERI, ROT, 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


C                    LIST OF VARIABLES
C
C  ANGLE      ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --
C               GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
C  AREA       CROSS SECTION AREA OF PIPE WALL
C  ARGNB, ARGNB2, ARGTM3
C             INTERMEDIATE CALCULATION VARIABLES USED IN OVALITY EFFECT
C               CALCULATIONS
C  B          TORUS EFFECT
C  C1PHI      EQUAL TO COS(1*PHI)
C  CCY, CCZ, CLY, CLZ
C             STRESS CARRY OVER FACTORS
C  DI         INTERIOR DIAMETER
C  EM         ELASTIC MODULUS
C  FK()       1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C               CONCENTRATION DUE TO WELD ECCENTRICITY
C  GCY, GCZ, GLY, GLZ
C             OVALITY EFFECT COEFFICIENTS
C  GNBI, GNBO, GTMI, GTMO
C             COEFFICIENTS USED IN OVALITY EFFECT CALCULATIONS
C  IFK        INTERPOLATED VALUE OF Fk CORRESPONDING TO PARTICULAR VALUE
C               OF r/t
C  IOUT       OUTPUT DUMP CONTROLLER
C  J          CONTROLS DUE LOOP FOR EACH POINT IN RT() AND FK() DURING
C               INTERPOLATION
C  K()        FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR PIPE
C               EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR PIPE EXTERIOR
C               FOR HOOP DIRECTION; K(1,2) IS FOR PIPE INTERIOR FOR AXIAL
```

```
C               DIRECTION; K(2,2) IS FOR PIPE INTERIOR FOR HOOP DIRECTION
C   KOFF        STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
C   KT1         STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
C   KT2         STRESS CONCENTRATION FACTOR FOR HOOP STRESS
C   L           OVALITY EFFECT VARIABLE
C   L2          EQUAL TO L ** 2
C   LAMW        ACCURACY FACTOR OF Fk - r/t CURVE
C   M()         2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                 ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C   MAXLD       MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C   MIIB        MOMENT OF INERTIA BASED ON MAXIMUM WALL THICKNESS (INNER BEND)
C   MIOB        MOMENT OF INERTIA BASED ON MINIMUM WALL THICKNESS (OUTER BEND)
C   MSTAT()     1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE
C                 M2 LOAD; M(2) IS THE M3 LOAD
C   NLOAD       NUMBER OF TIME-VARYING LOADS
C   NU          POISSON'S RATIO
C   NU2         EQUAL TO NU ** 2
C   OVAL        OVALITY ANALYSIS ACCURACY FACTOR
C   P()         1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C   PI          SELF EXPLANATORY CONSTANT
C   PSI         OVALITY EFFECT VARIABLE
C   PSTAT       STATIC AXIAL LOAD
C   PSUBI       INTERNAL PRESSURE
C   QO          OVALITY EFFECT STRESS DECAY FACTOR
C   QT          TORUS EFFECT STRESS DECAY FACTOR
C   R           RADIUS OF INTEREST -- RO FOR EXTERIOR ANALYSIS; RI FOR
C                 INTERIOR ANALYSIS
C   RB          BEND RADIUS MEASURED FROM CENTER OF PIPE
C   RI          INTERIOR RADIUS
C   RM          MEAN RADIUS
C   RO          OUTER RADIUS
C   ROIB        OUTER RADIUS AT THE MAXIMUM WALL THICKNESS (INNER BEND)
C   ROOB        OUTER RADIUS AT THE MINIMUM WALL THICKNESS (OUTER BEND)
C   ROVERI      EQUAL TO THE MAXIMUM R / I
C   ROT         EQUAL TO r / t (R Over T)
C   RT()        1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C                 CONCENTRATION DUE TO WELD ECCENTRICITY
C   S1PHI       EQUAL TO SIN(1*PHI)
C   STATIC()    1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES -- STATIC(1)
C                 IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS; STATIC(3)
C                 IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C   STRAMP()    2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C                 -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C                 THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                 STRAMP(4,*) ARE THE SHEAR STRESSES
C   T()         1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C   TIB         WALL THICKNESS AT INNER BEND
C   TM          MEAN WALL THICKNESS
C   TOB         WALL THICKNESS AT PIPE OUTER BEND
C   TSTAT       STATIC TORQUE LOAD
C   V()         2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                 ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C   VSTAT()     1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2
C                 LOAD; V(2) IS THE V3 LOAD
C   WD          WELD DISTANCE FROM ELBOW TANGENCY
C   WOFF        WELD OFFSET
C   X1, X2, X3, X4
C               OVALITY EFFECT VARIABLES


C     CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO
C     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 = ROIB
      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)) * (ROT - RT(J-1))
     &            / (RT(J) - RT(J-1)) + FK(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,*) 'LAMW = ', LAMW, ' KOFF = ', KOFF
      ENDIF

C     CALCULATE THE CROSS-SECTIONAL AREA AND MOMENT OF INERTIA

      AREA = PI * (ROOB ** 2 - RI ** 2)
      MIIB = PI * (ROIB ** 4 - RI ** 4) / 4.0
      MIOB = PI * (ROOB ** 4 - RI ** 4) / 4.0
      ROVERI = MAX (ROIB / MIIB, ROOB / MIOB)

C     OBTAIN STRESS CONCENTRATION FACTORS AND RADII APPROPRIATE TO LOCATION

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

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*) 'AREA = ', AREA, 'ROVERI = ', ROVERI
         WRITE(8,*) 'MIIB = ', MIIB, ' MIOB = ', MIOB
         WRITE(8,*) 'K(1,1) = ', K(1,1), ' KT1 = ', KT1
         WRITE(8,*) 'K(2,1) = ', K(2,1), ' KT2 = ', KT2
      ENDIF

C     CALCULATE STRESS INCREASE DUE TO TORUS EFFECT

      S1PHI = SIN (ANGLE)
      C1PHI = COS (ANGLE)
      QT = 1.0 - (WD / RM)
      B = 1.0 + QT * ((2.0 * RB + RM * S1PHI)
     &            / (2.0 * (RB + RM * S1PHI)) - 1.0)

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

C     CALCULATE STRESS INCREASE DUE TO OVALITY EFFECT

      L = TM * RB / ((RM ** 2) * SQRT (1.0 - NU2))
      PSI = PSUBI * RB ** 2 / (EM * RM * TM)

      IF (L .LT. 0.16)
     &  WRITE(8,*) 'WARNING:  LAMBDA < .16 DURING OVALITY CALCULATIONS'

      L2 = L ** 2
      X1 = 5.0 + 6.0 * L2 + 24.0 * PSI
      X2 = 17.0 + 600.0 * L2 + 480.0 * PSI
      X3 = X1 * X2 - 6.25
      X4 = (1.0 - NU2) * (X3 - 4.5 * X2)

      ARGTM3 = 1.5 * X2 - 18.75
      ARGNB = L / X4
      ARGNB2 = 9.0 * X2

      GTMI = S1PHI + (ARGTM3 * SIN (3.0 * ANGLE)
     &       + 11.25 * SIN (5.0 * ANGLE)) / X4
      GTMO = C1PHI + (ARGTM3 * COS (3.0 * ANGLE)
     &       + 11.25 * COS (5.0 * ANGLE)) / X4
```

```
      GNBI = ARGNB * (ARGNB2 * COS (2.0 * ANGLE)
     &               + 225.0 * COS (4.0 * ANGLE))
      GNBO = ARGNB * (ARGNB2 * SIN (2.0 * ANGLE)
     &               + 225.0 * SIN (4.0 * ANGLE))

      QO = 1.0 - WD / (4.0 * RM)

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

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*) 'PSUBI = ', PSUBI, ' EM = ', EM
         WRITE(8,*) 'L = ', L, ' PSI = ', PSI
         WRITE(8,*) 'L2 = ', L2
         WRITE(8,*) 'X1 = ', X1, ' X2 = ', X2
         WRITE(8,*) 'X3 = ', X3, ' X4 = ', X4
         WRITE(8,*) 'ARGTM3 = ', ARGTM3
         WRITE(8,*) 'ARGNB = ', ARGNB, ' ARGNB2 = ', ARGNB2
         WRITE(8,*) 'GTMI = ', GTMI, ' GTMO = ', GTMO
         WRITE(8,*) 'GNBI = ', GNBI, ' GNBO = ', GNBO
         WRITE(8,*) 'QO = ', QO, ' OVAL = ', OVAL
         WRITE(8,*) 'GLZ = ', GLZ, ' GCZ = ', GCZ
         WRITE(8,*) 'GLY = ', GLY, ' GCY = ', GCY
      ENDIF

C     CALL CALCS to CALCulate the Stresses

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

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*) 'RI = ', RI, ' 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, ' GLY = ', GLY
         WRITE(8,*) '    GCZ, = ', GCZ, ' GCY = ', GCY
      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,
     &                 CCY, CCZ, CLY, CLZ, OVAL, FK, RT, ANGLE, STATIC,
     &                 STRAMP)

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

C     IMPLICIT NONE

      COMMON   IOUT

      INTEGER  IOUT, J, MAXLD, NLOAD
```

```
      REAL      NU, PI
      PARAMETER (MAXLD = 16, NU = 0.30, PI = 3.1415926536)
      REAL      ANGLE, AREA, ARGNB, ARGNB2, ARGTM3, B, C1PHI, CCY, CCZ,
     &          CLY, CLZ, DI, EM, FK(10), GCY, GCZ, GLY, GLZ, GNBI,
     &          GNBO, GTMI, GTMO, IFK, K(2, 2), KOFF, KT1, KT2, L, L2,
     &          LAMW, M(2, MAXLD), MI, MSTAT(2), NU2, OVAL, P(MAXLD),
     &          PSI, PSTAT, PSUBI, QO, QT, R, RB, RI, RM, RO, ROIB,
     &          ROOB, ROT, 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

C                    LIST OF VARIABLES
C
C  ANGLE        ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --
C                 GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
C  AREA         CROSS SECTION AREA OF PIPE WALL
C  ARGNB, ARGNB2, ARGTM3
C               INTERMEDIATE CALCULATION VARIABLES USED IN OVALITY EFFECT
C                 CALCULATIONS
C  B            TORUS EFFECT
C  C1PHI        EQUAL TO COS(1*PHI)
C  CCY, CCZ, CLY, CLZ
C               STRESS CARRY OVER FACTORS
C  DI           INTERIOR DIAMETER
C  EM           ELASTIC MODULUS
C  FK()         1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C                 CONCENTRATION DUE TO WELD ECCENTRICITY
C  GCY, GCZ, GLY, GLZ
C               OVALITY EFFECT COEFFICIENTS
C  GNBI, GNBO, GTMI, GTMO
C               COEFFICIENTS USED IN OVALITY EFFECT CALCULATIONS
C  IFK          INTERPOLATED VALUE OF Fk CORRESPONDING TO PARTICULAR VALUE
C                 OF r/t
C  IOUT         OUTPUT DUMP CONTROLLER
C  J            CONTROLS DUE LOOP FOR EACH POINT IN RT() AND FK() DURING
C                 INTERPOLATION
C  K()          FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR PIPE
C                 EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR PIPE EXTERIOR
C                 FOR HOOP DIRECTION; K(1,2) IS FOR PIPE INTERIOR FOR AXIAL
C                 DIRECTION; K(2,2) IS FOR PIPE INTERIOR FOR HOOP DIRECTION
C  KOFF         STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
C  KT1          STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
C  KT2          STRESS CONCENTRATION FACTOR FOR HOOP STRESS
C  L            OVALITY EFFECT VARIABLE
C  L2           EQUAL TO L ** 2
C  LAMW         ACCURACY FACTOR OF Fk - r/t CURVE
C  M()          2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                 ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C  MAXLD        MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C  MI           MOMENT OF INERTIA
C  MSTAT()      1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE
C                 M2 LOAD; M(2) IS THE M3 LOAD
C  NLOAD        NUMBER OF TIME-VARYING LOADS
C  NU           POISSON'S RATIO
C  NU2          EQUAL TO NU ** 2
C  OVAL         OVALITY ANALYSIS ACCURACY FACTOR
C  P()          1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C  PI           SELF EXPLANATORY CONSTANT
C  PSI          OVALITY EFFECT VARIABLE
C  PSTAT        STATIC AXIAL LOAD
C  PSUBI        INTERNAL PRESSURE
C  QO           OVALITY EFFECT STRESS DECAY FACTOR
C  QT           TORUS EFFECT STRESS DECAY FACTOR
C  R            RADIUS OF INTEREST -- RO FOR EXTERIOR ANALYSIS; RI FOR INTERIOR
C                 ANALYSIS
C  RB           BEND RADIUS MEASURED FROM CENTER OF PIPE
C  RI           INTERIOR RADIUS
C  RM           MEAN RADIUS
C  RO           OUTER RADIUS
C  ROIB         OUTER RADIUS AT THE MAXIMUM WALL THICKNESS (INNER BEND)
```

```
C  ROOB       OUTER RADIUS AT THE MINIMUM WALL THICKNESS (OUTER BEND)
C  ROT        EQUAL TO r / t (R Over T)
C  RT()       1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C               CONCENTRATION DUE TO WELD ECCENTRICITY
C  S1PHI      EQUAL TO SIN(1*PHI)
C  STATIC()   1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES -- STATIC(1)
C               IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS; STATIC(3)
C               IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C  STRAMP()   2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES --
C               STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE THE HOOP
C               STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES; STRAMP(4,*)
C               ARE THE SHEAR STRESSES
C  T()        1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C  TIB        WALL THICKNESS AT INNER BEND
C  TM         MEAN WALL THICKNESS
C  TOB        WALL THICKNESS AT PIPE OUTER BEND
C  TSTAT      STATIC TORQUE LOAD
C  V()        2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C               ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C  VSTAT()    1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE
C               V2 LOAD; V(2) IS THE V3 LOAD
C  WD         WELD DISTANCE FROM ELBOW TANGENCY
C  WOFF       WELD OFFSET
C  X1, X2, X3, X4
C             OVALITY EFFECT VARIABLES


C     CALCULATE KOFF, THE STRESS CONCENTRATION FACTOR DUE TO
C     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)) * (ROT - RT(J-1))
     &            / (RT(J) - RT(J-1)) + FK(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,*) 'LAMW = ', LAMW, ' KOFF = ', KOFF
      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
```

```
         WRITE(8,*) 'K(1,2) = ', K(1,2), ' KT1 = ', KT1
         WRITE(8,*) 'K(2,2) = ', K(2,2), ' KT2 = ', KT2
      ENDIF

C     CALCULATE STRESS INCREASE DUE TO TORUS EFFECT

      S1PHI = SIN (ANGLE)
      C1PHI = COS (ANGLE)
      QT = 1.0 - (WD / RM)
      B = 1.0 + QT * ((2.0 * RB + RM * S1PHI)
     &             / (2.0 * (RB + RM * S1PHI)) - 1.0)

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

C     CALCULATE STRESS INCREASE DUE TO OVALITY EFFECT

      L = TM * RB / ((RM ** 2) * SQRT (1.0 - NU2))
      PSI = PSUBI * RB ** 2 / (EM * RM * TM)

      IF (L .LT. 0.16)
     &  WRITE(8,*) 'WARNING:  LAMBDA < .16 DURING OVALITY CALCULATIONS'

      L2 = L ** 2
      X1 = 5.0 + 6.0 * L2 + 24.0 * PSI
      X2 = 17.0 + 600.0 * L2 + 480.0 * PSI
      X3 = X1 * X2 - 6.25
      X4 = (1.0 - NU2) * (X3 - 4.5 * X2)

      ARGTM3 = 1.5 * X2 - 18.75
      ARGNB = L / X4
      ARGNB2 = 9.0 * X2

      GTMI = S1PHI + (ARGTM3 * SIN (3.0 * ANGLE)
     &        + 11.25 * SIN (5.0 * ANGLE)) / X4
      GTMO = C1PHI + (ARGTM3 * COS (3.0 * ANGLE)
     &        + 11.25 * COS (5.0 * ANGLE)) / X4

      GNBI = ARGNB * (ARGNB2 * COS (2.0 * ANGLE)
     &               + 225.0 * COS (4.0 * ANGLE))
      GNBO = ARGNB * (ARGNB2 * SIN (2.0 * ANGLE)
     &               + 225.0 * SIN (4.0 * ANGLE))

      QO = 1.0 - WD / (4.0 * RM)

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

      IF (IOUT .EQ. 25) THEN
         WRITE(8,*) 'PSUBI = ', PSUBI, ' EM = ', EM
         WRITE(8,*) 'L = ', L, ' PSI = ', PSI
         WRITE(8,*) 'L2 = ', L2
         WRITE(8,*) 'X1 = ', X1, ' X2 = ', X2
         WRITE(8,*) 'X3 = ', X3, ' X4 = ', X4
         WRITE(8,*) 'ARGTM3 = ', ARGTM3
         WRITE(8,*) 'ARGNB = ', ARGNB, ' ARGNB2 = ', ARGNB2
         WRITE(8,*) 'GTMI = ', GTMI, ' GTMO = ', GTMO
         WRITE(8,*) 'GNBI = ', GNBI, ' GNBO = ', GNBO
         WRITE(8,*) 'QO = ', QO, ' OVAL = ', OVAL
         WRITE(8,*) 'GLZ = ', GLZ, ' GCZ = ', GCZ
         WRITE(8,*) 'GLY = ', GLY, ' GCY = ', GCY
      ENDIF

C     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
```

```
         WRITE(8,*) 'RI = ', RI, ' AREA = ', AREA
         WRITE(8,*) 'RO = ', RO, ' R = ', R
         WRITE(8,*) 'KT1 = ', KT1, ' KT2 = ', KT2
         WRITE(8,*) 'MI = ', MI, ' KOFF = ', KOFF
         WRITE(8,*) 'B = ', B, ' GLZ, = ', GLZ, ' GLY = ', GLY
         WRITE(8,*) '    GCZ, = ', GCZ, ' GCY = ', GCY
      ENDIF

      RETURN
      END


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


C  SUBROUTINE CALCS USES THE LOADS, STRESS CONCENTRATION FACTORS, AND
C  GEOMETRIC INFORMATION TO CALCULATE THE STRESS COMPONENTS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  26APR90
C     VERSION:  PIPE V8, V8.1, V8.2, V8.3

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

C   INPUTS:  PSTAT, MSTAT, TSTAT, VSTAT, NLOAD, P, M, T, V, PSUBI, RI,
C            AREA, RO, R, KT1, KT2, ROVERI, KOFF, B, GLZ, GLY, GCZ, GCY,
C            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, ARGB, ARG1, ARG2, B, GCY, GCZ, GLY,
     &         GLZ, KOFF, KT1, KT2, M(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)


C                  LIST OF VARIABLES
C
C  ANGLE       ANGLE PHI MEASURED COUNTER-CLOCKWISE FROM Z-DIRECTION --
C                GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
C  AREA        CROSS SECTION AREA OF PIPE WALL
C  ARGA        INTERMEDIATE CALCULATION VARIABLE
C  ARGB        INTERMEDIATE CALCULATION VARIABLE
C  ARG1        INTERMEDIATE CALCULATION VARIABLE
C  ARG2        INTERMEDIATE CALCULATION VARIABLE
C  B           TORUS EFFECT
C  GCY, GCZ,   GLY, GLZ
C              OVALITY EFFECT COEFFICIENTS
C  I           CONTROLS DO LOOP FOR THE TIME-VARYING STRESSES
C  IOUT        OUTPUT DUMP CONTROLLER
C  KOFF        STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
C  KT1         STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
C  KT2         STRESS CONCENTRATION FACTOR FOR HOOP STRESS
C  M()         2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C  MAXLD       MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C  MSTAT()     1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE M2
C                LOAD; M(2) IS THE M3 LOAD
C  NLOAD       NUMBER OF TIME-VARYING LOADS
C  P()         1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C  PI          SELF EXPLANATORY CONSTANT
C  PSTAT       STATIC AXIAL LOAD
```

```
C  PSUBI       INTERNAL PRESSURE
C  R           RADIUS OF INTEREST -- RO FOR EXTERIOR ANALYSIS; RI FOR
C                INTERIOR ANALYSIS
C  RI          INTERIOR RADIUS
C  RO          OUTER RADIUS
C  RO2         EQUAL TO Ro ** 2
C  ROVERI      EQUAL TO THE MAXIMUM R / I
C  R2          EQUAL TO R ** 2
C  STATIC()    1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES -- STATIC(1)
C                IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS; STATIC(3)
C                IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C  STRAMP()    2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES --
C                STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*)  ARE THE
C                HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                STRAMP(4,*) ARE THE SHEAR STRESSES
C  T()         1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C  TSTAT       STATIC TORQUE LOAD
C  V()         2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C              ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C  VSTAT()     1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2
C                LOAD; V(2) IS THE V3 LOAD


C     PERFORM INTERMEDIATE CALCULATIONS

      R2 = R ** 2
      RO2 = RO ** 2

      ARG1 = RI ** 2 / (RO2 - RI ** 2)
      ARG2 = (RO2 + R2) / R2

      ARGA = PSUBI * ARG1
      ARGB = ARG2 * ARGA

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

C     CALCULATE STATIC STRESS COMPONENTS

         STATIC(1) = KT1 * (PSTAT / AREA + (GLZ * MSTAT(2)
     &               + GLY * MSTAT(1)) * ROVERI + ARGA) * KOFF
         STATIC(2) = KT2 * (B * ARGB + (GCZ * MSTAT(2)
     &               + GCY * MSTAT(1)) * ROVERI)
         STATIC(3) = - ARGA * (RO2 - R2) / R2
         STATIC(4) = (TSTAT * ROVERI / 2.0) - 2.0 * (VSTAT(1)
     &               * COS (ANGLE) + VSTAT(2) * SIN (ANGLE)) / AREA

C     CALCULATE TIME-VARYING STRESS COMPONENTS

      DO 100 I = 1, NLOAD

         STRAMP(1,I) = KT1 * (P(I) / AREA + (GLZ * M(2,I)
     &                 + GLY * M(1,I)) * ROVERI) * KOFF
         STRAMP(2,I) = KT2 * (GCZ * M(2,I) + GCY * M(1,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,*) I, STRAMP(1,I), STRAMP(2,I), STRAMP(3,I),
     &                 STRAMP(4,I)
  300    CONTINUE
      ENDIF
```

```
      RETURN
      END

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

C  SUBROUTINE NARBN1 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:  3MAY90
C     VERSION:  1.5    (PIPE V8.1, V8.2, V8.3)

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

C       INPUTS:  STRHIS, M, PERIOD, TRUNC, STATIC, STRAMP, NLOAD, FTY,
C                FTU, KT, MM, LNA, LPHIM, KRATIO, LNZ, SBND, SZERO,
C                ZROREG, NUMREG
C      OUTPUTS:  FATLIF
C  SUBPROGRAMS:  RAINF1

C     IMPLICIT NONE

      COMMON    IOUT

      INTEGER  I, IOUT, J, M, MAXLD, MAXM, MAXREG, NLOAD, NUMREG,
     &         ZROREG

      PARAMETER (MAXLD = 16, MAXM = 24000, MAXREG = 3)

      REAL     FATLIF, FTY, FTU, KRATIO, KT, LNA(0:MAXREG), LNZ,
     &         LPHIM(0:MAXREG), MM(0:MAXREG), PERIOD, RAINF1,
     &         S(4, MAXM),  SBND(0:MAXREG), SEFF(MAXM), STATIC(4),
     &         STRAMP(4, MAXLD), STRHIS(MAXLD, MAXM), SZERO, TRUNC


C                    LIST OF VARIABLES
C
C  FATLIF     VALUE OF FATIGUE LIFE CALCULATED
C  FTY        YIELD STRENGTH
C  FTU        ULTIMATE STRENGTH
C  I          CONTROLS DO LOOP FOR RANDOM AND SUPERIMPOSED SINE LOADS
C  IOUT       OUTPUT DUMP CONTROLLER
C  J          CONTROLS DO LOOP FOR EACH POINT IN THE STRESS-TIME HISTORY
C  KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  KT         STRESS CONCENTRATION FACTOR
C  LNA()      1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C  LNZ        NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  LPHIM()    1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
C               PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C  M          NUMBER OF POINTS IN STRESS-TIME HISTORY
C  MAXLD      MAXIMUM NUMBER OF TIME-VARYING LOADS
C  MAXM       MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING VALUES FOR m FOR EACH REGION
C  NLOAD      NUMBER OF TIME-VARYING LOADS
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  PERIOD     TIME IN SECONDS FOR ONE PERIOD OF STRESS-TIME HISTORY
C  RAINF1     FUNCTION WHICH CALCULATES THE TIME TO FAILURE FOR A GIVEN
C               UNI-AXIAL STRESS-TIME HISTORY
C  S()        2-D ARRAY CONTAINING THE TOTAL COMPONENT STRESS-TIME HISTORIES
C  SEFF()     1-D ARRAY CONTAINING THE EFFECTIVE (OR UNI-AXIAL) STRESS-TIME
C               HISTORY RESULTING FROM THE COMBINATION OF STATIC, RANDOM, AND
C               SINUSOIDAL LOADS FOR ALL FOUR COMPONENTS
C  SBND()     1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C               CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C               REGION CORRECTED FOR PHI, KRATIO, AND LNZ
C  STATIC()   1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES -- STATIC(1)
C               IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS; STATIC(3)
C               IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
```

```
C  STRAMP()     2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES --
C                 STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE THE
C                 HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                 STRAMP(4,*) ARE THE SHEAR STRESSES
C  STRHIS()     2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING
C                 STRESS-TIME HISTORIES
C  SZERO        STRESS TENSILE POINT (PSI)
C  TRUNC        VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
C  ZROREG       Zero REgion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                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)
C           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))) * 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)

  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)
            WRITE(8,*) 'SEFF = ', SEFF(J)
  125    CONTINUE
      ENDIF

      FATLIF = RAINF1 (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
      END


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


C**********************************************************************
C
C  FUNCTION RAINF1 CALCULATES THE TIME (in seconds) TO FAILURE FOR
C  THE GIVEN UNI-AXIAL (OR EFFECTIVE) STRESS-TIME HISTORY
C
C  PROGRAMMER:  L. NEWLIN
C        DATE:  23MAY89
C     VERSION:  3.1+ (MATCHR V8.3, V8.4, V8.5)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
```

```
C  is acknowledged.
C*************************************************************************

      FUNCTION RAINF1 (SEFF, M, TRUNC, PERIOD, KT, FTU, FTY, MM, LNA,
     &                 LPHIM, KRATIO, LNZ, SBND, SZERO, ZROREG, NUMREG)

C  INPUTS:    SEFF, M, TRUNC, PERIOD, KT, FTU, FTY, MM, LNA, LPHIM,
C             KRATIO, LNZ, SBND, SZERO, ZROREG, NUMREG
C  OUTPUTS:   RAINF1

C     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, RAINF1,
     &         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


C                          LIST OF VARIABLES
C
C  RAINF1        CYCLES TO FAILURE FOR THE GIVEN STRESS LEVELS
C
C input variables:
C
C  SEFF(M)       EFFECTIVE STRESSES BEFORE FILTERING/RAINFLOW
C  M             TOTAL NUMBER OF STRESS DATA POINTS PER PERIOD
C  TRUNC         VALUE USED TO FILTER OUT NOISE
C  PERIOD        TIME IN SECONDS FOR ONE PERIOD
C  KT            FATIGUE CONCENTRATION FACTOR
C  FTU           ULTIMATE TENSILE STRENGTH (PSI)
C  FTY           YIELD TENSILE STRENGTH (PSI)
C
C
C intermediate variables:
C
C  SEFMAX        LARGEST EFFECTIVE STRESS
C  JMAX          INDEX (LOCATION) OF SEFMAX IN SEFF()
C  I,J,K         COUNTERS FOR VARIOUS DO LOOPS
C  SP(M+1)       RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
C  S(NEWTOT)     FILTERED EFFECTIVE STRESSES
C  NEWTOT        TOTAL NUMBER OF EFFECTIVE STRESS VALUES AFTER FILTERING
C  E()           HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
C  N             NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
C  SEFFM(2,N)    EFFECTIVE STRESSES AFTER RESEQUENCING/FILTERING/RAINFLOW
C                     SEFFM(1,I) = sigma max,eff,i
C                     SEFFM(2,I) = sigma min,eff,i
C  SALTF(N)      SALTF(I) = sigma alternating,eff,i
C  SMEANF(N)     SMEANF(I) = sigma mean,eff,i
C  BIG1          VALUE OF i FOR SEFMAX
C  SM            SM = EQUIVALENT MEAN STRESS
C  ARGM          INTERMEDIATE CALCULATION VARIABLE EQUAL TO KT/(1 - SM/FTU)
C  STR(N)        STR(I) = EQUIVALENT (COMBINED) STRESS,i
C  LIFE(N)       LIFE(I) = CALCULATED LIFE FOR STRESS LEVEL STR(I)
C  INVLIF(N)     INVLIF(I) = 1/LIFE(I); DAMAGE FRACTION
C  SUMDAM        SUM OF ALL THE DAMAGE FRACTIONS
C
C  IOUT          OUTPUT DUMP CONTROLLER
C  KRATIO        RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  LNA()         1-D ARRAY CONTAINING VALUES OF Ln(A) = M Ln K FOR EACH REGION
C  LNZ             NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  LPHIM()       1-D ARRAY CONTAINING VALUES OF M Ln PHI FOR EACH REGION WHERE
```

```
C                PHI IS A WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
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
      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,*)'lna(i),      mm(i),      lphim(i),      sbnd(i)'
        write(8,*) (lna(i), mm(i), lphim(i), sbnd(i), i=zroreg,numreg)
        write(8,*)' '
      endif

C  INITIALIZE ARRAYS

      DO 50 I = 1, MAXM
         SP(I) = 0.0
         S(I) = 0.0
         E(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********************* B E G I N    R E S E Q U E N C E **********************
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
      SP(M+1) = SEFF(JMAX)
      if (iout.eq.20) then
```

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

C*********************** E N D    R E S Q U E N C E ****************************
C************************ B E G I N     F I L T E R ****************************
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
C

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

C         if (iout .eq. 20) then
C            do 305 i = 2, m
C               write(8,*) 'test1 = ', test1(i), ' test2 = ', test2(i)
C   305      continue
C         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,*)'newtot:',newtot
            write(8,*)'s(newtot):',(s(i),i=1,newtot)
         endif

C**************************** E N D     F I L T E R ****************************
C*********************** B E G I N     R A I N F L O W ************************
C   RAINFLOW ANALYSIS to identify cycles within effective stress data,
C    S(NEWTOT); places each cycle's max and min values into SEFFM(2,N)
C
C    counters: I counts # of cycles found, J counts how many S()'s counted,
C              K accumulates unmatched points
         I = 0
         J = 0
         K = 0
  400    CONTINUE
         J = J+1
         K = K+1
C    check J to avoid reading beyond end of filtered stress data
         IF ( J .GT. NEWTOT ) GOTO 499

C    read stress point into a holding array to be checked for cycles
         E(K) = S(J)
  410    IF ( K .LT. 3 ) GOTO 400
         IF ( ABS( E(K) - E(K-1) ) .LT. ABS( E(K-1) - E(K-2) ) ) GOTO 400
C     if not, then a cycle has been found, but we need to check for truncation
         IF (ABS (E(K-1) - E(K-2)) .GT. TRUNC) THEN
C           cycle is large enough to save
            I = I+1
            SEFFM(1,I) = AMAX1( E(K-1), E(K-2) )
            SEFFM(2,I) = AMIN1( E(K-1), E(K-2) )
         ENDIF
C    discard points K-1 and K-2, and decrement the counter of unmatched points
```

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

  499 CONTINUE
C   N equals the final number of cycles found
      N = I

      if (iout.eq.20) then
         write(8,*)'N :',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
C*********************** E N D   R A I N F L O W ***************************

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) then
         write(8,*)'saltf(n) :',(saltf(i),i=1,n)
         write(8,*)'smeanf(n):',(smeanf(i),i=1,n)
      endif

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

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), MM, LNA, LPHIM, KRATIO, LNZ, SBND,
     &                     ZROREG, NUMREG, SZERO)
 600  CONTINUE
      DO 650 I=1,N
         INVLIF(I) = 1.0 / LIFE(I)
 650  CONTINUE
      if (iout.eq.20) then
         do 14 i=1,n
            write(8,*)'life(n):',life(i),'      invlif(n):',invlif(i)
 14      continue
```

```
      endif
C  Miner's Rule -- sum the damage fractions
C
      SUMDAM = 0.0
      DO 700 I=1,N
         SUMDAM = SUMDAM + INVLIF(I)
  700 CONTINUE
  710 CONTINUE
      if (iout.eq.20) write(8,*)'sumdam:',sumdam
C  calculate fatigue life (time to failure) 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 7-3 List of Subprograms for Program HEXHCF
(Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
ADDREG[1]	4.1.3.9	Adds the *m* ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.
ADDRGN[1]	4.1.3.15	Adds the *m* ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.
BETAGN[2]	4.4.5	Generates Beta(a, b, ρ, θ) random variates.
CONCAV[3]	4.1.3.10	Adjusts the upper bound of the posterior ranges on *m* to be consistent with concavity constraints.
CONVRT[4]	4.1.3.3	Transforms stress data to equivalent zero-mean stresses with stress ratio of -1.0.
EXPCTD[5]	4.1.3.12	Calculates the median S/N curve parameters from the results of the information aggregation calculations.
FINDK	4.1.5.6	Calculates the value of the location parameter K (where $A = K^m$) for each life region by using *Equations 2-37* and *2-41.*
FINDM[6]	4.1.5.1	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.
FINDMC	4.1.3.5	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.*
FINDMN[6]	4.1.5.2	Obtains the value of *m* for each life region by sampling from the appropriate truncated Normal distribution on *m*.
FINDSB	4.1.5.7	Calculates the life region "tie-points" or stress values which correspond to the "life boundaries" conditional on the randomly

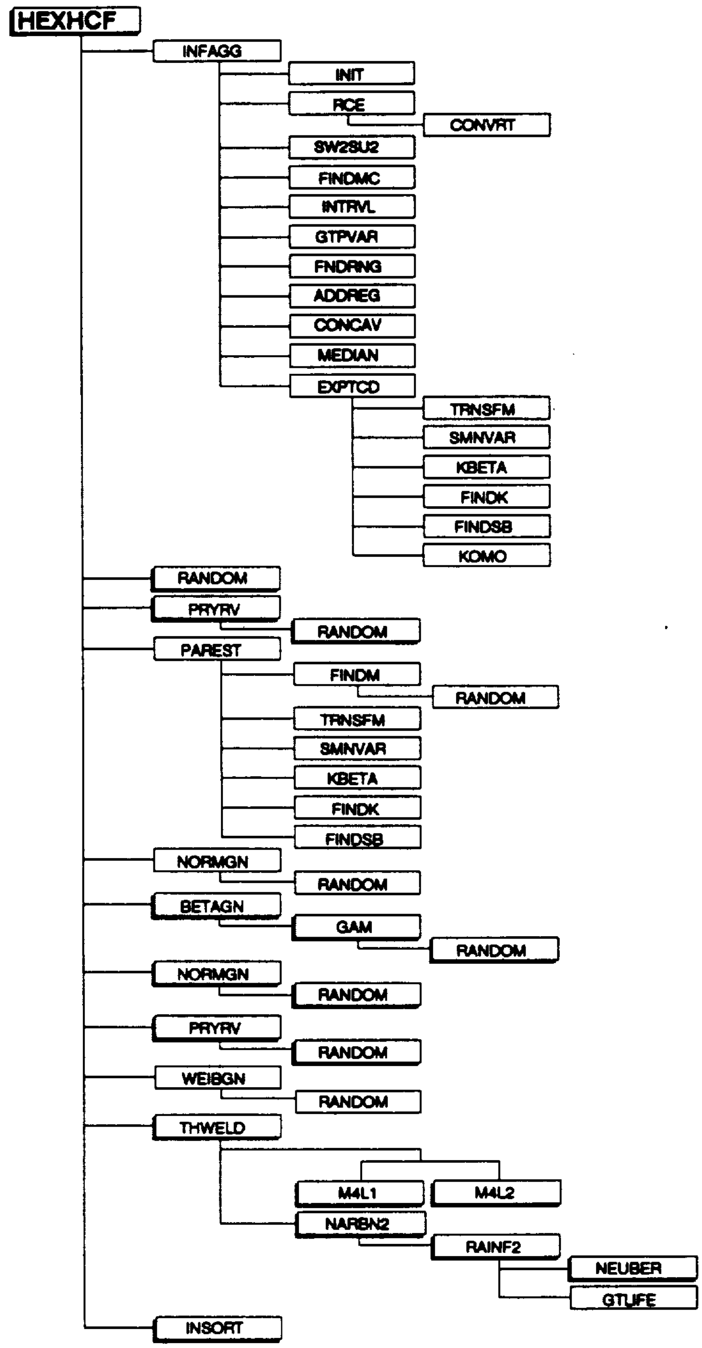

Figure 7-3 Tree Structure for Program HEXHCF for the Uniform Variation in Materials Shape Parameter *m*

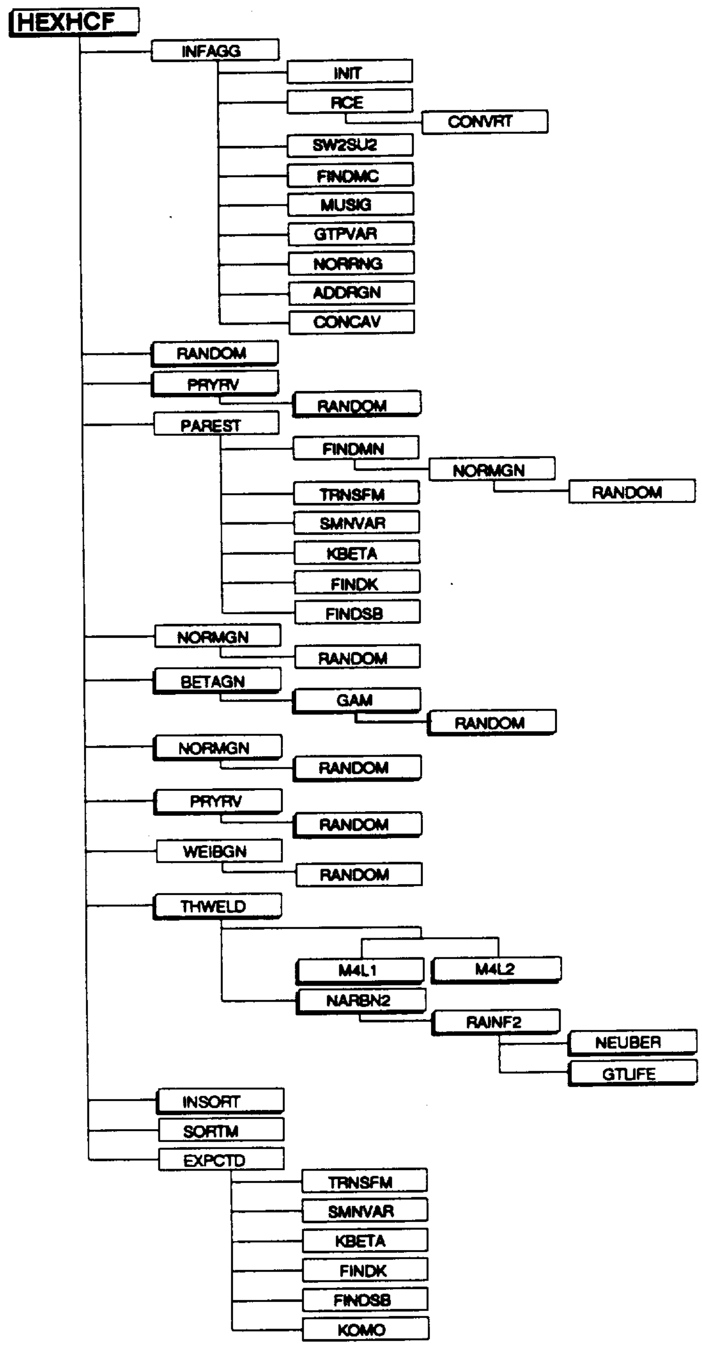

Figure 7-4 Tree Structure for Program HEXHCF for the Truncated Normal Variation in Materials Shape Parameter *m*

Table 7-3 List of Subprograms for Program HEXHCF (Cont'd)

NAME	SECTION	PURPOSE
		selected *m* for each region. Also calculates *K*, characterizing the specific material S/N data set, which is a function of β_o and *k*.
FNDRNG[7]	4.1.3.8	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.
GAM	4.4.4	Generates Gamma(α, 1) random variates.
GTLIFE	4.1.8	Calculates the cycles to failure for a particular stress based upon the materials characterization model S/N curve of *Equation 2-48.*
GTPVAR	4.1.3.7	Calculates σ^2, *Equation 2-49,* the extent of departures from the multiple heat median S/N curve warranted by the information available.
HEXHCF	5.1.3.1	The main routine that controls the logical flow of the high cycle fatigue plain welded duct program with thermal loads.
INFAGG[8]	4.1.3	Controls the logical flow for the information aggregation portion of the materials characterization model.
INIT	4.1.3.1	Initializes the entries of the arrays used in the information aggregation subroutine, INFAGG, to zero.
INSORT	5.B	Performs an insertion sort for the lowest fifty percent of the lives calculated.
INTRVL	4.1.3.6	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.*
KBETA	4.1.5.5	Calculates *k* and β_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.*
KOMO[9]	4.1.6	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.
M4L1	5.1.3.3	Performs the driver transformation, *Equations 2-68* through *2-73,* for location 1, the exterior surface of the duct.
M4L2	5.1.3.3	Performs the driver transformation, *Equations 2-68* through *2-73,* for location 2, the interior surface of the duct.
MEDIAN	4.1.3.11	Calculates the median values of *m* based on the posterior credibility ranges of *m* by using *Equation 2-34.*
MUSIG[10]	4.1.3.13	Calculates the posterior Normal distribution parameters: mean m_* and standard deviation σ_*, for each life region of the S/N curve.
NARBN2	5.1.3.4	Calculates the composite stress-time history by using *Equations 2-82* and *2-84* and then calls RAINF2 to calculate the fatigue life.

Table 7-3 List of Subprograms for Program HEXHCF (Cont'd)

NAME	SECTION	PURPOSE
NEUBER	5.1.3.6	Calculates the equivalent mean stress from the maximum stress based on Neuber's rule by using *Equations 2-88* and *2-89*.
NORMGN[11]	4.4.3	Generates Normal(μ, σ^2) random variates.
NORRNG[7]	4.1.3.14	Combines the implicit and explicit constraints on *m* to obtain the posterior credibility ranges of *m* for each life region.
PAREST[12]	4.1.5	Controls the logical flow for the parameter estimation model portion of the materials characterization model.
PRYRV[13]	7.6.6	Generates the Uniform(*a*, *b*) and Uniform(*c*, *d*) pair of independent random variates.
RAINF2[14]	5.1.3.5	Performs rainflow cycle counting, Miner's rule damage accumulation, and calls GTLIFE to calculate the fatigue life.
RANDOM[13]	4.4.2	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.
RCE	4.1.3.2	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.
SMNVAR	4.1.5.4	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*.
SORTM[15]	4.1.10	Sorts the *m* values in increasing order for each life region for the truncated Normal distribution case.
SW2SU2	4.1.3.4	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.
THWELD	5.1.3.2	Controls the logical flow for the driver transformation and fatigue life calculations.
TRMNAT	4.1.11	Performs premature program termination, when required.
TRNSFM[16]	4.1.5.3	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.
WEIBGN	4.4.6	Generates Weibull(β, $\eta(\beta)$) random variates.

---

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.*

7.1.2.3 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)

VARIABLE NAME	TYPE	DESCRIPTION
AERD	RE	$\lambda_{D_{AERO}}$ in *Equation 2-81*, the randomly selected load scale factor for the AERodynamic Dynamic load components.
AERDA	RE	Dynamic aerodynamic load scale factor Uniform distribution lower bound.
AERDB	RE	Dynamic aerodynamic load scale factor Uniform distribution upper bound.
AERS	RE	$\lambda_{ST_{AERO}}$ in *Equation 2-81*, the randomly selected load scale factor for the AERodynamic Static load components.
AERSA	RE	Static aerodynamic load scale factor Uniform distribution lower bound.
AERSB	RE	Static aerodynamic load scale factor Uniform distribution upper bound.
ALLM(MAXMM, MAXREG)	RE	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]
ALPHA	RE	α (/ °R) in *Equation 2-70*, the coefficient of thermal expansion. (see **COEXP** in HEXHCF)
ANGLE	RE	ϕ (rad) in *Equation 2-68*, the angle measured counterclockwise from Z-direction to the critical circumferential location.
AREA	RE	A (in.^{2}) in *Equation 2-68*, the cross-sectional area of the duct wall.
ASTR	RE	$\lambda_{AERO_{str}}$ in *Equation 2-81*, the randomly selected aerodynamic stress analysis accuracy factor.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
ASTRA	RE	Aerodynamic stress analysis accuracy factor Uniform distribution lower bound.
ASTRB	RE	Aerodynamic stress analysis accuracy factor Uniform distribution upper bound.
BIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter *K*, *Equation 2-12*, where $A = K^m$.
BIGK1	RE	Dummy variable used during calls to subroutine EXPCTD, equal to **BIGK(1)**.
BLFPER(MAXBLF)	RE	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%.
BLFPOS(MAXBLF)	INT	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( )**.
BZERO	RE	Estimate of Weibull distribution shape parameter β_0, *Equation 2-11*, which characterizes the intrinsic variation of the S/N data set.
COEXP	RE	α (/ °R) in *Equation 2-70*, the COefficient of thermal EXPansion. (see **ALPHA** in THWELD)
DI	RE	D_i (in.), the randomly selected duct inner diameter at the weld used to calculate R_i in *Equation 2-68*.
DIA	RE	D_i lower bound of Beta distribution.
DIB	RE	D_i upper bound of Beta distribution.
DIR	RE	Randomly selected Beta distribution location parameter ρ for D_i.
DIR1	RE	ρ Uniform distribution lower bound of Beta distribution of D_i.
DIR2	RE	ρ Uniform distribution upper bound of Beta distribution of D_i.
DIT	RE	Randomly selected Beta distribution shape parameter θ for D_i.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
DIT1	RE	θ Uniform distribution lower bound of Beta distribution of D_i.
DIT2	RE	θ Uniform distribution upper bound of Beta distribution of D_i.
DLTAT	RE	DeLTA T. ΔT (°R) in *Equation 2-70*, the temperature difference across the wall of the duct.
DPCMU	RE	Value of (**PCMUB – PCMUA**).
DPCSIG	RE	Value of (**PCSIGB – PCSIGA**).
DSTR	RE	λDYN_{str} in *Equation 2-81*, the randomly selected dynamic stress analysis accuracy factor.
DSTRA	RE	Dynamic stress analysis accuracy factor Uniform distribution lower bound.
DSTRB	RE	Dynamic stress analysis accuracy factor Uniform distribution upper bound.
DTIMU	RE	Value of (**TIMUB – TIMUA**).
DTISIG	RE	Value of (**TISIGB – TISIGA**).
DTOMU	RE	Value of (**TOMUB – TOMUA**).
DTOSIG	RE	Value of (**TOSIGB – TOSIGA**).
DUM	RE	Dummy variable
E(MAXSEG)	RE	1-D array containing the strain ε values for the stress/strain versus strain curve.[2]
EM	RE	E (psi) in *Equation 2-70*, Young's modulus of elasticity for the material.
FATLIF	RE	Value of FATIgue LIFe calculated (sec).
FIFTY	RE	Variable used to access the fifty-percent point in the **LIFE( )** array.
FILNUM(MAXLD)	INT	1-D array containing the file unit numbers for the reference time history files.
FK(10)	RE	1-D array containing values of F_k, *Equation 2-73*, used to find stress concentration due to weld eccentricity, K_{OFF}.
FTEST	LOG	File TEST. Used to test for the existence of a reference time history file before attempting to open it.
FTU	RE	Material ultimate strength (psi).
FTY	RE	Material yield strength (psi).

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
GAM	RE	λ_{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.
GAMA	RE	Damage accumulation model accuracy factor Uniform distribution lower bound.
GAMB	RE	Damage accumulation model accuracy factor Uniform distribution upper bound.
I	INT	Controls inner DO loop.
II	INT	Controls DO loop for narrow-band random, superimposed sinusoidal, and dynamic aerodynamic loads.
IOUT	INT	Output dump controller.
J	INT	Controls DO loop for each B-life.[3]
K	INT	Controls outer DO loop.
K(2, 2)	RE	2-D array containing the fatigue stress concentration factors required for the stress analysis. **K(1,*)** is K_{T1} in *Equation 2-68* and **K(2,*)** is K_{T2} in *Equation 2-69*. **K(1,1)** is the outer diameter axial stress concentration factor, the value of **KGOD * KWOD**; **K(1,2)** is the inner diameter axial stress concentration factor, the value of **KGID * 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)
KGID	RE	Axial stress concentration factor due to geometry for the duct inner diameter used to calculate K_{T1} in *Equation 2-68.*
KGOD	RE	Axial stress concentration factor due to geometry for the duct outer diameter used to calculate K_{T1} in *Equation 2-68.*
KOFF	RE	K_{OFF} in *Equation 2-73,* the stress concentration factor due to eccentricity of the weld.
KRATIO	RE	Ratio of *MED K*/MED K* in *Equation 2-48.* **KRATIO** is constant over life regions for the materials model.
KT(2, 2)	RE	2-D array containing the fatigue stress concentration factors required for the stress analysis. **KT(1,*)** is K_{T1} in *Equation 2-68* and **KT(2,*)** is K_{T2}

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
		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)
KWID	RE	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.*
KWIDA	RE	Inner diameter weld axial stress concentration factor lower bound of Beta distribution.
KWIDB	RE	Inner diameter weld axial stress concentration factor upper bound of Beta distribution.
KWIDR	RE	Randomly selected Beta distribution location ρ for the inner diameter weld axial stress concentration factor.
KWIDR1	RE	ρ Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWIDR2	RE	ρ Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWIDT	RE	Randomly selected Beta distribution shape parameter θ for the inner diameter weld axial stress concentration factor.
KWIDT1	RE	θ Uniform distribution lower bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWIDT2	RE	θ Uniform distribution upper bound of Beta distribution of the inner diameter weld axial stress concentration factor.
KWOD	RE	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.*
KWODA	RE	Outer diameter weld axial stress concentration factor lower bound of Beta distribution.
KWODB	RE	Outer diameter weld axial stress concentration factor upper bound of Beta distribution.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
KWODR	RE	Randomly selected Beta distribution location parameter ρ for the outer diameter weld axial stress concentration factor.
KWODR1	RE	ρ Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.
KWODR2	RE	ρ Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.
KWODT	RE	Randomly selected Beta distribution shape parameter θ of the outer diameter weld axial stress concentration factor.
KWODT1	RE	θ Uniform distribution lower bound of Beta distribution of the outer diameter weld axial stress concentration factor.
KWODT2	RE	θ Uniform distribution upper bound of Beta distribution of the outer diameter weld axial stress concentration factor.
L	INT	Controls DO loop for each life region of the S/N curve.
LAMN	RE	$\lambda_{D_{RANDOM}}$ 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*.
LAMNA	RE	Lower bound of the Uniform distribution of *k* for the narrow-band random load scale factor.
LAMNB	RE	Upper bound of the Uniform distribution of *k* for the narrow-band random load scale factor.
LAMNC	RE	Coefficient of variation *C* for the narrow-band random load scale factor.
LAMND	RE	Strain gage correction factor *d* for the narrow-band random load scale factor.
LAMNK	RE	Randomly selected *k* for the narrow-band random load scale factor.
LAMNMU	RE	The resulting mean μ of the Normal distribution for the narrow-band random load scale factor, where $\mu = d/(1 + kC)$.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LAMNSG	RE	The resulting standard deviation σ of the Normal distribution for the narrow-band random load scale factor, where $\sigma = C/(1 + kC)$.
LAMS	RE	$\lambda_{D_{SINUSOIDAL}}$ 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.
LAMSA	RE	Lower bound of the Uniform distribution of k for the superimposed sinusoidal load scale factor.
LAMSB	RE	Upper bound of the Uniform distribution of k for the superimposed sinusoidal load scale factor.
LAMSC	RE	Coefficient of variation C for the superimposed sinusoidal load scale factor.
LAMSD	RE	Strain gage correction factor d for the superimposed sinusoidal load scale factor.
LAMSK	RE	Randomly selected k for the superimposed sinusoidal load scale factor.
LAMSMU	RE	The resulting mean μ of the Normal distribution for the superimposed sinusoidal load scale factor, where $\mu = d/(1 + kC)$.
LAMSSG	RE	The resulting standard deviation σ of the Normal distribution for the superimposed sinusoidal load scale factor, where $\sigma = C/(1 + kC)$.
LAMW	RE	LAMbda Weld offset, the randomly selected λ_{OFF} in *Equation 2-73*, the accuracy factor for the weld offset eccentricity stress concentration factor, K_{OFF}.
LAMWA	RE	λ_{OFF} Uniform distribution lower bound.
LAMWB	RE	λ_{OFF} Uniform distribution upper bound.
LDNAME(MAXLD)	CH6	1-D array containing LoaD NAMEs for the dynamic or time-varying loads. These are the names of the reference time history files.
LIFE(MAXLIF)	RE	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.
LNA(0:MAXREG)	RE	1-D array containing values of $\ln(A) = \ln(\text{BIGK}) * \text{MM}$ for each life region of the S/N curve.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LNZ	RE	ln(Z) in *Equation 2-48*, the Normal(0, **PVAR**) random variate for the materials process variation aspect of the materials model.
LOCAT	INT	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.
LPHIM(0:MAXREG)	RE	1-D array containing values of ln(PHI) * **MM** for each life region of the S/N curve.
M(2, MAXLD)	RE	2-D array containing the dynamic or time-varying moment load components. **M(1,*)** is M_y (in.-lb) in *Equation 2-68*, the moment load components about the y axis; and **M(2,*)** is M_z (in.-lbs) in *Equation 2-68*, the moment load components about the z axis.
MAXBLF	INT	Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.[3]
MAXDAT	INT	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.
MAXLD	INT	Maximum number of dynamic or time-varying loads allowed. The maximum number of loads is 16.
MAXLIF	INT	Maximum number of fatigue lives allowed for the simulated failure distribution. The maximum number of fatigue lives to be saved is 10,000.
MAXM	INT	Maximum number of points allowed in the time history arrays. The maximum number of points is 24,000.
MAXMM	INT	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.
MAXREG	INT	Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.
MAXSEG	INT	Maximum number of segments allowed in the stress/strain versus strain curve. The maximum number of segments is 10.[2]
MCOUNT	INT	Counts number of *m*'s to be used to calculate median S/N curve for the truncated Normal distribution case.[1]

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
MEDM(MAXMM)	RE	1-D array containing the empirical median *m* for each life region of the S/N curve.[4]
MI	RE	*I* (in.^{4}) in *Equation 2-68,* the cross-sectional Moment of Inertia.
MID	INT	Pointer to the median *m* values in array **SORTM( )** for the truncated Normal median S/N curve. Value of half of **MCOUNT**.
MLAM(2, MAXLD)	RE	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,*)** is M_y (in.-lbs) in *Equation 2-68*, the moment load components about the y axis; and **MLAM(2,*)** is M_z (in.-lbs) in *Equation 2-68*, the moment load components about the z axis.
MM(0:MAXREG)	RE	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.
MPROC	INT	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]
MSLAM(2)	RE	1-D array containing the static moment load components scaled by **ASTR** and **AERS**. **MSLAM(1)** is M_y (in.-lbs) in *Equation 2-68,* the moment load component about the y axis; and **MSLAM(2)** is M_z (in.-lbs) in *Equation 2-68*, the moment load component about the z axis.
MSTAT(2)	RE	1-D array containing the static moment load components. **MSTAT(1)** is M_y (in.-lbs) in *Equation 2-68*, the moment load component about the y axis; and **MSTAT(2)** is M_z (in.-lbs) in *Equation 2-68*, the moment load component about the z axis.
MU(MAXREG)	RE	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.
NBLIFE	INT	Number of B-lives to be obtained from the simulated failure distribution.[3]

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NBND(0:MAXREG)	RE	$N^*_{i,i+1}$ 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.
NEUB	RE	λ_{neu} in *Equation 2-89*, the randomly selected Neuber's rule model accuracy factor.
NEUBA	RE	Neuber's rule model accuracy factor Uniform distribution lower bound.
NEUBB	RE	Neuber's rule model accuracy factor Uniform distribution upper bound.
NEWLIF	RE	Fatigue life value (sec) returned from call to function THWELD.
NF(MAXDAT, MAXREG)	RE	2-D array containing values from the array **RAWNF( )** for the specific material S/N data set partitioned into life regions.
NHYPER	INT	The outer loop size.
NLIFE	INT	The inner loop size.
NLIFET	INT	Total number of lives calculated by program HEXHCF. Value of **NHYPER * NLIFE**.
NLOAD	INT	NLOAD in *Equation 2-81*, the number of dynamic or time-varying loads.
NMED	INT	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.
NORM	RE	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 μ and σ 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.
NPTS(MAXREG)	INT	1-D array containing the number of points per life region for the specific material S/N data set.
NRAN	INT	Number of RANdom points. Number of points in the reference time history.
NU	RE	ν in *Equation 2-70*, the materials Poisson's ratio.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NUMREG	INT	R in *Equation 2-11*, the number of life regions of interest in the S/N curve.
NUMSEG	INT	Number of segments of interest in stress/strain versus strain curve.[2]
P(MAXLD)	RE	1-D array containing P (lbs) in *Equation 2-68*, the dynamic or time-varying axial load components.
PC	RE	p_i (psi) in *Equation 2-68*, the randomly selected internal pressure.
PCMU	RE	Randomly selected Normal distribution parameter μ for the internal pressure p_i.
PCMUA	RE	μ Uniform distribution lower bound of Normal distribution of the internal pressure p_i.
PCMUB	RE	μ Uniform distribution upper bound of Normal distribution of the internal pressure p_i.
PCO	RE	p_o (psi) in *Equation 2-68*, the external pressure.
PCSIG	RE	Randomly selected Normal distribution parameter σ for the internal pressure p_i.
PCSIGA	RE	σ Uniform distribution lower bound of Normal distribution of the internal pressure p_i.
PCSIGB	RE	σ Uniform distribution upper bound of Normal distribution of the internal pressure p_i.
PERIOD	RE	T (sec) in *Equation 2-91*, the length of time in seconds of the reference time history.
PHI	RE	φ in *Equation 2-11*, the materials intrinsic variation, or scatter, given by a Weibull(β_o, $\eta_o(\beta_o)$) random variate.
PI	RE	π, constant equal to 3.1415926536.
PLAM(MAXLD)	RE	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( )**.
PSIG	RE	σ in *Equation 2-48*, the value of SQRT(**PVAR**).
PSLAM	RE	P (lbs) in *Equation 2-68*, the static axial load component scaled by **ASTR** and **AERS**.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
PSTAT	RE	P (lbs) in *Equation 2-68*, the static axial load component.
PVAR	RE	σ^2 in *Equation 2-48*, characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.
RAINF2	RE	Real function which performs rainflow cycle counting, Miner's Rule damage accumulation, and calls GTLIFE to calculate the fatigue life.
RAND	DRE	Random number seed.
RANGEM(2, MAXREG)	RE	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.[7]
RI	RE	R_i (in.) in *Equation 2-68*, the duct inner radius.
RO	RE	R_o (in.) in *Equation 2-68*, the duct outer radius.
ROT	RE	R Over T, the value of the ratio R/t.
RT(10)	RE	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}.
S(4, MAXM)	RE	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,*)** is the axial stress-time history $\sigma_1(t)$; **S(2,*)** is the hoop stress-time history $\sigma_2(t)$; **S(3,*)** is the radial stress-time history $\sigma_3(t)$; and **S(4,*)** is the shear stress-time history $\sigma_4(t)$.
SBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each life region of the S/N curve contained in array **NBND( )**.
SE(MAXSEG)	RE	1-D array containing values of the product of stress and strain $\sigma\varepsilon$ for each segment of the stress/strain versus strain curve.[2]

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
SEFF(MAXM)	RE	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.
SIG(MAXREG)	RE	1-D array containing the posterior Normal distribution standard deviation[8] of the materials model shape parameter m, for each life region of the truncated Normal S/N curve.
SKT1	RE	K_{T1} in *Equation 2-68*, the stress concentration factor for the axial stress.
SKT2	RE	K_{T2} in *Equation 2-69*, the stress concentration factor for the hoop stress.
STATIC(4)	RE	1-D array containing values of the static stresses σ_{STk} (psi), *Equation 2-82*. **STATIC(1)** is the axial stress σ_{ST1}; **STATIC(2)** is the hoop stress σ_{ST2}; **STATIC(3)** is the radial stress σ_{ST3}; and **STATIC(4)** is the shear stress σ_{ST4}.
STR(MAXDAT, MAXREG)	RE	2-D array containing stress points with stress ratio = −1.0, for the specific material S/N data set partitioned into life regions.
STRAMP(4, MAXLD)	RE	2-D array containing values of the amplitudes of the dynamic or time-varying stresses $\overline{\sigma_{Dki}}$ (psi), *Equation 2-82*. **STRAMP(1,I)** is $\overline{\sigma_{D1i}}$, the amplitude of the i^{th} axial stress; **STRAMP(2,I)** is $\overline{\sigma_{D2i}}$, the amplitude of the i^{th} hoop stress; **STRAMP(3,I)** is $\overline{\sigma_{D3i}}$, the amplitude of the i^{th} radial stress; and **STRAMP(4,I)** is $\overline{\sigma_{D4i}}$, the amplitude of the i^{th} shear stress.
STRHIS(MAXLD, MAXM)	RE	2-D array containing $\sigma_i(t)$, *Equation 2-82*, the reference time histories for the dynamic or time-varying load components.
SZERO	RE	Stress tensile test point, S_o (psi).[9]
T(MAXLD)	RE	1-D array containing M_x (in.-lbs) in *Equation 2-72*, the dynamic or time-varying torsional load components.
TEST	RE	Uniform(0, 1) random variate used to determine Beta distribution for W_{OFF}.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
THIC	RE	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.*
THICA	RE	t lower bound of Beta distribution.
THICB	RE	t upper bound of Beta distribution.
THICR	RE	Randomly selected Beta distribution location parameter ρ for the wall thickness t.
THICR1	RE	ρ Uniform distribution lower bound of Beta distribution of t.
THICR2	RE	ρ Uniform distribution upper bound of Beta distribution of t.
THICT	RE	Randomly selected Beta distribution location parameter θ for the wall thickness t.
THICT1	RE	θ Uniform distribution lower bound of Beta distribution of t.
THICT2	RE	θ Uniform distribution upper bound of Beta distribution of t.
THWELD	RE	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).
TIN	RE	T_i (°R) the randomly selected inner wall surface temperature, used to calculate ΔT (°R), in *Equation 2-70*, the temperature difference across the wall of the duct.
TIMU	RE	Randomly selected Normal distribution parameter μ for the inner wall surface temperature T_i.
TIMUA	RE	μ Uniform distribution lower bound of Normal distribution of the inner wall surface temperature T_i.
TIMUB	RE	μ Uniform distribution upper bound of Normal distribution of the inner wall surface temperature T_i.
TISIG	RE	Randomly selected Normal distribution parameter σ for the inner wall surface temperature T_i.
TISIGA	RE	σ Uniform distribution lower bound of Normal distribution of the inner wall surface temperature T_i.
TISIGB	RE	σ Uniform distribution upper bound of Normal distribution of the inner wall surface temperature T_i.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
TLAM(MAXLD)	RE	1-D array containing M_x (in.-lbs) 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( )**.
TOUT	RE	T_o (°R) the randomly selected outer wall surface temperature, used to calculate ΔT (°R), in *Equation 2-70,* the temperature difference across the wall of the duct.
TOMU	RE	Randomly selected Normal distribution parameter μ for the outer wall surface temperature T_o.
TOMUA	RE	μ Uniform distribution lower bound of Normal distribution of the outer wall surface temperature T_o.
TOMUB	RE	μ Uniform distribution upper bound of Normal distribution of the outer wall surface temperature T_o.
TOSIG	RE	Randomly selected Normal distribution parameter σ for the outer wall surface temperature T_o.
TOSIGA	RE	σ Uniform distribution lower bound of Normal distribution of the outer wall surface temperature T_o.
TOSIGB	RE	σ Uniform distribution upper bound of Normal distribution of the outer wall surface temperature T_o.
TRSBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each region of the S/N curve contained in array **NBND( )** for each **PHI** draw consistent with the tensile point S_o.[9]
TRUNC	RE	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.
TSLAM	RE	M_x (in.-lbs) in *Equation 2-72,* the static torsional load component scaled by **ASTR** and **AERS**.
TSTAT	RE	M_x (in.-lbs) in *Equation 2-72,* the static torsional load component.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
TYPE(MAXLD)	INT	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; TYPE(*) = 2, use the superimposed sinusoidal load scale factor; and TYPE(*) = 3, use the dynamic aerodynamic load factor.
V(2, MAXLD)	RE	2-D array containing the dynamic or time-varying shear load components. V(1,*) is V_y (lbs) in *Equation 2-72,* the shear load components along the y axis; and V(2,*) is V_z (lbs) in *Equation 2-72,* the shear load components along the z axis.
VARY	INT	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.
VLAM(2, MAXLD)	RE	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,*) is V_y (lbs) in *Equation 2-72,* the shear load components along the y axis; and VLAM(2,*) is V_z (lbs) in *Equation 2-72,* the shear load components along the z axis.
VSLAM(2)	RE	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.
VSTAT(2)	RE	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.
WOFF	RE	W_{OFF} in *Equation 2-73,* the randomly selected Weld OFFset (%).
WOFFA	RE	W_{OFF} lower bound of Beta distribution 1.
WOFFB	RE	W_{OFF} upper bound of Beta distribution 1.
WOFFC	RE	W_{OFF} lower bound of Beta distribution 2.

Table 7-4 List of Variables for Program HEXHCF (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
WOFFD	RE	W_{OFF} upper bound of Beta distribution 2.
WOFFE	RE	Decimal equivalent percentage weight occurring in Beta distribution 1 of the weld offset W_{OFF}.
WOFFHI	RE	Upper bound of the randomly selected Beta distribution for the weld offset W_{OFF}.
WOFFLO	RE	Lower bound of the randomly selected Beta distribution for the weld offset W_{OFF}.
WOFFR	RE	Randomly selected Beta distribution location parameter ρ for the weld offset W_{OFF}.
WOFFR1	RE	ρ Uniform distribution lower bound of Beta distribution 1 of W_{OFF}.
WOFFR2	RE	ρ Uniform distribution upper bound of Beta distribution 1 of W_{OFF}.
WOFFR3	RE	ρ Uniform distribution lower bound of Beta distribution 2 of W_{OFF}.
WOFFR4	RE	ρ Uniform distribution upper bound of Beta distribution 2 of W_{OFF}.
WOFFT	RE	Randomly selected Beta distribution shape parameter θ for the weld offset W_{OFF}.
WOFFT1	RE	θ Uniform distribution lower bound of Beta distribution 1 of W_{OFF}.
WOFFT2	RE	θ Uniform distribution upper bound of Beta distribution 1 of W_{OFF}.
WOFFT3	RE	θ Uniform distribution lower bound of Beta distribution 2 of W_{OFF}.
WOFFT4	RE	θ Uniform distribution upper bound of Beta distribution 2 of W_{OFF}.
Z	RE	Z in *Equation 2-48,* the randomly selected process variation shift factor given by a Lognormal(0,**PVAR**) random variate.
ZROREG	INT	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**.[9]

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_* 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 σ_* 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

HEXHCF Version 4.2

Program HEXHCF Listing Temporal Order, Uniform Distribution

Routine	Page
HEXHCF	7-151
INFAGG	7-168
INIT	7-173
RCE	7-175
CONVRT	7-181
SW2SU2	7-182
FINDMC	7-188
INTRVL	7-185
GTPVAR	7-190
FNDRNG	7-191
ADDREG	7-195
CONCAV	7-196
MEDIAN	7-197
EXPCTD	7-199
TRNSFM	7-213
SMNVAR	7-214
KBETA	7-215
FINDK	7-216
FINDSB	7-217
KOMO	7-219
RANDOM	7-210
PRYRV	7-166
RANDOM	7-210
PAREST	7-206
FINDM	7-208
RANDOM	7-210
TRNSFM	7-213
SMNVAR	7-214
KBETA	7-215
FINDK	7-216
FINDSB	7-217
NORMGN	7-212
RANDOM	7-210
BETAGN	7-167
GAM	7-167
RANDOM	7-210
NORMGN	7-212
RANDOM	7-210
PRYRV	7-166
RANDOM	7-210
WEIBGN	7-219
RANDOM	7-210
THWELD	7-223
M4L1	7-225

Routine	Page

Program HEXHCF Listing Temporal Order, Truncated Normal Distribution

```
C***********************************************************************
C  PROGRAM HEXHCF CONTROLS THE FLOW OF LOGIC OF THE HIGH CYCLE
C  FATIGUE PLAIN WELDED DUCT PROBLEM UNDER THERMAL LOADS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  6SEP91
C     VERSION:  4.2 -- (MATCHR V8.5, THDUCT V4.1, INSORT V2.1)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      PROGRAM HEXHCF

C  SUBPROGRAMS:  INFAGG, PAREST, NORMGN, PRYRV, BETAGN, WEIBGN, THWELD,
C                TRMNAT, INSORT, SORTM, EXPCTD
C
C        FILES:  1:HEXHCD-OLD; 3:HEXHCO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
C                7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
C                11-26:user named-OLD
C
C                NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

C     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, MCOUNT, MID, MPROC, NBLIFE, NHYPER, NLIFE,
     &         NLIFET, NLOAD, NMED, NPTS(MAXREG), NRAN, NUMREG, NUMSEG,
     &         TYPE(MAXLD), VARY, ZROREG

      DOUBLE PRECISION  RAND

      REAL     AERD, AERDA, AERDB, AERS, AERSA, AERSB,
     &         ALLM(MAXMM, MAXREG), ANGLE, ASTR, ASTRA, ASTRB,
     &         BIGK(0:MAXREG), BIGK1, BLFPER(MAXBLF), BZERO, COEXP,
     &         DI, DIA, DIB, DIR, DIR1, DIR2, DIT, DIT1, DIT2, DLTAT,
     &         DPCMU, DPCSIG, DSTR, DSTRA, DSTRB, DTIMU, DTISIG, DTOMU,
     &         DTOSIG, DUM, E(MAXSEG), EM, FIFTY, FK(10), FTU, FTY,
     &         GAM, GAMA, GAMB, KGID, KGOD, KRATIO, KT(2, 2), KWID,
     &         KWIDA, KWIDB, KWIDR, KWIDR1, KWIDR2, KWIDT, KWIDT1,
     &         KWIDT2, KWOD, KWODA, KWODB, KWODR, KWODR1, KWODR2,
     &         KWODT, KWODT1, KWODT2, LAMN, LAMNA, LAMNB, LAMNC, LAMND,
     &         LAMNK, LAMNMU, LAMNSG, LAMS, LAMSA, LAMSB, LAMSC, LAMSD,
     &         LAMSK, LAMSMU, LAMSSG, LAMW, LAMWA, LAMWB, LIFE(MAXLIF),
     &         LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG)

      REAL     M(2, MAXLD), MEDM(MAXREG), MLAM(2, MAXLD), MM(0:MAXREG),
     &         MSLAM(2), MSTAT(2), MU(MAXREG), NBND(0:MAXREG), NEUB,
     &         NEUBA, NEUBB, NEWLIF, NF(MAXDAT, MAXREG), NORM, NU,
     &         P(MAXLD), PC, PCMU, PCMUA, PCMUB, PCO, PCSIG, PCSIGA,
     &         PCSIGB, PERIOD, PHI, PLAM(MAXLD), PSIG, PSLAM, PSTAT,
     &         PVAR, RANGEM(2, MAXREG), RT(10), SBND(0:MAXREG),
     &         SE(MAXSEG), SIG(MAXREG), STR(MAXDAT, MAXREG),
     &         STRHIS(MAXLD, MAXM), SZERO, T(MAXLD), TEST, THIC, THICA,
     &         THICB, THICR, THICR1, THICR2, THICT, THICT1, THICT2,
     &         THWELD, TIN, TIMU, TIMUA, TIMUB, TISIG, TISIGA, TISIGB,
     &         TLAM(MAXLD), TOUT, TOMU, TOMUA, TOMUB, TOSIG, TOSIGA,
     &         TOSIGB, TRSBND(0:MAXREG), TRUNC, TSLAM, TSTAT,
     &         V(2, MAXLD), VLAM(2, MAXLD), VSLAM(2), VSTAT(2), WOFF,
     &         WOFFA, WOFFB, WOFFC, WOFFD, WOFFE, WOFFHI, WOFFLO,
     &         WOFFR, WOFFR1, WOFFR2, WOFFR3, WOFFR4, WOFFT, WOFFT1,
     &         WOFFT2, WOFFT3, WOFFT4, Z

      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 = 'HEXHCD', STATUS = 'OLD')
      OPEN (3, FILE = 'HEXHCO', 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
      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
    5 CONTINUE

      READ(1,*) RAND
      WRITE(8,*)'                                RANDOM NUMBER SEED =', RAND
      READ(1,*) IOUT
      WRITE(8,*)'IOUT (MATCHR = 10, HEXHCF = 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'
      WRITE(8,*)'                               (0 - NO, 1 - YES) =', MPROC

      IF ((VARY .LT. 0) .OR. (VARY .GT. 3)) THEN
         WRITE(8,*) 'ERROR:  INVALID TYPE OF S/N VARIATION DESIRED'
         CALL TRMNAT
      ENDIF
      IF ((NMED .NE. 0) .AND. (NMED .NE. 1)) THEN
         WRITE(8,*) 'ERROR:  INVALID RESPONSE TO NORMAL MEDIAN ',
     &              'CURVE QUESTION'
         CALL TRMNAT
      ENDIF

      IF ((MPROC .LT. 0) .OR. (MPROC .GT. 1)) THEN
```

```
         WRITE(8,*) 'ERROR:  INVALID TYPE OF MATERIALS PROCESS ',
     &              'VARIATION DESIRED'
         CALL TRMNAT
      ENDIF

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


C ** READ DATA FROM HEXHCD

      READ(1,*) WOFFA, WOFFB, WOFFR1, WOFFR2, WOFFT1, WOFFT2,
     &          WOFFC, WOFFD, WOFFR3, WOFFR4, WOFFT3, WOFFT4,
     &          WOFFE,
     &          KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2,
     &          KWIDA, KWIDB, KWIDR1, KWIDR2, KWIDT1, KWIDT2,
     &          DIA,   DIB,   DIR1,   DIR2,   DIT1,   DIT2,
     &          THICA, THICB, THICR1, THICR2, THICT1, THICT2,
     &          LAMNA, LAMNB, LAMNC,  LAMND,
     &          LAMSA, LAMSB, LAMSC,  LAMSD,
     &          TIMUA, TIMUB, TISIGA, TISIGB,
     &          TOMUA, TOMUB, TOSIGA, TOSIGB,
     &          PCMUA, PCMUB, PCSIGA, PCSIGB,
     &          AERDA, AERDB, AERSA,  AERSB,
     &          DSTRA, DSTRB, ASTRA,  ASTRB,
     &          LAMWA, LAMWB, NEUBA,  NEUBB, GAMA,   GAMB

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

      READ(1,*) 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, WOFFR3, WOFFR4, WOFFT3, WOFFT4,
     &             WOFFE
      WRITE(3,902) KWODA, KWODB, KWODR1, KWODR2, KWODT1, KWODT2
      WRITE(3,903) KWIDA, KWIDB, KWIDR1, KWIDR2, KWIDT1, KWIDT2
      WRITE(3,904) DIA,   DIB,   DIR1,   DIR2,   DIT1,   DIT2
      WRITE(3,905) THICA, THICB, THICR1, THICR2, THICT1, THICT2
      WRITE(3,906) LAMNA, LAMNB, LAMNC,  LAMND
      WRITE(3,907) LAMSA, LAMSB, LAMSC,  LAMSD
      WRITE(3,908) TIMUA, TIMUB, TISIGA, TISIGB,
     &             TOMUA, TOMUB, TOSIGA, TOSIGB,
     &             PCMUA, PCMUB, PCSIGA, PCSIGB
      WRITE(3,909) AERDA, AERDB, AERSA,  AERSB,
     &             DSTRA, DSTRB, ASTRA,  ASTRB,
     &             LAMWA, LAMWB, NEUBA,  NEUBB, EXP(GAMA), EXP(GAMB)
      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(1,I), M(2,I), V(1,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)
   26       CONTINUE
            CLOSE (FILNUM(I))
         ELSE
            WRITE(8,*) 'ERROR:  CANNOT OPEN FILE, ', LDNAME(I),
     &                 ' DOES NOT EXIST'
            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), I = 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)
         WRITE(3,931) 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, FTU, FTY, 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
      DTISIG = TISIGB - TISIGA
      DTOMU = TOMUB - TOMUA
      DTOSIG = TOSIGB - TOSIGA
      DPCMU = PCMUB - PCMUA
      DPCSIG = PCSIGB - PCSIGA
```

```
      IF (IOUT .EQ. 15) THEN
         WRITE(8,*) 'DTIMU = ', DTIMU, ' DTISIG = ', DTISIG
         WRITE(8,*) 'DTOMU = ', DTOMU, ' DTOSIG = ', DTOSIG
         WRITE(8,*) 'DPCMU = ', DPCMU, ' DPCSIG = ', DPCSIG
      ENDIF

C ** 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, WOFFR,
     &                  WOFFT)
            WOFFLO = WOFFA
            WOFFHI = WOFFB
         ELSE
            CALL PRYRV (RAND, WOFFR3, WOFFR4, WOFFT3, WOFFT4, WOFFR,
     &                  WOFFT)
            WOFFLO = WOFFC
            WOFFHI = WOFFD
         ENDIF
         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'TEST =', TEST, ' WOFFE =', WOFFE
            WRITE(8,*) 'WOFFLO =', WOFFLO, ' WOFFHI =', WOFFHI
         ENDIF

         CALL PRYRV (RAND, KWIDR1, KWIDR2, KWIDT1, KWIDT2, KWIDR, KWIDT)
         CALL PRYRV (RAND, KWODR1, KWODR2, KWODT1, KWODT2, KWODR, KWODT)
         CALL PRYRV (RAND, DIR1,   DIR2,   DIT1,   DIT2,   DIR,   DIT)
         CALL PRYRV (RAND,THICR1, THICR2, THICT1, THICT2, THICR, THICT)

         CALL PRYRV (RAND, LAMNA, LAMNB,  LAMSA,  LAMSB,  LAMNK, LAMSK)
         LAMNMU = LAMND / (1.0 + LAMNK * LAMNC)
         LAMNSG = LAMNC / (1.0 + LAMNK * LAMNC)
         LAMSMU = LAMSD / (1.0 + LAMSK * LAMSC)
         LAMSSG = LAMSC / (1.0 + LAMSK * LAMSC)

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

         CALL RANDOM (NORM, RAND)
         TIMU = TIMUA + NORM * DTIMU
         TISIG = TISIGA + NORM * DTISIG
         TOMU = TOMUA + NORM * DTOMU
         TOSIG = TOSIGA + NORM * DTOSIG
         PCMU = PCMUA + NORM * DPCMU
         PCSIG = PCSIGA + NORM * DPCSIG

         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'NORM = ', NORM
            WRITE(8,*) '   TIMU = ', TIMU, ' TISIG = ', TISIG
            WRITE(8,*) '   TOMU = ', TOMU, ' TOSIG = ', TOSIG
            WRITE(8,*) '   PCMU = ', PCMU, ' PCSIG = ', PCSIG
         ENDIF

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
```

```
            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 FATIGUE LIVES

         DO 200 I = 1, NLIFE

C ** INITILIZE S/N CURVE PARAMETERS

            DO 225 L = 0, MAXREG
               LNA(L) = 0.0
               LPHIM(L) = 0.0
               TRSBND(L) = 0.0
  225       CONTINUE

C ** SELECT DRIVERS FOR CALCULATING LIFE

            CALL BETAGN (RAND, WOFFR, WOFFT, WOFFLO, WOFFHI, WOFF)
            CALL BETAGN (RAND, KWIDR, KWIDT, KWIDA,  KWIDB,  KWID)
            CALL BETAGN (RAND, KWODR, KWODT, KWODA,  KWODB,  KWOD)
            CALL BETAGN (RAND, DIR,   DIT,   DIA,    DIB,    DI)
            CALL BETAGN (RAND, THICR, THICT, THICA,  THICB,  THIC)
            CALL NORMGN (RAND, LAMNMU, LAMNSG, LAMN)
            CALL NORMGN (RAND, LAMSMU, LAMSSG, LAMS)
            CALL NORMGN (RAND, TIMU,   TISIG,  TIN)
            CALL NORMGN (RAND, TOMU,   TOSIG,  TOUT)
            DLTAT = TIN - TOUT
            CALL NORMGN (RAND, PCMU,   PCSIG,  PC)
            CALL PRYRV (RAND, AERDA, AERDB, AERSA, AERSB, AERD, AERS)
            CALL PRYRV (RAND, DSTRA, DSTRB, ASTRA, ASTRB, DSTR, ASTR)
            CALL PRYRV (RAND, LAMWA, LAMWB, NEUBA, NEUBB, LAMW, NEUB)
            CALL PRYRV (RAND, GAMA,  GAMB,  GAMA,  GAMB,  GAM,  DUM)
            GAM = EXP(GAM)
            CALL WEIBGN (BZERO, RAND, 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,*) 'LAMW =',LAMW,' NEUB =',NEUB,' GAM =', GAM
               WRITE(8,*) 'PHI =',PHI,' DI =', DI,' PC =',PC
               WRITE(8,*) 'TIN =',TIN,' TOUT =',TOUT,' DLTAT =',DLTAT
            ENDIF

C ** SCALE AERO STATIC LOADS

            PSLAM = AERS * ASTR * PSTAT
            TSLAM = AERS * ASTR * TSTAT
            MSLAM(1) = AERS * ASTR * MSTAT(1)
            MSLAM(2) = AERS * ASTR * MSTAT(2)
            VSLAM(1) = AERS * ASTR * VSTAT(1)
            VSLAM(2) = AERS * ASTR * VSTAT(2)

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) = LAMS * DSTR * P(II)
                  TLAM(II) = LAMS * DSTR * T(II)
```

```
              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,
     &                 ' M2 = ', MSLAM(1), ' M3 = ', MSLAM(2),
     &                 ' V2 = ', VSLAM(1), ' V3 = ', VSLAM(2)
            WRITE(8,*) 'TIME-VARYING LOADS'
            DO 240  II = 1, NLOAD
              WRITE(8,*) II, ' 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

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

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

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, LAMW, FK, RT, LNA, LNZ,
     &             LOCAT, LPHIM, MLAM, MM, MSLAM, NEUB, NLOAD, NRAN,
     &             NU, NUMREG, NUMSEG, PLAM, PC, PCO, PERIOD, PSLAM,
     &             TRSBND, SE, STRHIS, SZERO, TLAM, THIC, TRUNC,
     &             TSLAM, VLAM, VSLAM, WOFF, ZROREG)

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

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

200     CONTINUE

150   CONTINUE

      IF (NLIFET .GE. 100) THEN

C ** PRINT SORTED LIVES

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

```
C ** PRINT EMPIRICAL BLIVES
         FIFTY = 0.50E0
         WRITE(3,935)
         DO 350 J = 1, NBLIFE
            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, NUMREG, MCOUNT)
         MID = MCOUNT / 2
         DO 400 L = 1, NUMREG
            MEDM(L) = ALLM(MID, L)
  400    CONTINUE
         CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGK1, 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.',
     &       ////,33X,'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, 'Be(',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,')')
  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,'INNER DIAMETER',4X,'Be(',F6.4,',',F7.4,')',2X,
     &       'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')
  905 FORMAT(/,2X,'WALL THICKNESS',4X,'Be(',F6.4,',',F7.4,')',2X,
     &       'U(',F7.5,',',F8.5,')',4X,'U(',F4.1,',',F5.1,')')
  906 FORMAT(//,2X,'LAMBDA RANDOM',5X,'k: U(',F7.5,',',F8.5,')',
     &       /,20X,'COEFFICIENT OF VARIATION: ',F5.3,
     &       /,20X,'STRAIN GAGE FACTOR: ',F9.7)
  907 FORMAT(/,2X,'LAMBDA SINE',7X,'k: U(',F7.5,',',F8.5,')',
     &       /,20X,'COEFFICIENT OF VARIATION: ',F5.3,
     &       /,20X,'STRAIN GAGE FACTOR: ',F9.7)
  908 FORMAT(//,40X,'MU',15X,'SIGMA',
     &       //,2X,'INNER TEMPERATURE',4X,'NORMAL:  U(',
     &       F6.1,',',F7.1,')   U(',F5.1,',',F6.1,')',
     &       //,2X,'OUTER TEMPERATURE',4X,'NORMAL:  U(',
     &       F6.1,',',F7.1,')   U(',F5.1,',',F6.1,')',
     &       //,2X,'INNER PRESSURE',7X,'NORMAL:  U(',
     &       F6.1,',',F7.1,')   U(',F5.1,',',F6.1,')')
```

```
909 FORMAT(//,2X,'DYNAMIC AERO LOAD FACTOR',
     &         5X,'U(',F8.5,',',F9.5,')',
     &         //,2X,'STATIC AERO LOAD FACTOR',6X,'U(',F8.5,',',F9.5,')',
     &         //,2X,'DYNAMIC STRESS ANALYSIS',6X,'U(',F8.5,',',F9.5,')',
     &         //,2X,'AERO STRESS ANALYSIS',9X,'U(',F8.5,',',F9.5,')',
     &         //,2X,'LAMBDA KOFF',18X,'U(',F8.5,',',F9.5,')',
     &         //,2X,'NEUBERS RULE'17X,'U(',F8.5,',',F9.5,')',
     &    //,2X,'DAMAGE MODEL ACCURACY',8X,'U(ln',F8.5,', ln',F8.5,')')

920 FORMAT(////,28X,'LOADS INPUT'
     &        ///,5X,'P LOADS',5X,'T LOADS',5X,'M2 LOADS',
     &        4X,'M3 LOADS',4X,'V2 LOADS',4X,'V3 LOADS',
     &        /,6X,'(LBS)',5X,'(IN.-LBS)',4X,'(IN.-LBS)',
     &        3X,'(IN.-LBS)',5X,'(LBS)',6X,'(LBS)',
     &        //,2X,'STATIC AERO',
     &        /,2X,F9.6,3X,F9.6,3X,F9.6,3X,F9.6,3X,F9.6,3X,F9.6)

921 FORMAT(2X,A6,/,2X,F9.6,3X,F9.6,3X,F9.6,3X,F9.6,3X,F9.6,3X,F9.6)

925 FORMAT(////,20X,'GEOMETRIC AND OTHER INPUT',
     &        ///,2X,'K GEOM (OD)',41X,F4.2,
     &        //,2X,'K GEOM (ID)',41X,F4.2,
     &        //,2X,'K HOOP (OD)',41X,F4.2,
     &        //,2X,'K HOOP (ID)',41X,F4.2,
     &        //,2X,'EXTERNAL PRESSURE, PSI',26X, F6.0,
     &        //,2X,'ANALYSIS LOCATION',35X,I1
     &        //,2X,'ANGLE THETA (DEGREES)',28X,F6.1,
     &        //,2X,'STRESS-TIME HISTORY PERIOD, SEC',20X,F5.2,
     &        //,2X,'STRESS-TIME HISTORY NOISE FILTER, PSI',11X,F7.1,
     &        //,2X,'NUMBER OF TIME-VARYING LOADS',23X,I2,
     &        //,2X,'NUMBER OF POINTS IN HISTORIES',19X,I5)

926 FORMAT (/,2X,'ANGLE THETA (RADIANS)',29X,F6.2)

927 FORMAT (/,2X,'ELASTIC MODULUS, PSI',27X,E9.3,
     &         //,2X,'COEFF OF THERMAL EXPANSION',21X,E14.8,
     &         //,2X,'POISSONS RATIO',33X,F5.3)

930 FORMAT (///,25X,'STRESS-STRAIN CURVE INPUT',
     &         //,2X,'MAXIMUM NUMBER OF SEGMENTS ',25X,I1,
     &         //,2X,'STRESS-STRAIN PRODUCT',5X,'STRAIN VALUES',/)

931 FORMAT(13X,F8.2,10X,F7.5/)

935 FORMAT(////,2X,'B LIVES:      EMPIRICAL',/)

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

    STOP
    END


C***********************************************************************
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 0.01..........................B1 LIFE
C 0.10    0.10    0.50    0.50    5.0  5.....WELD OFFSET  (A,B) (R1,R2) (T1,T2)
C 0.00    0.00    0.00    0.00    0.0  0.0..................(C,D) (R3,R4) (T3,T4)
C 1.00..........................WELD OFFSET TEST FOR HYPER-DISTRIBUTION 1
C 0.88    1.32    0.30    0.70    0.5  10....K WELD (OD)   (A,B) (R1,R2) (T1,T2)
C 1.26    1.74    0.30    0.70    0.5  10....K WELD (ID)   (A,B) (R1,R2) (T1,T2)
C 0.1885   0.1915  0.50   0.75    0.5   20...INNER DIAM (A,B) (R1,R2) (T1,T2)
C 0.0113   0.0137  0.50   0.75    0.5   20...WALL THICK (A,B) (R1,R2) (T1,T2)
```

```
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 0.50          1.50..................LAMBDA DYNAMIC AERO U(A,B)
C 0.80          1.20..................LAMBDA STATIC 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
C  16.................................NUMBER OF TIME-VARYING LOADS
C
C  PSTAT    TSTAT    MSTAT(1)  MSTAT(2)  VSTAT(1)  VSTAT(2)
C
C  0.00      0.00    -0.07214   0.00      0.00      0.00..........STATIC AERO
C
C FILE   TYPE P()       T()       M(1, )    M(2, )    V(1, )    V(2, )     LOADS
C
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 'NBM2'   1 0.00      0.00      0.89901   0.00      0.00      0.00......RANDOM M2
C 'NBM3'   1 0.00      0.00      0.00      0.064034  0.00      0.00......RANDOM M3
C 'NBV2'   1 0.00      0.00      0.00      0.00      0.031219  0.00......RANDOM V2
C 'NBV3'   1 0.00      0.00      0.00      0.00      0.00      0.385245..RANDOM V3
C 'SIN1'   2 0.001150 0.001927   0.051544 0.000312 0.000596 0.022412..SINE 1
C 'SIN2'   2 0.002284 0.001917   0.050250 0.000457 0.000295 0.021862..SINE 2
C 'SIN3'   2 0.008944 0.003591   0.043037 0.002533 0.001717 0.026548..SINE 3
C 'SIN4'   2 0.023373 0.004778   0.427142 0.006577 0.004080 0.179889..SINE 4
C 'SIN5'   2 0.012763 0.000495   0.102889 0.007273 0.005673 0.051970..SINE 5
C 'SIN6'   2 0.00     0.00       0.00     0.00      0.00      0.00......SINE 6
C 'AERO1'  3 0.00     0.00       0.00     0.07179   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
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 (PCO, PSI)
C 2..............................SURFACE LOCATION (1=OUTER, 2=INNER)
C 270............................ANGLE (DEGREES)
C 1.0............................STRESS-TIME HISTORY PERIOD (SEC)
C 500............................STRESS-TIME HISTORY NOISE FILTER (PSI)
C 1000...........................NUMBER OF POINTS IN STRESS-TIME HISTORY
C 29000000.    8.8E-06    0.30....YOUNG'S MODULUS, COEFF OF THERMAL EXPANSION,
C                                   POISSON'S RATIO
C  0.615      2.00...............FK(1), RT(1)
C  0.693      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  0.873     12.50...............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    200.00...............FK(10), RT(10)
C 6..............................NUMBER OF SEGMENTS IN STRESS-STRAIN CURVE
C 21.95      0.001...............STRESS-STRAIN PRODUCT: SE(1), STRAIN: E(1)
C 55.77      0.002...............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    -1.0     1 ..............# PTS IN DIV, STRESS RATIO, REGION
C 40000.      1000...............S(1)     N(1)
C 40000.      2000...............S(2)     N(2)         RAW
C 40000.      3000...............S(3)     N(3)        STRESS
C 40000.      4000...............S(4)     N(4)         LIFE
```

```
C 40000.     5000...............S(5)    N(5)      (S/N)
C 40000.     6000...............S(6)    N(6)      DATA
C 30000.     23000..............S(7)    N(7)     POINTS
C 30000.     66000..............S(8)    N(8)      FOR
C 25000.     72000..............S(9)    N(9)      THE
C 25000.     190000.............S(10)   N(10)    SPECIFIC
C 20000.     789000.............S(11)   N(11)    MATERIAL
C 20000.     1070000............S(12)   N(12)
C 20000.     1450000............S(13)   N(13)
C 0.00..........................NO VALUE OF So SUPPLIED
C 1    0........................NUMBER OF REGIONS: W/DATA, W/O DATA
C 1.0E+36.......................LIFE BOUNDRIES REGION 1
C 0.00..........................CONSTRAINT ON COEFF. OF VARIATION
C 0     0.000     0.000..........2 PTS IN RANGE, LOWER BOUND, UPPER BOUND
C 0.0   0.0       0.0............NORMAL DISTRIBUTION PRIORS: DELTA, Mo, SIGMA2
C
C
C***********************************************************************
C                         LIST OF VARIABLES
C***********************************************************************
C
C AERD          SELECTED AERO DYNAMIC LOAD SCALE FACTOR
C AERDA         AERO DYNAMIC LOAD SCALE FACTOR LOWER BOUND
C AERDB         AERO DYNAMIC LOAD SCALE FACTOR UPPER BOUND
C AERS          SELECTED AERO STATIC LOAD SCALE FACTOR
C AERSA         AERO STATIC LOAD SCALE FACTOR LOWER BOUND
C AERSB         AERO STATIC LOAD SCALE FACTOR UPPER BOUND
C ALLM()        2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C ANGLE         ANGLE THETA MEASURED COUNTER-CLOCKWISE FROM THE X2-DIRECTION
C                 GIVEN IN DEGREES, TRANSFORMED TO RADIANS FOR CALCULATIONS
C ASTR          SELECTED AERO STRESS ANALYSIS ACCURACY FACTOR
C ASTRA         AERO STRESS ANALYSIS ACCURACY FACTOR LOWER BOUND
C ASTRB         AERO STRESS ANALYSIS ACCURACY FACTOR UPPER BOUND
C BIGK()        1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                 EACH REGION
C BIGK1         EQUAL TO BIGK(1) -- DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C                 EXPCTD
C BLFPER()      1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS()      1-D ARRAY CONTAINING POSITIONS IN LIFE() OF EMPIRICAL BLIVES
C BZERO         VALUE OF BETAo RANDOMLY SELECTED FROM BETA() INTERVAL
C COEXP         COEFFICIENT OF THERMAL EXPANSION
C DI            SELECTED WELD INTERIOR DIAMETER
C DIA           WELD INTERIOR DIAMETER LOWER BOUND
C DIB           WELD INTERIOR DIAMETER UPPER BOUND
C DIR           SELECTED RHO FOR WELD INTERIOR DIAMETER
C DIR1          WELD INTERIOR DIAMETER - RHO LOWER BOUND
C DIR2          WELD INTERIOR DIAMETER - RHO UPPER BOUND
C DIT           SELECTED THETA FOR WELD INTERIOR DIAMETER
C DIT1          WELD INTERIOR DIAMETER - THETA LOWER BOUND
C DIT2          WELD INTERIOR DIAMETER - THETA UPPER BOUND
C DLTAT         SELECTED TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER
C                 SURFACES -- DeLTA T
C DPCMU         EQUAL TO PCMUB - PCMUA
C DPCSIG        EQUAL TO PCSIGB - PCSIGA
C DSTR          SELECTED DYNAMIC STRESS ANALYSIS ACCURACY FACTOR
C DSTRA         DYNAMIC STRESS ANALYSIS ACCURACY FACTOR LOWER BOUND
C DSTRB         DYNAMIC STRESS ANALYSIS ACCURACY FACTOR UPPER BOUND
C DTIMU         EQUAL TO TIMUB - TIMUA
C DTISIG        EQUAL TO TISIGB - TISIGA
C DTOMU         EQUAL TO TOMUB - TOMUA
C DTOSIG        EQUAL TO TOSIGB - TOSIGA
C DUM           DUMMY VARIABLE
C E()           1-D ARRAY WHICH CONTAINS THE STRAIN VALUES
C EM()          ELASTIC MODULUS
C FIFTY         EQUAL TO .5 -- USED TO ACCESS 50% POINT IN LIFE()
C FILNUM()      1-D ARRAY CONTAINING UNIT NUMBER FOR STRESS-TIME HISTORIES
C                 FILES
C FK()          1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C                 CONCENTRATION DUE TO WELD ECCENTRICITY
C FTEST         File TEST -- USED TO TEST EXISTENCE OF FILE
C FTU           MATERIAL ULTIMATE STRENGTH
C FTY           MATERIAL YIELD STRENGTH
C GAM           SELECTED DAMAGE ACCUMULATION MODEL ACCURACY FACTOR, LAMBDAdam
C GAMA          GAM LOWER BOUND
C GAMB          GAM UPPER BOUND
```

```
C I            CONTROLS DO LOOP FOR EACH LIFE CALCULATION
C II           CONTROLS DO LOOP FOR LOADS
C IOUT         CONTROLS DUMP TO SCREEN/PRINTER
C J            CONTROLS DO LOOP FOR EACH BLIFE
C K            CONTROLS DO LOOP FOR EACH HYPER PARAMETER SET
C KGID         K GEOM (ID)
C KGOD         SELECTED K GEOM (OD)
C KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C KT()         FATIGUE STRESS CONCENTRATION FACTORS -- KT(1,1) = Kt AXIAL (OD)
C                (= KGOD * KWOD); KT(1,2) = Kt AXIAL (ID) (= KGID * KWID);
C                KT(2,1) = Kt HOOP (OD); KT(2,2) = Kt HOOP (ID)
C KWID         SELECTED K WELD (ID)
C KWIDA        K WELD (ID) LOWER BOUND
C KWIDB        K WELD (ID) UPPER BOUND
C KWIDR        SELECTED RHO FOR K WELD (ID)
C KWIDR1       K WELD (ID) - RHO LOWER BOUND
C KWIDR2       K WELD (ID) - RHO UPPER BOUND
C KWIDT        SELECTED THETA FOR K WELD (ID)
C KWIDT1       K WELD (ID) - THETA LOWER BOUND
C KWIDT2       K WELD (ID) - THETA UPPER BOUND
C KWOD         SELECTED K WELD (OD)
C KWODA        K WELD (OD) LOWER BOUND
C KWODB        K WELD (OD) UPPER BOUND
C KWODR        SELECTED RHO FOR K WELD (OD)
C KWODR1       K WELD (OD) - RHO LOWER BOUND
C KWODR2       K WELD (OD) - RHO UPPER BOUND
C KWODT        SELECTED THETA FOR K WELD (OD)
C KWODT1       K WELD (OD) - THETA LOWER BOUND
C KWODT2       K WELD (OD) - THETA UPPER BOUND
C L            CONTROLS DO LOOP FOR EACH REGION
C LAMN         SELECTED LAMBDA FOR ONE SIGMA NARROW-BAND RANDOM LOADS
C LAMNA        LAMBDA FOR NARROW-BAND RANDOM LOADS -- LOWER BOUND OF k
C LAMNB        LAMBDA FOR NARROW-BAND RANDOM LOADS -- UPPER BOUND OF k
C LAMNC        LAMBDA FOR NARROW-BAND RANDOM LOADS COEFFICIENT OF VARIATION
C LAMND        NARROW-BAND RANDOM LOADS STRAIN GAGE ACCURACY FACTOR
C LAMNK        LAMBDA FOR NARROW-BAND RANDOM LOADS k -- INDICATES VARIATION
C                DUE TO SAMPLE SIZE
C LAMNMU       MEAN OF LAMBDA FOR NARROW-BAND RANDOM LOADS (MU, NORMAL
C                DISTRIBUTION)
C LAMNSG       STANDARD DEVIATION OF LAMBDA FOR NARROW-BAND RANDOM LOADS
C                (SIGMA, NORMAL DISTRIBUTION)
C LAMS         SELECTED LAMBDA FOR SUPERIMPOSED SINE LOADS
C LAMSA        LAMBDA FOR SUPERIMPOSED SINE LOADS -- LOWER BOUND OF k
C LAMSB        LAMBDA FOR SUPERIMPOSED SINE LOADS -- UPPER BOUND OF k
C LAMSC        LAMBDA FOR SUPERIMPOSED SINE LOADS COEFFICIENT OF VARIATION
C LAMSD        SUPERIMPOSED SINE LOADS STRAIN GAGE ACCURACY FACTOR
C LAMSK        LAMBDA FOR SUPERIMPOSED SINE LOADS k -- INDICATES VARIATION
C                DUE TO SAMPLE SIZE
C LAMSMU       MEAN OF LAMBDA FOR SUPERIMPOSED SINE LOADS (MU, NORMAL
C                DISTRIBUTION)
C LAMSSG       STANDARD DEVIATION OF LAMBDA FOR SUPERIMPOSED SINE LOADS
C                (SIGMA, NORMAL DISTRIBUTION)
C LAMW         SELECTED ACCURACY FACTOR FOR WELD ECCENTRICITY STRESS
C                CONCENTRATION FACTOR, Koff
C LAMWA        LAMW LOWER BOUND
C LAMWB        LAMW UPPER BOUND
C LDNAME()     1-D ARRAY CONTAINING LoaD NAMEs FOR THE TIME-VARYING LOADS
C LIFE()       1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
C                -- SORTED VALUES OF THE LEFT-HAND TAIL
C LNA()        1-D ARRAY CONTAINING Ln(A) = Ln(BIGK)*MM FOR EACH REGION
C LNZ          NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C LOCAT        LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE OF THE
C                DUCT, AND 2 IS THE INTERIOR SURFACE OF THE DUCT
C LPHIM()      1-D ARRAY CONTAINING Ln(PHI)*MM FOR EACH REGION
C M()          2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C MAXBLF       MAXIMUM NUMBER OF BLIVES TO BE CALCULATED
C MAXDAT       MAXIMUM NUMBER OF POINTS PER DATA SET PER REGION ALLOWED
C MAXLD        MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C MAXLIF       MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA,
C                ALPHA CALCULATION
C MAXM         MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C MAXMM        MAXIMUM NUMBER OF M's TO BE SORTED FOR MEDIAN CALCULATION
C MAXREG       MAXIMUM NUMBER OF REGION ALLOWED (S/N CURVE)
C MAXSEG       MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN CURVE)
```

```
C MCOUNT       NUMBER OF M's TO BE USED TO CALCULATE MEDIAN S/N CURVE
C MEDM()       1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C MID          POINTER TO THE MEDIAN M VALUES -- EQUAL TO HALF OF MCOUNT
C MLAM()       2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS SCALED
C                BY LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY TYPE())
C                -- MLAM(1,*) ARE THE M2 LOADS; MLAM(2,*) ARE THE M3 LOADS
C MM()         1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C MPROC        Materials PROCess variation -- CONTROLS MATERIALS PROCESS
C                VARIATION -- 0 - NO VARIATION;  1 - INCLUDE VARIATION
C MSLAM()      1-D ARRAY CONTAINING THE STATIC LOADS SCALED BY AERS --
C                MSLAM(1) IS THE M2 LOAD; MSLAM(2) IS THE M3 LOAD
C MSTAT()      1-D ARRAY CONTAINING THE STATIC LOADS -- MSTAT(1) IS THE M2
C                LOAD; MSTAT(2) IS THE M3 LOAD
C MU()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C NBLIFE       NUMBER OF BLIVES TO BE CALCULATED
C NBND()       1-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE
C                REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE)
C                MATERIAL S/N DATA SET
C NEUB         SELECTED NEUBER'S RULE MODEL ACCURACY FACTOR
C NEUBA        NEUBER'S RULE MODEL ACCURACY FACTOR LOWER BOUND
C NEUBB        NEUBER'S RULE MODEL ACCURACY FACTOR UPPER BOUND
C NEWLIF       LIFE VALUE RETURNED FROM CALL TO THWELD
C NF()         2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL
C                S/N DATA SET BROKEN INTO LIFE REGIONS
C NHYPER       SIZE OF OUTER LOOP
C NLIFE        SIZE OF INNER LOOP
C NLIFET       TOTAL NUMBER OF LIVES CALCULATED BY PFM
C NLOAD        NUMBER OF TIME-VARYING LOADS
C NMED         CONTROLS MEDIAN CALCULATION FOR THE NORMAL DISTRIBUTION CASE
C                -- 0 - NO MEDIAN CALCULATION;  1 - MEDIAN CALCULATION DESIRED
C NORM         RANDOM VARIABLE (SOMETIMES UNIFORM, SOMETIMES NORMAL) USED
C                TO OBTAIN SELECTED TEMPERATURES AND PRESSURE
C NPTS()       1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION
C                FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C NRAN         NUMBER OF POINTS IN STRESS-TIME HISTORY (Number of RANdom
C                points)
C NU           POISSON'S RATIO
C NUMREG       NUMBER OF REGIONS OF INTEREST
C NUMSEG       NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
C P()          1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C PC           SELECTED INTERNAL PRESSURE, PSI
C PCMU         SELECTED MEAN OF INTERNAL PRESSURE, PSI (MU, NORMAL
C                DISTRIBUTION)
C PCMUA        MEAN OF INTERNAL PRESSURE LOWER BOUND
C PCMUB        MEAN OF INTERNAL PRESSURE UPPER BOUND
C PCO          EXTERNAL PRESSURE, PSI
C PCSIG        SELECTED STANDARD DEVIATION OF INTERNAL PRESSURE, PSI (SIGMA,
C                NORMAL DISTRIBUTION)
C PCSIGA       STANDARD DEVIATION OF INTERNAL PRESSURE LOWER BOUND
C PCSIGB       STANDARD DEVIATION OF INTERNAL PRESSURE UPPER BOUND
C PERIOD       LENGTH OF TIME IN SECONDS OF RANDOM STRESS-TIME HISTORY
C PHI          WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C PI           CONSTANT FOR THE VALUE 3.1415926536
C PLAM()       1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS SCALED
C                BY LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY TYPE())
C PSIG         EQUAL TO SQRT(PVAR) -- MATERIALS PROCESS STANDARD DEVIATION
C PSLAM        STATIC AXIAL LOAD SCALED BY AERS
C PSTAT        STATIC AXIAL LOAD
C PVAR         MATERIALS PROCESS VARIATION
C RAND         RANDOM NUMBER SEED
C RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C RT()         1-D ARRAY CONTAINING VALUES OF Rt USED TO FIND STRESS
C                CONCENTRATION DUE TO WELD ECCENTRICITY
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()
C SE()         1-D ARRAY OF PRODUCT OF STRESS AND STRAIN FOR EACH SEGMENT OF
C                THE STRESS-STRAIN VS STRAIN CURVE
C SIG()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C STR()        2-D ARRAY CONTAINING STRESS POINTS (STRESS RATIO = -1.0)
```

```
C                 FOR THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO
C                 LIFE REGIONS
C STRHIS()      2-D ARRAY CONTAINING THE AMPLITUDES FOR THE TIME-VARYING
C                 STRESS-TIME HISTORIES
C SZERO         STRESS TENSILE TEST POINT, So
C T()           1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C TEST          UNIFORM(0,1) RANDOM VARIATE USED TO DETERMINE
C                 HYPER-DISTRIBUTION TO SELECT FROM
C THIC          SELECTED WALL THICKNESS AT BEND (ID) AT WELD, IN
C THICA         WALL THICKNESS AT BEND (ID) LOWER BOUND
C THICB         WALL THICKNESS AT BEND (ID) UPPER BOUND
C THICR         SELECTED RHO FOR WALL THICKNESS AT BEND (ID)
C THICR1        WALL THICKNESS AT BEND (ID) - RHO LOWER BOUND
C THICR2        WALL THICKNESS AT BEND (ID) - RHO UPPER BOUND
C THICT         SELECTED THETA FOR WALL THICKNESS AT BEND (ID)
C THICT1        WALL THICKNESS AT BEND (ID) - THETA LOWER BOUND
C THICT2        WALL THICKNESS AT BEND (ID) - THETA UPPER BOUND
C THWELD        REAL FUNCTION WHICH CALCULATES THE DUCT LIFE (IN SECONDS)
C                 AT A PLAIN WELD UNDERGOING THERMAL LOADING
C TIN           SELECTED INNER WALL SURFACE TEMPERATURE (RANKINE)
C TIMU          SELECTED MEAN OF INNER WALL TEMPERATURE, (MU, NORMAL
C                 DISTRIBUTION)
C TIMUA         MEAN OF INNER WALL TEMPERATURE LOWER BOUND
C TIMUB         MEAN OF INNER WALL TEMPERATURE UPPER BOUND
C TISIG         SELECTED STANDARD DEVIATION OF INNER WALL TEMPERATURE, (SIGMA,
C                 NORMAL DISTRIBUTION)
C TISIGA        STANDARD DEVIATION OF INNER WALL TEMPERATURE LOWER BOUND
C TISIGB        STANDARD DEVIATION OF INNER WALL TEMPERATURE UPPER BOUND
C TLAM()        1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS SCALED
C                 BY LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY TYPE())
C TOUT          SELECTED OUTER WALL SURFACE TEMPERATURE (RANKINE)
C TOMU          SELECTED MEAN OF OUTER WALL TEMPERATURE (MU, NORMAL
C                 DISTRIBUTION)
C TOMUA         MEAN OF OUTER WALL TEMPERATURE LOWER BOUND
C TOMUB         MEAN OF OUTER WALL TEMPERATURE UPPER BOUND
C TOSIG         SELECTED STANDARD DEVIATION OF OUTER WALL TEMPERATURE, (SIGMA,
C                 NORMAL DISTRIBUTION)
C TOSIGA        STANDARD DEVIATION OF OUTER WALL TEMPERATURE LOWER BOUND
C TOSIGB        STANDARD DEVIATION OF OUTER WALL TEMPERATURE UPPER BOUND
C TRSBND()      1-D ARRAY CONTAINING VALUES OF SBND * PHI * KRATIO * Z FOR
C                 EACH REGION AND TRIAL
C TRUNC         VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
C TSLAM         STATIC AXIAL LOAD SCALED BY AERS
C TSTAT         STATIC TORQUE LOAD
C TYPE()        1-D ARRAY CONTAINING THE TYPE OF TIME-VARYING LOAD, USED FOR
C                 LOAD FACTORS -- TYPE(*) = 1 INDICATES NARROW-BAND RANDOM;
C                 TYPE(*) = 2 INDICATES SUPERIMPOSED SINUSOID; TYPE(*) = 3
C                 INDICATES DYNAMIC AERO
C V()           2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                 ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C VARY          CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO VARIATION;
C                 1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;  3 -
C                 TRUNCATED NORMAL VARIATION
C VLAM()        2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS SCALED
C                 BY LAMS, LAMN, OR AERD AS APPROPRIATE (INDICATED BY TYPE())
C                 -- VLAM(1,*) ARE THE V2 LOADS; VLAM(2,*) ARE THE V3 LOADS
C VSLAM()       1-D ARRAY CONTAINING THE STATIC SHEAR LOADS SCALED BY AERS
C                 -- VSLAM(1) IS THE V2 LOAD; VSLAM(2) IS THE V3 LOAD
C VSTAT()       1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- VSTAT(1) IS
C                 THE V2 LOAD; VSTAT(2) IS THE V3 LOAD
C WOFF          SELECTED WELD OFFSET (%)
C WOFFA         WELD OFFSET LOWER BOUND - HYPER-DISTRIBUTION 1
C WOFFB         WELD OFFSET UPPER BOUND - HYPER-DISTRIBUTION 1
C WOFFC         WELD OFFSET LOWER BOUND - HYPER-DISTRIBUTION 2
C WOFFD         WELD OFFSET UPPER BOUND - HYPER-DISTRIBUTION 2
C WOFFE         PERCENTAGE OCCURRING IN HYPER-DISTRIBUTION 1
C WOFFHI        SELECTED WELD OFFSET UPPER BOUND
C WOFFLO        SELECTED WELD OFFSET LOWER BOUND
C WOFFR         SELECTED RHO FOR WELD OFFSET
C WOFFR1        WELD OFFSET - RHO LOWER BOUND - HYPER-DISTRIBUTION 1
C WOFFR2        WELD OFFSET - RHO UPPER BOUND - HYPER-DISTRIBUTION 1
C WOFFR3        WELD OFFSET - RHO LOWER BOUND - HYPER-DISTRIBUTION 2
C WOFFR4        WELD OFFSET - RHO UPPER BOUND - HYPER-DISTRIBUTION 2
C WOFFT         SELECTED THETA FOR WELD OFFSET
C WOFFT1        WELD OFFSET - THETA LOWER BOUND - HYPER-DISTRIBUTION 1
```

```
C WOFFT2        WELD OFFSET - THETA UPPER BOUND - HYPER-DISTRIBUTION 1
C WOFFT3        WELD OFFSET - THETA LOWER BOUND - HYPER-DISTRIBUTION 2
C WOFFT4        WELD OFFSET - THETA UPPER BOUND - HYPER-DISTRIBUTION 2
C Z             LOG-NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C ZROREG        ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE -- 0 - ZERO REGION EXITS, 1 - NO ZERO
C                 REGION

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

C  SUBROUTINE INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20JUN90
C     VERSION:  2.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C   INPUTS:  NEWLIF, LIFE, NLIFET
C  OUTPUTS:  LIFE

C     IMPLICIT NONE

      INTEGER  MAXLIF

      PARAMETER (MAXLIF = 10000)

      COMMON   IOUT

      INTEGER  I, IOUT, NLIFET, NUM, PLACE

      REAL     LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)

C                    LIST OF VARIABLES
C
C  I        CONTROLS DO LOOP FOR INSERTION
C  IOUT     OUTPUT DUMP CONTROLLER
C  LIFE()   1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
C             PFM TO BE SORTED
C  MAXLIF   MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
C             CALCULATION
C  NEWLIF   LIFE VALUE TO BE INSERTED INTO LIFE()
C  NLIFET   TOTAL NUMBER OF LIVES CALCULATED BY PFM
C  NUM      NUMBER OF LIFE VALUES IN LIFE()
C  PLACE    POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C  TEMP()   1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
C             INSERTION OF NEWLIF

      NUM = NLIFET / 2

C     FIND POSITION IN LIFE() FOR NEWLIF

      IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

      DO 100 I = 1, NUM
         IF (NEWLIF .LT. LIFE(I)) THEN
            PLACE = I
            GOTO 110
         ENDIF
  100 CONTINUE
  110 CONTINUE

C     STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()
```

```
      DO 200 I = (PLACE + 1), NUM
         TEMP(I) = LIFE(I-1)
  200 CONTINUE

C     INSERT NEWLIF

      LIFE(PLACE) = NEWLIF

C     SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

      DO 300 I = (PLACE + 1), NUM
         LIFE(I) = TEMP(I)
  300 CONTINUE

C     IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN

  400 CONTINUE

      RETURN
      END


C***********************************************************************
C   SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C   INDEPENDENT RANDOM VARIATES
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  9MAR87
C   SUBPROGRAM:  RANDOM
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, RHO1, RHO2, THE1, THE2, X, Y

      INTEGER IOUT

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

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

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &         ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      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

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

C  TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

      X = W * (B - A) + A
      IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X

      RETURN
      END

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

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    pp. 407-409.

      REAL FUNCTION GAM (ALPHA, RAND)

C     SUBPROGRAM:  RANDOM

      COMMON  IOUT

      INTEGER IOUT

      REAL    A, ALPHA, ARG, U1, U2, V1, V2

      DOUBLE PRECISION RAND

      A = ALPHA - 1.
C     IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA

   10 CALL RANDOM (U1, RAND)
      CALL RANDOM (U2, RAND)
      V1 = - ALOG(U1)
      V2 = - ALOG(U2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =',
C    &                     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
C  is acknowledged.

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

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

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

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

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
     &        NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
     &        NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG

      REAL    BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
     &        FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
     &        KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
     &        MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET),
     &        RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &        SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT, MAXREG),
     &        SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(MAXREG),
     &        SXY(MAXREG), SY2(MAXREG), SZERO


C                      LIST OF VARIABLES
C
C  BIGKHT      EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
C  BZERO       VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N
C                DATA SET
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
C                AND SIG() CALCULATION
C  FTUZ        ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  IOUT        OUTPUT DUMP CONTROLLER
C  IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                FOR EACH REGION
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LAMN        LAMBDA-N -- RATIO OF Var(Ln N given S) / (m**2 C**2),
C                CONSTANT OVER REGIONS AND COMPONENTS
C  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 IN 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  MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                BOUND
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
```

```
C                   FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                   MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                   MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()        1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                   MC() FOR EACH REGION
C  MEDM()         1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C  MO()           1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                   MEAN FOR EACH REGION
C  MPNT()         1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                   MZERO() FOR EACH REGION
C  MPROC          Materials PROCess variation --CONTROLS MATERIALS PROCESS
C                   VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MU()           1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                   DISTRIBUTION MEAN FOR EACH REGION
C  MZERO()        2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                   EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                   IS THE UPPER BOUND
C  NBND()         1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                   REGIONS OF INTEREST
C  NF()           2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                   SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NNODAT         Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()           2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                   SET IN EACH REGION
C  NPPR()         1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                   ALL DATA SETS IN A REGION (Number of Points Per Region)
C  NPTS()         1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS          NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUMREG         NUMBER OF REGIONS OF INTEREST
C  PVAR           MATERIALS PROCESS VARIATION
C  RANGEM()       2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND
C  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) OR 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  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  STR()          2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                   S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SUHAT2()       1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                   REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT2()       1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
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  SXY()          1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                   REGION (X = Ln S, Y = Ln N)
C  SY2()          1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                   (Y = Ln N)
C  SZERO          STRESS TENSILE TEST POINT, So
C  VARY           CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                   VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                   VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  ZROREG         ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 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
C  INFORMATION

C  PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION
```

```
C   INITIALIZE PRIMARY ARRAYS

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

C   READ, CONVERT, ECHO INFORMATION

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

C   CALCULATE RESIDUAL VARIANCES

       CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, 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, MCHAT)

C   CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

       IF (MPROC .EQ. 1) THEN
          CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
       ENDIF

C   COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C   OBTAIN POSTERIOR RANGES ON M

          CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
     &                 RANGEM)

C   ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

          CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C   ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

          CALL CONCAV (NUMREG, RANGEM)

C   WRITE RESULTS TO FILE DUMP

          WRITE(7,900)

          DO 25 L = 1, NUMREG
             WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
     &                       JZERO(1, L), JZERO(2, L)
   25     CONTINUE

          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,
     &                  NBND, BIGKHT, BZERO)

C  CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED

           IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
              DO 200 L = 1, NUMREG
                 RANGEM(1,L) = MEDM(L)
                 RANGEM(2,L) = MEDM(L)
 200          CONTINUE
           ENDIF

        ELSE

C  NORMAL VARIATION IS DESIRED

C  CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION

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

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

        IF (MPROC .EQ. 1) THEN
           CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
        ENDIF

C  COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M

           CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

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

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

           CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

           WRITE(7,975)

           DO 350 L = 1, NUMREG
              WRITE(7,980) L, MCHAT(1,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(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)
         DO 380 L = 1, NUMREG
            WRITE(7,990) L, MU(L), SIG(L)
 380  CONTINUE

      ENDIF

C  PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS

      IF (MPROC .EQ. 1) THEN
         WRITE(7,995) PVAR
      ENDIF

C  FORMAT STATEMENTS

 900  FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &          'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &          'NASA Contract NAS7-918 is acknowledged.',////,
     &          2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
     &          ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
     &          'FOR EACH REGION',/)

 905  FORMAT(7X,'REGION:  ',I1,7X,'Io = (',F12.9,',',F12.9,')',
     &          /,24X,'Jo = (',F12.9,',',F12.9,')')

 910  FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
     &          //,7X,'REGION',8X,'E(C)',12X,'E(m)',/)

 915  FORMAT(9X,I1,8X,F11.9,5X,F9.6)

 920  FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH '
     &          'REGION')

 930  FORMAT(//,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

 940  FORMAT(6X,I1,8X,F8.4,8X,F8.4)

 950  FORMAT(///)

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

 965  FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')

 970  FORMAT(6X,I1,8X,F8.4,8X,F8.4)

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

 980  FORMAT(9X,I1,11X,F10.6)

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

 990  FORMAT(5X,I1,5X,F7.4,5X,E11.5)

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

```
     &       'INFORMATION',//,7X,E11.5)

      RETURN
      END

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


C  SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C  ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  5OCT87
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRMNAT

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

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


C  SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C  USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   ---
C  OUTPUTS:  NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
C            NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

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

      COMMON  IOUT

      INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
     &        NPTS(0:MAXSET), REFNP(MAXREG)

      REAL    DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
     &        RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG)

C                     LIST OF VARIABLES
C
C  DELTA()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C               SIG() CALCULATION
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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 IN 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  MO()         1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  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
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)

      DO 100 J = 0, MAXSET
         NPTS(J) = 0.0
  100 CONTINUE

      DO 200 L = 1, MAXREG
         DO 250 J = 0, MAXSET
            NP(J, L) = 0.0
  250    CONTINUE
  200 CONTINUE

      DO 300 J = 0, MAXSET
         DO 350 I = 1, MAXDAT
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            RATSTR(I,J) = 0.0
  350    CONTINUE
  300 CONTINUE

      DO 400 L = 1, MAXREG
         DO 425 K = 1, MAXDAT
            DO 450 J = 0, MAXSET
               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"
C   THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C   SPECFO AND RELATO.  RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C   SPECIFIED BY USER
C   PROGRAMMER:  L. NEWLIN
C         DATE:  21JUN88      FORMAT/COMMENTS:  12AUG91
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
     &                LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
     &                NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
     &                FTUZ, FTYZ, DELTA, MO, SIGMA2, 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,
C              CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2,
C              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),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RATIO, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG), SZERO

      CHARACTER*40  DESCRP(0:MAXSET)

C                          LIST OF VARIABLES
C
C   COUNT        INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO,
C                  CONVERSION, AND BREAK UP
C   CZERO        EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                  COEFFICIENT OF VARIATION, Co
C   DELTA()      1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                  SIG() CALCULATION
C   DESCRP()     1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C   FTU          ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
C   FTUZ         ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   FTY          YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   FTYZ         YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   I            CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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   KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C   L            CONTROLS DO LOOP FOR EACH REGION
C   LAMN         LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                  CONSTANT OVER ALL REGIONS AND COMPONENTS
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   M            CONTROLS DO LOOP FOR EACH DATA DIVISION
```

```
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN 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  MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MPROC       Materials PROCess variation -- CONTROLS MATERIALS PROCESS
C                VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NDIV        NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
C                REGION PAIRS DURING INPUT
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                IN EACH REGION
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM         NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  RATIO       STRESS RATIO (R = -1.0 IS DESIRED)
C  RATSTR()    2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C                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) OR 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  REG         REGION OF INTEREST IN A PARTICULAR DIVISION
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
C                S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZERO       STRESS TENSILE TEST POINT, So
C  VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  ZROREG      ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                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), FTY, FTU, NDIV, NPTS(0)

      IF (NPTS(0) .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF

      WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
      IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)

      WRITE(3,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

C     STORE VALUES OF SPECIFIC MATERIAL FTU AND FTY INTO FTUZ AND FTYZ
```

```
      FTUZ = FTU
      FTYZ = FTY
C     INPUT STRESS/LIFE INFORMATION -- INCLUDING STRESS RATIO AND REGION
C     INFORMATION FROM SPECFD AND ECHO TO SPECFO

      DO 100 M = 1, NDIV

         READ (1,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE FOR RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

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

         DO 110 I = (COUNT + 1), (COUNT + NUM)
            READ(1,*) RAWSTR(I,0), RAWNF(I,0)
  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, FTY)

         ENDIF

C        ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL

         DO 130 I = (COUNT + 1), (COUNT + NUM)

            WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
     &                   RATSTR(I,0), RAWNF(I,0)

            IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
     &                   RATIO, REG, RATSTR(I,0), RAWNF(I,0)

  130    CONTINUE

C        BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C        EXPCTD, AND PAREST

         K = NP(0,REG)

         DO 140 I = (COUNT + 1), (COUNT + NUM)

            K = K + 1
            LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
            LNNF(K,0,REG) = ALOG(RAWNF(I,0))
            STR(K,REG) = RATSTR(I,0)
            NF(K,REG) = RAWNF(I,0)

  140    CONTINUE

         IF (K .GT. MAXDAT) THEN
            WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &                 'SPECIFIC MATERIAL'
            CALL TRMNAT
         ENDIF
```

```
         NP(0,REG) = K
         REFNP(REG) = K
         COUNT = COUNT + NUM

  100 CONTINUE

      IF (NPTS(0) .NE. COUNT) THEN
         WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &              'INCORRECTLY SPECIFIED'
         WRITE(8,*) 'IN SPECIFIC DATA SET'
         CALL TRMNAT
      ENDIF

      READ(1,*) SZERO
      IF (NINT (SZERO) .GT. 0) THEN
         ZROREG = 0
      ELSE
         ZROREG = 1
      ENDIF
      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG

C     INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION

      READ(1,*) NUMREG, NNODAT

      IF ((NUMREG + NNODAT) .GT. MAXREG)  THEN
         WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON NUMBER OF REGIONS'
         CALL TRMNAT
      ENDIF

      DO 150 L = ZROREG, (NUMREG + NNODAT)
         READ(1,*) NBND(L)
  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
         ENDIF
  180 CONTINUE

      IF (VARY .EQ. 3) THEN
C        READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
         WRITE(3,945)
         IF (IOUT .EQ. 10) WRITE(8,945)
```

```
         DO 190 L = 1, (NUMREG + NNODAT)
            READ(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
            ENDIF
 190     CONTINUE
      ENDIF

      IF (MPROC .EQ. 1) THEN
         READ(1,*) KRATIO, LAMN
         WRITE(3,955) KRATIO, LAMN
         IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
      ENDIF


C  BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATD
C  AND THEN ECHO TO RELATO

      READ(5,*) NSETS

      IF (NSETS .GT. MAXSET) THEN
         WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF RELATED DATA SETS'
         CALL TRMNAT
      ENDIF

      WRITE(6,935) NSETS

      DO 200 J = 1, NSETS

         COUNT = 0

         IF (IOUT. EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS

         READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)

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

         WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
         IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)

         WRITE(6,905)
         IF (IOUT .EQ. 10) WRITE(8,905)

         DO 300 M = 1, NDIV

            READ(5,*) NUM, RATIO, REG

            IF (ABS(RATIO) .GT. 1.0) THEN
               WRITE(8,*) 'ERROR:  INVALID VALUE OF RATIO:  ', RATIO
               CALL TRMNAT
            ENDIF

            IF (REG .GT. MAXREG) THEN
               WRITE(8,*)
     &            'ERROR:  OVER REGION LIMIT IN RELATED MATERIAL ', J
               CALL TRMNAT
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
               WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
            ENDIF

            DO 310 I = (COUNT + 1), (COUNT + NUM)
```

```
               READ(5,*) RAWSTR(I,J), RAWNF(I,J)
  310       CONTINUE

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,I2)

  905 FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
     &       /,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
     &       'STRESS',7X,'LIFE'/)

  910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)
```

```
 913 FORMAT(//)

 914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN So OF',
    &        5X,E11.5)

 915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA  ',
    &        /,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
    &        /,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
    &        '(CYCLES): ',/)

 920 FORMAT(10X,E9.3)

 925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
    &        'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
    &        //,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
    &        //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
    &        5X,'UPPER BOUND',/)

 930 FORMAT(6X,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***********************************************************************


C  THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C  STRESS RATIO, R, IS NOT -1.0
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:   6OCT87      COMMENTS:   13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
```

```
C
C  FTU        ULTIMATE STRENGTH OF MATERIAL (PSI)
C  FTY        YIELD STRENGTH OF MATERIAL (PSI)
C  I          CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  J          DATA SET OF INTEREST
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXSET     MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  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 * (1 - R)/2 , TO BE COMPARED WITH FTY

C  Kt IS ASSUMED TO BE ONE

      DO 100 I = NUM1, NUM2

         TEST = STR(I,J) * (1.0 - R)/2.0

         IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST

         IF (TEST .GE. FTY) THEN

            RSTR(I,J) = TEST

            IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() =',RSTR(I,J)

         ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN

            RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))

            IF (IOUT.EQ.10) WRITE(8,*)'2:RSTR() =',RSTR(I,J)

         ELSE

            RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
     &                  /(2.0 * FTU)))

            IF (IOUT.EQ.10) WRITE(8,*)'3:RSTR() =',RSTR(I,J)

         END IF

  100 CONTINUE

      RETURN
      END


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


C  SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C  AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C  X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  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               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   NUMREG, NSETS, NP, LNSTR, LNNF
C  OUTPUTS:  SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR

C     IMPLICIT NONE
```

```
      INTEGER 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)
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                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
C               L AND DATA SET J (X = Ln S)
C  MEANY()    1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
C               L AND DATA SET J (Y = Ln N)
C  NP()       2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C               SET IN EACH REGION
C  NPPR()     1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C               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
C               REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SWHAT2()   1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C               REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
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  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
         DO 200 J = 0, NSETS
C           FIRST CALCULATE SAMPLE X AND Y MEANS
C           FOR DATA SET J IN REGION L
            MEANY(J) = 0.0
            MEANX(J) = 0.0
            IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
     &                 ' NP =', NP(J,L)

            DO 250 K = 1, NP(J,L)
               MEANY(J) = MEANY(J) + LNNF(K,J,L)
               MEANX(J) = MEANX(J) + LNSTR(K,J,L)
               IF (IOUT .EQ. 10) WRITE(8,*) 'LNNF =', LNNF(K,J,L),
     &            ' LNSTR =', LNSTR(K,J,L)
  250       CONTINUE

            MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
            MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
            IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY(J) =', MEANY(J),
     &         ' MEANX(J) =', MEANX(J)

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

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

            NPPR(L) = NPPR(L) + NP(J,L) - 1
            IF (IOUT .EQ. 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
C           OUR MODEL
            WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
            CALL TRMNAT
         ENDIF

         NPPR(L) = NPPR(L) - 1

         IF (NPPR(L) .LE. 0) THEN
            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))

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,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
     &                 ' SX2(L) =', SX2(L)
```

```
            WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
     &                 ' BB(L) =', BB(L)
         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) =',
     &            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 INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, Io, ON
C  C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:   5OCT87    COMMENTS:   15SEP89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                   JZERO, MCHAT)

C  INPUTS:    NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C  OUTPUTS:   IZERO, JZERO, MCHAT
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER CHITAB, MAXREG, TTAB

      PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)

      COMMON  IOUT

      INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG

      REAL     ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
     &         IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
     &         SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
     &         SX2(MAXREG), T, T025(TTAB)

      DATA (CHI025(I), I = 1, 75) /
     &               0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
     &               1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
     &               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,
```

```
     &              29.15,      29.95,     30.75,     31.55,     32.3574,
     &              33.16,      33.96,     34.77,     35.58,     36.39,
     &              37.21,      38.02,     38.84,     39.66,     40.4817,
     &              41.30,      42.12,     42.95,     43.77,     44.60,
     &              45.43,      46.26,     47.09,     47.92,     48.7576,
     &              49.59,      50.42,     51.26,     52.10,     52.94 /
      DATA (CHI025(I), I = 76, 150) /
     &              53.78,      54.62,     55.46,     56.30,     57.1532,
     &              57.80,      58.84,     59.69,     60.54,     61.39,
     &              62.24,      63.09,     63.94,     64.79,     65.6466,
     &              66.50,      67.35,     68.21,     69.07,     69.92,
     &              70.78,      71.64,     72.50,     73.36,     74.2219,
     &              75.08,      75.94,     76.80,     77.67,     78.53,
     &              79.40,      80.27,     81.13,     82.00,     82.87,
     &              83.73,      84.60,     85.47,     86.34,     87.21,
     &              88.08,      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 /

      DATA (CHI975(I), I = 1, 75) /
     &              5.02389,    7.37776,   9.34840,   11.1433,   12.8325,
     &              14.4494,    16.0128,   17.5346,   19.0228,   20.4831,
     &              21.9200,    23.3367,   24.7356,   26.1190,   27.4884,
     &              28.8454,    30.1910,   31.5264,   32.8523,   34.1696,
     &              35.4789,    36.7807,   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,
     &              54.44,      55.67,     56.89,     58.12,     59.3417,
     &              60.56,      61.77,     62.99,     64.20,     65.41,
     &              66.62,      67.82,     69.02,     70.22,     71.4202,
     &              72.61,      73.81,     75.00,     76.19,     77.38,
     &              78.57,      79.75,     80.93,     82.12,     83.2976,
     &              84.48,      85.65,     86.83,     88.00,     89.18,
     &              90.35,      91.52,     92.69,     93.86,     95.0231,
     &              96.19,      97.35,     98.52,     99.68,     100.84 /
      DATA (CHI975(I), I = 76, 150) /
     &              102.00,     103.16,    104.31,    105.47,    106.629,
     &              107.78,     108.94,    110.09,    111.24,    112.39,
     &              113.54,     114.69,    115.84,    116.99,    118.136,
     &              119.28,     120.43,    121.57,    122.72,    123.86,
     &              125.00,     126.14,    127.28,    128.42,    129.561,
     &              130.70,     131.84,    132.98,    134.11,    135.25,
     &              136.38,     137.52,    138.65,    139.79,    140.92,
     &              142.05,     143.18,    144.31,    145.44,    146.57,
     &              147.70,     148.83,    149.96,    151.09,    152.21,
     &              153.34,     154.47,    155.59,    156.72,    157.84,
     &              158.97,     160.09,    161.21,    162.33,    163.46,
     &              164.58,     165.70,    166.82,    167.94,    169.06,
     &              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 /

C  VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:
C
C     1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719
C
C     31-39, 41-49, 51-59, 61-69, 71-79, 81-89, 91-99, 101-150
C        -- CALCULATED USING CUBE RULE APPROXIMATION

      DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
     &             2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
     &             2.160, 2.145, 2.131, 2.120, 2.110, 2.101,
     &             2.093, 2.086, 2.080, 2.074, 2.069, 2.064,
     &             2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /


C                         LIST OF VARIABLES
C
```

```
C  ARG          INTERMEDIATE CALCULATION VARIABLE
C  CHI025()     TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHI975()     TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHITAB       MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
C  DD()         1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  I            CONTROLS LOOP FOR CHI025() AND CHI975()
C  IOUT         OUTPUT DUMP CONTROLLER
C  IZERO()      2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                 FOR EACH REGION
C  JZERO()      2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C  L            CONTROLS DO LOOP FOR EACH REGION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                 MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
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  NUM          EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  SUHAT        EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SUHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT        EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SWHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SX           EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
C  SX2()        1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                 (X = Ln S)
C  T            VALUE OF T025() USED IN CALCULATIONS
C  T025()       TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
C  TTAB         MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025


C  INITIALIZE IZERO, JZERO AND MCHAT

      DO 50 L = 1, MAXREG
         IZERO(1,L) = 0.0
         IZERO(2,L) = 0.0
         JZERO(1,L) = 0.0
         JZERO(2,L) = 0.0
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
   50 CONTINUE

C  CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

      DO 75 L = 1, NUMREG
         IF (NPPR(L) .GT. CHITAB) THEN
            WRITE(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,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
     &                   IZERO(2,L)
             WRITE(8,*) 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
     &                   JZERO(2,L)
             WRITE(8,*) 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
     &                   MCHAT(2,L)
          ENDIF
  100 CONTINUE

      RETURN
      END


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


C  SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C  THE Co GIVEN BY THE USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  8OCT87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                          V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

C  INPUTS:    NUMREG, CZERO, SX2, SXY, SY2
C  OUTPUTS:   MCPNT, MC

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, MCPNT(MAXREG), NUMREG

      REAL     ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
     &         SXY(MAXREG), SY2(MAXREG)


C                     LIST OF VARIABLES
C
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     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         MCPNT(L) = 0
         MC(1,L) = 0.0
         MC(2,L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATIONS

      CZERO2 = CZERO ** 2

      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

      DO 100 L = 1, NUMREG

         ARG1 = SX2(L) - CZERO2
         ARG2 = 0.0

         IF (CZERO .EQ. 0.0) THEN

C           THEN NO M CONSTRAINT IS REQUIRED

            MCPNT(L) = 0

         ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

C           THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M

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

         ELSE

C           THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C           COMMON CALCULATIONS

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

            IF (SX2(L) .LT. CZERO2) THEN

C              AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

               MCPNT(L) = 1
               MC(1,L) = (- SXY(L) - ARG2) / ARG1

            ELSE

C              SX2(L) .GT. CZERO2 -- THIS TIME THE M CONSTRAINT IS A RANGE

               MCPNT(L) = 2
```

```
               MC(1,L) = (- SXY(L) - ARG2) / ARG1
               MC(2,L) = (- SXY(L) + ARG2) / ARG1

            ENDIF

         ENDIF

  100 CONTINUE

      IF (IOUT .EQ. 10) THEN

         DO 200 L = 1, NUMREG

            WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
            WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

  200    CONTINUE

      ENDIF

      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
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM 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

      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


C                      LIST OF VARIABLES
C
C  IOUT          OUTPUT DUMP CONTROLLER
C  J             CONTROLS DO LOOP FOR EACH DATA SET
C  L             CONTROLS DO LOOP FOR EACH REGION
C  LAMN          LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                   CONSTANT OVER REGIONS AND COMPONENTS
C  MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET        MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  MCHAT()       2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                   FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                   MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C                   MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  NP()          2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                   SET IN EACH REGION
```

```
C  NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM()        EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
C                 NUMBER OF POINTS IN EACH DATA SET
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  PSIG2()      1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C                 VARIATION IN EACH REGION
C  PVAR         THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                 CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  SUM          WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
C                 AVERAGE
C  TOTAL        SUM OF NUM() OVER ALL REGIONS

C     INITIALIZE VARIABLES

      SUM = 0.0
      TOTAL = 0.0

      DO 50 L = 1, MAXREG
         PSIG2(L) = 0.0
         NUM(L) = 0
   50 CONTINUE

      DO 100 L = 1, NUMREG
         DO 150 J = 0, NSETS
            NUM(L) = NUM(L) + NP(J,L)
  150    CONTINUE
         NUM(L) = NUM(L) - 1
         TOTAL = TOTAL + NUM(L)
  100 CONTINUE

      DO 200 L = 1, NUMREG
         PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
         SUM = SUM + PSIG2(L) * NUM(L)
  200 CONTINUE

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'LAMN = ', LAMN
         DO 300 L = 1, NUMREG
            WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
            WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
  300    CONTINUE
         WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
      ENDIF

      PVAR = SUM / FLOAT (TOTAL)

      RETURN
      END

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


C  SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C  M AND Co WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C  TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
     &                   MCHAT, RANGEM)

C  INPUTS:    NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG
```

```
      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL     JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
     &         MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

C                              LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  JZERO()      2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LOWER        LOWER BOUND OF INTERSECTION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()         2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                 REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                 -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                 BOUND
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION -- MCHAT(1,L) = - DD(L), THE ESTIMATE
C                 FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  UPPER        UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

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

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

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

         IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS NO EXOGENOUS INFORMATION
C           ASSUME RANGE TO BE Jo

            RANGEM(1,L) = JZERO(1,L)
            RANGEM(2,L) = JZERO(2,L)

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' JZERO(1,L) = ', JZERO(1,L)
               WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

C           NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
C           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

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

C        THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
C        CORRESPONDING TO THE Co CONSTRAINT, ADJUST Jo ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

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

C        THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER
C        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
C        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)
            WRITE(8,*) '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

C        THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C        CONSTRAINT, INTERSECT Jo AND Mo, ADJUSTING THE LOWER BOUND
C        BY Mc ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &            'AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &            ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &            ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANEGM(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
C        INTERSECT THESE TWO RANGES WITH Jo

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &            'AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &            ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &            ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &            ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSE

         WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
         CALL TRMNAT

      ENDIF

C     RESTRICT RANGE TO BE NON-NEGATIVE

      RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)
```

```
      IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)
  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END


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


C  SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C  WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  INPUTS:   RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C  OUTPUTS:  RANGEM, MCHAT, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

      REAL    MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)


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

```
      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           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

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            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


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
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONCAV (NUMREG, RANGEM)

C  INPUTS:    NUMREG, RANGEM
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    RANGEM(2, MAXREG), TESTM


C                           LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
```

```
C  L             CONTROLS DO LOOP FOR EACH REGION
C  MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUMREG        NUMBER OF REGIONS OF INTEREST
C  RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                  FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                  RANGEM(2,L) IS THE UPPER BOUND
C  TESTM        UPPER BOUND OF RANGE ON M IN REGION L-1 -- USED DURING
C                 CONCAVITY ADJUSTMENT

C     ADJUST RANGE TO INSURE CONCAVITY

      DO 100 L = NUMREG, 2, -1

         IF (RANGEM(2,L-1) .EQ. 0.0) THEN
C           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
C           RANGE IS AN INTERVAL IN REGION L-1
            TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
            IF (TESTM .LT. RANGEM(1,L-1)) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &            ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
     &            'REGION ', L-1
               CALL TRMNAT
            ELSE
               RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
            ENDIF
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L-1) =', RANGEM(1,L-1),
     &                 ' RANGEM(2,L-1) =', RANGEM(2,L-1)
            WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                 ' RANGEM(2,L) =', RANGEM(2,L)
            WRITE(8,*) 'TESTM =', TESTM, ' L = ', L
         ENDIF

  100 CONTINUE

      RETURN
      END


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


C  SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C  BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5OCT87      COMMENTS:  1DEC87
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C  INPUTS:    NUMREG, RANGEM
C  IOUTPUT:   MEDM

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT
```

```
      INTEGER  IOUT, L, NUMREG
      REAL     LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)

C                   LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LOWERM       LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
C                 TO BE USED IN MEDIAN CALCULATION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEDM()       1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND

C     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
C           -- 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)
            WRITE(8,*) '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
C     VERSION:  MATCHR V8.3, V8.4, V8.5   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 EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
     &                   ZROREG, NBND, BIGK1, BZHAT)

C  INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C  OUTPUTS:  BIGK1, BZHAT
C  SUBPROGRAMS:  TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL     BIGK(0:MAXREG), BIGK1, 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)


C                   LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  BIGK1      EQUAL TO BIGK(1)
C  BZHAT      E(BETAO)
C  FACTR      A SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  KHAT       E(k)
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MEDM()     1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  MM()       1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NCOMPS     Number of COMPonents -- 1 FOR STRESS AND STRAIN WHEN DECOMPOSED
C               DATA UNAVAILABLE -- 2 FOR DECOMPOSED STRAIN DATA
C  NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C               SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                DATA SET
C  NPTS()     1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR
C                THE SPECIFIC MATERIAL S/N DATA SET
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 REGION
C               CONTAINED IN NBND()
C  STR()      2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C               DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZ2        SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  SZERO      STRESS TENSILE TEST POINT, So
C  TRBIGK()   1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C               TRBIGK(i) = BIGK(i)
C  ZROREG     ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C               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  INITIALIZE VARIABLES

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

```
C  CREATE MM() ARRAY FROM MEDM() ARRAY

      DO 100 L = 1, NUMREG
         MM(L) = MEDM(L)
  100 CONTINUE

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

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

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE BETAo AND k

      CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C  CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

      CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

      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), SBND(L)
            IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
     &                                     NBND(L), SBND(L)
  200    CONTINUE

         WRITE(7,920)

      ELSE

         WRITE(7,930) MM(1), BIGK(1), KHAT

      ENDIF

C  FORMAT STATEMENTS

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

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

  920 FORMAT(///)

  930 FORMAT(//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &       //,11X,'m',14X,'K',13X,'E(k)',
```

```
      &       //,7X,F8.5,5X,E12.5,6X,F7.4,/)

        RETURN
        END

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

C  SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C  MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88     COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      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
C  ARG        INTERMEDIATE CALCULATION VARIABLE
C  DD()       1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
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               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  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 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  SUMX2      EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
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     INITALIZE ARRAYS
```

```
      DO 50 L = 1, MAXREG
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
         MU(L) = 0.0
         SIG(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATION FOR EACH REGION

      DO 100 L = 1 , NUMREG

         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SQRT (SUHAT2(L))
         SUMX2 = NPPR(L) * SX2(L)
         ARG = SUMX2 + DELTA(L)

         IF (DELTA(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C           USE THE ESTIMATE OF M
            MU(L) = MCHAT(1,L)
         ELSE
C           UPDATE THE ESTIMATE OF M WITH Mo USING DELTA
            MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
         ENDIF

         IF (SIGMA2(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
C           USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
            SIG(L) = SQRT (SWHAT2(L) / ARG)
         ELSE
            SIG(L) = SQRT (SIGMA2(L) / ARG)
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
     &                 MCHAT(1,L)
            WRITE(8,*) 'SUHAT2 = ', SUHAT2(L),  ' MCHAT2 = ',
     &                 MCHAT(2,L)
            WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
     &                 ' SUMX2 = ', SUMX2
            WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
            WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
            WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
     &                 ' SIG = ', SIG(L)
         ENDIF

  100 CONTINUE

      RETURN
      END

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     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  INPUTS:   NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG
```

```
      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
     &        RANGEM(2, MAXREG), UPPER

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

C     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
C           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 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. 1)) THEN

C      THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C      CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc

       LOWER = AMAX1(MZERO(1,L), MC(1,L))
       UPPER = MZERO(2,L)
       IF (UPPER .LT. LOWER) THEN
          WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
          CALL TRMNAT
       ELSE
          RANGEM(1,L) = LOWER
          RANGEM(2,L) = UPPER
       ENDIF

       IF (IOUT .EQ. 10) THEN
          WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &               ' MZERO(2,L) = ', MZERO(2,L)
          WRITE(8,*) 'MC(1,L) = ', MC(1,L)
          WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
          WRITE(8,*) 'RANEGM(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
C      INTERSECT THESE TWO RANGES

       LOWER = AMAX1(MZERO(1,L), MC(1,L))
       UPPER = AMIN1(MZERO(2,L), MC(2,L))
       IF (UPPER .LT. LOWER) THEN
          WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
          CALL TRMNAT
       ELSE
          RANGEM(1,L) = LOWER
          RANGEM(2,L) = UPPER
       ENDIF

       IF (IOUT .EQ. 10) THEN
          WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &               ' MZERO(2,L) = ', MZERO(2,L)
          WRITE(8,*) 'MC(1,L) = ', MC(1,L)
          WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
          WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &               ' RANGEM(2,L) = ', RANGEM(2,L)
       ENDIF

    ELSE

       WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
       CALL TRMNAT

    ENDIF

C   RESTRICT RANGE TO BE NON-NEGATIVE

    RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

    IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

100 CONTINUE

C   CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

    DO 300 L = 1, NUMREG
```

```
      IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &     .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &   WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &              'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END


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


C  SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C  DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88     FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:    RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C             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)


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

      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
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            MU(LL) = MO(LL)
            SIG(LL) = SQRT(SIGMA2(LL))
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
               WRITE(8,*) 'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
               WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
     &                    SIGMA2(LL)
            ENDIF

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
               MU(LL) = RANGEM(1,LL)
               SIG(LL) = 0.0
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
     &         ' MU = ', MU(LL), ' SIG = ', SIG(LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END


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


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               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  INPUTS:    VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
C             NBND, STR
C  OUTPUTS:   BIGK, BZERO, MM, SBND
C  SUBPROGRAMS:  FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB
```

```
C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG

      REAL    BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG),
     &        MU(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
     &        STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

      DOUBLE PRECISION RAND


C                        LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT    MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ     SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C               DISTRIBUTION MEAN FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NF()      2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C               SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP        TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()    1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C               SPECIFIC MATERIAL S/N DATA SET
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  RANGEM()  2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C               FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C               RANGEM(2,L) IS THE UPPER BOUND
C  RAND      RANDOM NUMBER SEED
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()
C  SIG()     1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
C               STANDARD DEVIATION FOR EACH REGION
C  STR()     2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C               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;
C               1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;
C               3 - TRUNCATED NORMAL VARIATION
C  ZROREG    ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C               BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C  ZZ()      1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C               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, 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,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. 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

      SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C   INPUTS:    RAND, NUMREG, RANGEM
C   OUTPUTS:   MM
C   SUBPROGRAMS:  RANDOM, TRMNAT

C      IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT
```

```
      INTEGER IOUT, L, NUMREG
      REAL    MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X
      DOUBLE PRECISION RAND

C                       LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  L            CONTROLS DO LOOP FOR EACH REGION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()         1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  PICK()       1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C  RAND         RANDOM NUMBER SEED
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  X            UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                 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)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              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),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF

  100 CONTINUE
```

```
      RETURN
      END

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


C*************************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C*************************************************************************

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

C                LIST OF VARIABLES
C
C  FRAC       UNIFORM (0,1) RANDOM VARIATE
C  IOUT       OUTPUT DUMP CONTROLLER
C  RANA       CONSTANT FOR LCG
C  RANC       CONSTANT FOR LCG
C  RAND       RANDOM NUMBER SEED
C  RANDIV     INTERNAL CALCULATION
C  RANM       CONSTANT FOR LCG
C  RANSUB     INTERNAL CALCULATION
C  RANT       INTERNAL CALCULATION
C  RANX       INTERNAL CALCULATION


C          USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

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

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

      RETURN
      END

C     NOTES:  IOUT=2 DUMPS TO SCREEN
```

```
C***********************************************************************

C  SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13FEB89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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


C                        LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()         1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()        1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  PICK()       1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C  RAND        RANDOM NUMBER SEED
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C  SIG()       1-D ARRAY CONTAINING THE STANDARD DEVIATION OF M FOR EACH
C                REGION
C  X           NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                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
   10       CALL NORMGN (RAND, MU(L), SIG(L), X)
```

```
            IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
            MM(L) = X
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                    ' RANGEM(2,L) =', RANGEM(2,L)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              SAMPLE ON ADJUSTED RANGE
   20          CALL NORMGN (RAND, MU(L), SIG(L), X)
               IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
               MM(L) = X
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF

  100 CONTINUE

      RETURN
      END


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


C***********************************************************************
C  SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C  WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  3FEB88
C     VERSION:  MATCHR V7, V7.1, 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
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C***********************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT
```

```
C              LIST OF VARIABLES
C
C  FRAC      UNIFORM(0,1) RANDOM VARIATE
C  IOUT      OUTPUT DUMP CONTROLLER
C  MU        MEAN OF NORMAL DISTRIBUTION
C  RAND      RANDOM NUMBER SEED
C  SIGMA     STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X         NORMAL RANDOM VARIATE
C  U1        UNIFORM RANDOM NUMBER U(0,1)
C  U2        UNIFORM RANDOM NUMBER U(0,1)
C  Z1        NORMAL RANDOM NUMBER ON N(0,1)
C  Z2        NORMAL RANDOM NUMBER ON N(0,1)

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

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

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

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

      X = SIGMA * Z1 + MU
      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 V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   NPTS, STR, NF, NUMREG, MM, NBND
C  OUTPUTS:  NP, ZZ

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON IOUT

      INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG

      REAL      MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &          STR(MAXDAT, MAXREG), ZZ(MAXDAT)

C                     LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR EACH DATA POINT
C  IOUT       OUTPUT DUMP CONTROLLER
C  K          CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
```

```
C  LL          CONTROLS INNER DO LOOP FOR EACH REGION
C  MAXDAT      MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()        1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C  MML         EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()      1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C                SPECIFIC MATERIAL S/N DATA SET
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  ZZ()        1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
C                Z = F(STR,NF,NBND,MM)

C  INITIALIZE VARIABLES

      NP = 0

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

C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG
         MML = MM(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =',
     &       MML, ' NPTS =', NPTS(L)

         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))
     &                  * ((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) =',
     &             MM(LL), ' ZZ =', ZZ(NP)
  300       CONTINUE

  200    CONTINUE

  100 CONTINUE

      RETURN
      END

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

C  SUBROUTINE SMNVAR CALCULATES THE Sample MeaN and VARiance OF
C  Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  24AUG87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V3.3, V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C               DATA SET
C  SZ2        SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZZ()       1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C                Z = F(STR,NF,NBND,MM)


C  INITIALIZE VARIABLES

       MEANZ = 0.0
       SZ2 = 0.0

C  CALCULATE THE MEAN OF ZZ(), MEANZ

       DO 100 I = 1, NP
          MEANZ = MEANZ + ZZ(I)
          IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I,
     &      ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
  100 CONTINUE
       MEANZ = MEANZ / FLOAT(NP)
       IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ

C  CALCULATE THE VARIANCE OF ZZ(), SZ2

       DO 200 I = 1, NP
          SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
          IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
  200 CONTINUE
       SZ2 = SZ2 / FLOAT(NP - 1)
       IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

       RETURN
       END


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


C  SUBROUTINE KBETA CALCULATES k AND BETAo FROM THE SAMPLE MEAN AND
C  VARIANCE OF Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

       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

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

C  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
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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)

C                        LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
C              MATERIAL DATA BASE
C  GAMMA     EULER'S CONSTANT
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C              DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST

C  INITIALIZE VARIABLES

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

C  CALCULATE K FOR REGION ONE

      BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C       WRITE(7,*) 'REGION:  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)))
C         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
C        DATE:  22DEC88
C     VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.2, V4.3, V4.4, V4.5

      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)

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

C  INITIALIZE SBND()

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

C   CALCULATE SBND(0) IF ZROREG = 0

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

C  CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES

      DO 100 L = 1, NUMREG
         IF (NBND(L) .GE. 1.0E+36) THEN
            SBND(L) = 0.0
         ELSE
            SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
         ENDIF
  100 CONTINUE

      RETURN
      END


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


C  THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
C  MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C  TRANSFORM METHOD"
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  18MAR87     COMMENTS:  15SEP89
C     VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                      V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                      V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
```

```
      SUBROUTINE WEIBGN (BETA, RAND, WEIB)

C  INPUTS:    BETA, RAND
C  OUTPUTS:   WEIB
C  SUBPROGRAMS:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER IOUT

      REAL    ARG, BETA, ETA, FRAC, WEIB

      DOUBLE PRECISION RAND


C              LIST OF VARIABLES
C
C  ARG      INTERMEDIATE CALCULATION VARIABLE
C  BETA     WEIBULL DISTRIBUTION SHAPE PARAMETER
C  ETA      WEIBULL DISTRIBUTION LOCATION PARAMETER
C  FRAC     UNIFORM (0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  RAND     RANDOM NUMBER SEED
C  WEIB     WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE


C     CALCULATE CONSTRAINED ETA

      ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))

C     GENERATE WEIBULL RANDOM VARIATE

      CALL RANDOM(FRAC, RAND)
      ARG = -ALOG(1.0 - FRAC)
      WEIB = ETA * ARG**(1.0/BETA)
      IF (IOUT .EQ. 10) WRITE(8,*)'BETA = ', BETA, ' ETA =', ETA,
     &   ' FRAC =', FRAC, ' ARG =', ARG, ' WEIB =', WEIB

      RETURN
      END


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


C  SUBROUTINE KOMO CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA
C  REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
C  AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
C  PROGRAMMER :  L. NEWLIN
C        DATE:   1AUG91
C     VERSION:   MATCHR V8.5    MATGRM V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
     &                 FACTR, NUMREG)

C  INPUTS:    SZERO, BIGK, MM, NBND, TRSBND, FACTR
C  OUTPUTS:   TRBIGK, MM, TRSBND

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)
```

```
      COMMON    IOUT

      INTEGER   IOUT, L, NUMREG

      REAL      BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1          SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)


C                       LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  FACTR      SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NUMREG     NUMBER OF REGIONS
C  SCLK       ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
C  SZERO      STRESS TENSILE TEST POINT, So
C  TRBIGK()   1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
C               SBND(0) < So FOR EACH TRIAL
C  TRSBND()   1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
C               LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
C               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, ' BIGK1 = ', TRBIGK(1)
         WRITE(8,*) 'MM1 = ', MM(1), ' MMo = ', MM(0)
      ENDIF

      RETURN
      END

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


C  FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C  BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB89
C     VERSION:  MATCHR V8.3, V8.4, V8.5 -- FOR USE WITH PFM'S
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.


      REAL FUNCTION GTLIFE (S, MM, LNA, LPHIM, KRATIO, LNZ, SBND,
     &                      ZROREG, NUMREG, SZERO)

C  INPUTS:   S, MM, LNA, LPHIM, KRATIO, LNZ, SBND, ZROREG, NUMREG, SZERO
C  OUTPUTS:  GTLIFE
```

```
C       IMPLICIT NONE

        INTEGER IOUT, L, MAXREG, NUMREG, ZROREG

        PARAMETER (MAXREG = 3)

        COMMON  IOUT

        REAL    GETLIF, KRATIO, LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG),
       &        MM(0:MAXREG), S, SBND(0:MAXREG), SZERO, TEMP

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

        GETLIF = 0.0

C  CALCULATE CYCLES TO FAILURE

        IF ((S .GE. SZERO) .AND. (ZROREG .EQ. 0)) THEN
           GETLIF = 1.0
        ELSE
           DO 100 L = ZROREG, NUMREG
              IF (S .GT. SBND(L)) THEN
                 TEMP = LNA(L) + LPHIM(L) + MM(L) * ( - ALOG(S)
       &                     +  ALOG (KRATIO) + LNZ)
                 IF (TEMP .GT. 86.0) THEN
                    TEMP = 86.0
                 ENDIF
                 GETLIF = EXP (TEMP)
                 GOTO 150
              ENDIF
  100      CONTINUE
        ENDIF
  150 CONTINUE

        GTLIFE = GETLIF

        RETURN
        END


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


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

```
C       VERSION:  MATCHR V7, V7.1, 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
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, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

      PARAMETER (MAXMM = 20001, MAXREG = 3)

      LOGICAL INORDR

      REAL    ALLM(MAXMM, MAXREG), TEMP

C                    LIST OF VARIABLES
C
C  ALLM()      2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
C  I           CONTROLS INSERTION POINTER
C  INC         SORT INCREMENT VARIABLE
C  INORDR      FLAG TO INDICATE WHETHER SORT IS FINISHED
C  IOUT        OUTPUT DUMP CONTROLLER
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXMM       MAXIMUM NUMBER OF M'S TO BE SORTED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUM         NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  TEMP        TEMPORARY SORTING VARIABLE


      DO 400 L = 1, NUMREG

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

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

            IF (.NOT. INORDR) GOTO 20
            GOTO 10
         ENDIF

 400  CONTINUE

      RETURN
      END


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


C***************************************************************************
C  FUNCTION THWELD CONTROLS THE CALLS REQUIRED TO CALCULATE A LIFE FOR A
```

```
C  PLAIN WELD UNDER A THERMAL LOAD
C  PROGRAMMER:  L. NEWLIN
C        DATE:  11JUL90
C     VERSION:  THDUCT V4, V4.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C*********************************************************************

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


C  INPUTS: ALPHA, ANGLE, DLTAT, E, EM, FTU, FTY, DI, K, LAMW, FK, RT, LNA,
C          LOCAT, LPHIM, M, MM, MSTAT, NEUB, NLOAD, NRAN, NU, NUMREG,
C          NUMSEG, P, PC, PCO, PERIOD, PSTAT, SBND, SE, STRHIS, SZERO, T,
C          THIC, TRUNC, TSTAT, V, VSTAT, WOFF
C
C  OUTPUTS: THWELD
C
C  SUBPROGRAMS:  M4L1, M4L2, NARBN2, TRMNAT


C     IMPLICIT NONE

      INTEGER  MAXLD, MAXM, MAXREG, MAXSEG

      PARAMETER (MAXLD = 16, MAXM = 24000, MAXREG = 3, MAXSEG = 10)

      COMMON   IOUT

      INTEGER  IOUT, LOCAT, NLOAD, NUMSEG, NRAN, NUMREG, ZROREG

      REAL     ALPHA, ANGLE, DI, DLTAT, E(MAXSEG), EM, FATLIF, FK(10),
     &         FTY, FTU, K(2, 2), KRATIO, LAMW, LNA(0:MAXREG), LNZ,
     &         LPHIM(0:MAXREG), M(2, MAXLD), MM(0:MAXREG), MSTAT(2),
     &         NEUB, NU, P(MAXLD), PC, PCO, PERIOD, PSTAT, RT(10),
     &         SBND(0:MAXREG), SE(MAXSEG), STRAMP(4, MAXLD),
     &         STRHIS(MAXLD, MAXM), STATIC(4), SZERO, T(MAXLD), THIC,
     &         THWELD, TRUNC, TSTAT, WOFF, V(2, MAXLD), VSTAT(2)

C                 LIST OF VARIABLES
C
C  ALPHA       COEFFICIENT OF THERMAL EXPANSION
C  ANGLE       ANGLE THETA IN RADIANS
C  DI          INTERIOR DIAMETER
C  DLTAT       TEMPERATURE DIFFERENCE BETWEEN THE INNER AND OUTER SURFACES
C  E()         1-D ARRAY CONTAINING THE STRAIN VALUES
C  EM          YOUNG'S MODULUS BEFORE YIELD
C  FATLIF      VALUE OF LIFE CALCULATED
C  FK()        1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C                CONCENTRATION DUE TO WELD ECCENTRICITY
C  FTY         YIELD STRENGTH
C  FTU         ULTIMATE STRENGTH
C  IOUT        OUTPUT DUMP CONTROLLER
C  K()         FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT
C                EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR
C                FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL
C                DIRECTION; K(2,2) IS FOR DUCT INTERIOR FOR HOOP DIRECTION
C  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  LAMW        ACCURACY FACTOR OF Fk - r/t CURVE
C  LNA()       1-D ARRAY CONTAINING VALUES OF Ln(A) = Ln(K**m) FOR EACH
C                REGION
C  LNZ         NORMAL (0,PVAR) GENERATED FROM RANDOM VARIATE
C  LOCAT       LOCATION OF INTEREST WHERE 1 IS THE EXTERIOR SURFACE
```

```
C                2 IS THE INTERIOR SURFACE OF THE DUCT
C   LPHIM()     1-D ARRAY CONTAINING VALUES OF Ln(PHI)m FOR EACH REGION
C   M()         2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C                 ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C   MAXLD       MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C   MAXM        MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MAXSEG      MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN)
C   MM()        1-D ARRAY CONTAINING VALUES FOR m FOR EACH REGION
C   MSTAT()     1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE
C                 M2 LOAD; M(2) IS THE M3 LOAD
C   NEUB       NEUBER'S RULE MODEL ACCURACY FACTOR
C   NLOAD       NUMBER OF TIME-VARYING LOADS
C   NRAN        NUMBER OF POINTS IN STRESS-TIME HISTORY
C   NU          POISSON'S RATIO
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   NUMSEG      NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
C   P()         1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C   PC          LIMIT PRESSURE ON INSIDE OF THE TUBE
C   PCO         LIMIT PRESSURE ON OUTSIDE OF THE TUBE
C   PERIOD      LENGTH OF TIME IN SECONDS FOR RANDOM STRESS-TIME HISTORY
C   PSTAT       STATIC AXIAL LOAD
C   RT()        1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C                 CONCENTRATION DUE TO WELD ECCENTRICITY
C   SBND()      1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C                 REGION CORRECTED FOR PHI, KRATIO AND LNZ
C   SE()        1-D ARRAY CONTAINING THE STRESS AND STRAIN PRODUCTS
C   STATIC()    1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES --
C                 STATIC(1) IS THE AXIAL STRESS; STATIC(2) IS THE HOOP
C                 STRESS; STATIC(3) IS THE RADIAL STRESS; STATIC(4) IS THE
C                 SHEAR STRESS
C   STRAMP()    2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C                 -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C                 THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                 STRAMP(4,*) ARE THE SHEAR STRESSES
C   SZERO       STRESS TENSILE POINT (PSI)
C   T()         1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C   THIC        WALL THICKNESS AT DUCT OUTER RADIUS
C   THWELD      FUNCTION WHICH CALCULATES THE LIFR FOR THE PLAIN WELD UNDER
C                 THERMAL LOADS
C   TRUNC       VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
C   TSTAT       STATIC TORQUE LOAD
C   V()         2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                 ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C   VSTAT()     1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2
C                 LOAD; V(2) IS THE V3 LOAD
C   WOFF        WELD OFFSET
C   ZROREG      ZeRO REGion--VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE --0 - ZERO REGION EXISTS, 1-NO ZERO REGION


      IF (LOCAT .EQ. 1) THEN

         CALL M4L1 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT,
     &              NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP, T,
     &              THIC, TSTAT, V, VSTAT, WOFF, FK, RT)

         CALL NARBN2 (E, EM, FATLIF, FTU, FTY, 1.0, KRATIO, LNA, LNZ,
     &               LPHIM, MM, NRAN, NEUB, NLOAD, NUMREG, NUMSEG,
     &               SBND, PERIOD, SE, STATIC, STRAMP, STRHIS, SZERO,
     &               TRUNC, ZROREG)

      ELSE IF (LOCAT .EQ. 2) THEN

         CALL M4L2 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT,
     &              NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP, T,
     &              THIC, TSTAT, V, VSTAT, WOFF, FK, RT)
```

```
         CALL NARBN2 (E, EM, FATLIF, FTU, FTY, 1.0, KRATIO, LNA, LNZ,
     &                LPHIM, MM, NRAN, NEUB, NLOAD, NUMREG, NUMSEG,
     &                SBND, PERIOD, SE, STATIC, STRAMP, STRHIS, SZERO,
     &                TRUNC, ZROREG)

      ELSE

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

      ENDIF


      THWELD = FATLIF

      RETURN
      END


C***********************************************************************
C  SUBROUTINE M4L1 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
C  FOR  LOCATION 1 (PLAIN WELD, EXTERIOR SURFACE OF THE DUCT) UNDER THERMAL
C  LOADING
C  PROGRAMMER:  L. NEWLIN
C        DATE:  4MAR91
C     VERSION:  THDUCT V4.1
C***********************************************************************

      SUBROUTINE M4L1 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT,
     &                 NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP,
     &                 T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT)

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


C     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), KOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU,
     &         P(MAXLD), PC, PCO, PSTAT, RDIFF, RI, RI2, RO, RO2,
     &         ROT, RT(10), SIG1A(MAXLD), SIG1B(MAXLD), SKT1, SKT2,
     &         STATIC(4), STHMA, STR1A, STR2A, STR1B, STR2B, STR1C,
     &         STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD),
     &         VSTAT(2), WOFF


C                     LIST OF VARIABLES
C
C  ALPHA      COEFFICIENT OF THERMAL EXPANSION
C  ANGLE      ANGLE THETA IN RADIANS
C  AREA       CROSS SECTION AREA OF DUCT WALL
C  DI         INTERIOR DIAMETER
C  DLTAT      TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
C  EM         YOUNG'S MODULUS PRIOR TO YIELD
C  FK()       1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C               CONCENTRATION DUE TO WELD ECCENTRICITY
C  GEOM       INTERMEDIATE THERMAL STRESS CALCULATION VARIABLE
C  I          CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND
C               AERODYNAMIC LOADS
C  IFK        INTERPOLATED VALUE OF Fk CORRESPONDING TO THE VALUE OF r/t
```

```
C  IOUT      OUTPUT DUMP CONTROLLER
C  J         CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING
C              INTERPOLATION
C  K()       FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT
C              EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR
C              FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL
C              DIRECTION; K(2,2) IS FOR DUCT INTERIOR FOR HOOP DIRECTION
C  KOFF      STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
C  LAMW      ACCURACY FACTOR OF Fk - r/t CURVE
C  M()       2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C              ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C  MAXLD     MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C  MI        MOMENT OF INERTIA
C  MSTAT()   1-D ARRAY CONTAINING THE STATIC MOMENT LOADS -- M(1) IS THE M
C              2 LOAD; M(2) IS THE M3 LOAD
C  NLOAD     NUMBER OF TIME-VARYING LOADS
C  NU        POISSON'S RATIO
C  P()       1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C  PC        LIMIT PRESSURE ON INSIDE OF THE VESSEL
C  PCO       LIMIT PRESSURE ON OUTSIDE OF THE VESSEL
C  PI        SELF EXPLANATORY CONSTANT
C  PSTAT     STATIC AXIAL LOAD
C  RDIFF     EQUAL TO RO2 - RI2
C  RI        INTERIOR RADIUS
C  RI2       INNER RADIUS SQUARED
C  RO        OUTER RADIUS
C  RO2       OUTER RADIUS SQUARED
C  ROT       EQUAL TO r / t (R Over T)
C  RT()      1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C              CONCENTRATION DUE TO WELD ECCENTRICITY
C  SIG1A()   1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO FORCE
C              FOR THE TIME-VARYING LOADS
C  SIG1B()   1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO BENDING
C              FOR THE TIME-VARYING LOADS
C  SKT1      STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
C  SKT2      STRESS CONCENTRATION FACTOR FOR HOOP STRESS
C  STATIC()  1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES --
C              STATIC(1) IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS;
C              STATIC(3) IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C  STHMA     THE STATIC AXIAL STRESS DUE TO THERMAL GRADIENT
C  STR1A     THE STATIC AXIAL STRESS DUE TO FORCE
C  STR1B     THE STATIC AXIAL STRESS DUE TO BENDING
C  STR1C     THE STATIC AXIAL STRESS DUE TO MOMENTUM CHANGE (FLUID)
C  STR2A     THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO INTERNAL PRESSURE
C  STR2B     THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO EXTERNAL PRESSURE
C  STRAMP()  2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C              -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C              THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C              STRAMP(4,*) ARE THE SHEAR STRESSES
C  T()       1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C  THIC      WALL THICKNESS AT DUCT OUTER RADIUS
C  TSTAT     STATIC TORQUE LOAD
C  V()       2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C              ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C  VSTAT()   1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2
C              LOAD; V(2) IS THE V3 LOAD
C  WOFF      WELD OFFSET

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

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

      DO 50 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)) * (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 RADII APPROPRIATE TO LOCATION
C     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
         WRITE(8,*)
      ENDIF


      RI2 = RI ** 2
      RO2 = RO ** 2
      RDIFF = RO2 - RI2

      GEOM = 1.00 - 2.00 * LOG (RO / RI) * RI2 / RDIFF

C     TEMPERATURE STRESS

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

C     AXIAL STRESS CALCULATIONS

      STR1A = PSTAT / AREA
      STR1B = (MSTAT(1) * COS (ANGLE) + MSTAT(2) * SIN (ANGLE)) * RO
     &        / MI
      STR1C = (PC - PCO) * RI2 / RDIFF

      STATIC(1) = (STR1A + STR1B + STR1C) * SKT1 * KOFF + STHMA

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

      STR2A = 2.0 * PC * RI2 / RDIFF
      STR2B = - PCO * (RO2 + RI2) / RDIFF

      STATIC(2) = (STR2A + STR2B) * SKT2 + STHMA

      STATIC(3) = - PCO

C     SHEAR STRESS

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


      IF (IOUT.EQ.25) THEN
         WRITE(8,*) 'RO2 = ', RO2, ' RI2 = ', RI2
         WRITE(8,*) 'RDIFF = ', RDIFF, ' GEOM = ', GEOM
         WRITE(8,*) 'STATIC STRESS VALUES '
         WRITE(8,*) 'AXIAL STRESSES'
         WRITE(8,*) '    STR1A = ', STR1A, ' STR1B = ', STR1B
         WRITE(8,*) '    STR1C = ', STR1C, ' STHMA = ', STHMA
         WRITE(8,*) '    STATIC(1) = ', STATIC(1)
```

```
         WRITE(8,*) 'HOOP STRESSES'
         WRITE(8,*) '    STR2A = ', STR2A, ' STR2B = ', STR2B,
     &              ' STHMA = ', STHMA
         WRITE(8,*) '    STATIC(2) = ', STATIC(2)
         WRITE(8,*) 'RADIAL STRESS', '    STATIC(3) = ', STATIC(3)
         WRITE(8,*) 'SHEAR STRESS', '    STATIC(4) = ', STATIC(4)
         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        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,*) '    STRAMP(1,I) = ', STRAMP(1,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


C*****************************************************************************
C  SUBROUTINE M4L2 PERFORMS THE CALCULATIONS NECESSARY TO FIND THE STRESS
C  FOR LOCATION 2 (PLAIN WELD, INTERIOR SURFACE OF THE DUCT), UNDER
C  THERMAL LOADING
C  PROGRAMMER:  L. NEWLIN
C        DATE:  11JUL90
C     VERSION:  THDUCT V4, V4.1
C*****************************************************************************

      SUBROUTINE M4L2 (ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT,
     &                 NLOAD, NU, P, PC, PCO, PSTAT, STATIC, STRAMP,
     &                 T, THIC, TSTAT, V, VSTAT, WOFF, FK, RT)
```

```
C  INPUTS:  ALPHA, ANGLE, DLTAT, EM, DI, K, LAMW, M, MSTAT, NLOAD, NU,
C           P, PC, PCO, PSTAT, T, THIC, TSTAT V, VSTAT, WOFF, FK, RT
C
C  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, DLTAT, EM, FK(10), GEOM, IFK, DI,
     &          K(2, 2), KOFF, LAMW, M(2, MAXLD), MI, MSTAT(2), NU,
     &          P(MAXLD), PC, PCO, PSTAT, RDIFF, RI, RI2, RO, RO2,
     &          ROT, RT(10), SIG1A(MAXLD), SIG1B(MAXLD), SKT1, SKT2,
     &          STATIC(4), STHMA, STR1A, STR2A, STR1B, STR2B, STR1C,
     &          STRAMP(4, MAXLD), T(MAXLD), THIC, TSTAT, V(2, MAXLD),
     &          VSTAT(2), WOFF


C                LIST OF VARIABLES
C
C  ALPHA      COEFFICIENT OF THERMAL EXPANSION
C  ANGLE      ANGLE THETA IN RADIANS
C  AREA       CROSS SECTION AREA OF DUCT WALL
C  DI         INTERIOR DIAMETER
C  DLTAT      TEMPERATURE DIFFERENCE BETWEEN INNER AND OUTER SURFACES
C  EM         YOUNG'S MODULUS PRIOR TO YIELD
C  FK()       1-D ARRAY CONTAINING VALUES OF Fk USED TO FIND STRESS
C               CONCENTRATION DUE TO WELD ECCENTRICITY
C  GEOM       INTERMEDIATE THERMAL STRESS CALCULATION VARIABLE
C  I          CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND
C               AERODYNAMIC LOADS
C  IFK        INTERPOLATED VALUE OF Fk CORRESPONDING TO THE VALUE OF r/t
C  IOUT       OUTPUT DUMP CONTROLLER
C  J          CONTROLS DO LOOP FOR EACH POINT IN RT() AND FK() DURING
C               INTERPOLATION
C  K()        FATIGUE STRESS CONCENTRATION FACTORS -- K(1,1) IS FOR DUCT
C               EXTERIOR FOR AXIAL DIRECTION; K(2,1) IS FOR DUCT EXTERIOR
C               FOR HOOP DIRECTION; K(1,2) IS FOR DUCT INTERIOR FOR AXIAL
C               DIRECTION; K(2,2) IS FOR DUCT INTERIOR FOR HOOP DIRECTION
C  KOFF       STRESS CONCENTRATION FACTOR DUE TO ECCENTRICITY OF WELD
C  LAMW       ACCURACY FACTOR OF Fk - r/t CURVE
C  M()        2-D ARRAY CONTAINING THE TIME-VARYING MOMENT LOADS -- M(1,*)
C               ARE THE M2 LOADS; M(2,*) ARE THE M3 LOADS
C  MAXLD      MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C  MI         MOMENT OF INERTIA
C  MSTAT()    1-D ARRAY CONTAINING THE STATIC TIME-VARYING LOADS -- M(1) IS
C               THE M2 LOAD; M(2) IS THE M3 LOAD
C  NLOAD      NUMBER OF TIME-VARYING LOADS
C  NU         POISSON'S RATIO
C  P()        1-D ARRAY CONTAINING THE TIME-VARYING AXIAL LOADS
C  PC         LIMIT PRESSURE ON INSIDE OF THE VESSEL
C  PCO        LIMIT PRESSURE ON OUTSIDE OF THE VESSEL
C  PI         SELF EXPLANATORY CONSTANT
C  PSTAT      STATIC AXIAL LOAD
C  RDIFF      EQUAL TO RO2 - RI2
C  RI         INTERIOR RADIUS
C  RI2        INNER RADIUS SQUARED
C  RO         OUTER RADIUS
C  RO2        OUTER RADIUS SQUARED
C  ROT        EQUAL TO r / t (R Over T)
C  RT()       1-D ARRAY CONTAINING VALUES OF r/t USED TO FIND STRESS
C               CONCENTRATION DUE TO WELD ECCENTRICITY
C  SIG1A()    1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO FORCE
C               FOR THE TIME-VARYING LOADS
C  SIG1B()    1-D ARRAY CONTAINING VALUES OF THE AXIAL STRESS DUE TO BENDING
C               FOR THE TIME-VARYING LOADS
```

```
C  SKT1        STRESS CONCENTRATION FACTOR FOR AXIAL STRESS
C  SKT2        STRESS CONCENTRATION FACTOR FOR HOOP STRESS
C  STATIC()    1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES --
C                STATIC(1) IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS;
C                STATIC(3) IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C  STHMA       THE STATIC AXIAL STRESS DUE TO THERMAL GRADIENT
C  STR1A       THE STATIC AXIAL STRESS DUE TO FORCE
C  STR1B       THE STATIC AXIAL STRESS DUE TO BENDING
C  STR1C       THE STATIC AXIAL STRESS DUE TO MOMENTUM CHANGE (FLUID)
C  STR2A       THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO INTERNAL PRESSURE
C  STR2B       THE STATIC HOOP STRESS AT OUTER SURFACE DUE TO EXTERNAL PRESSURE
C  STRAMP()    2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C                -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C                THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                STRAMP(4,*) ARE THE SHEAR STRESSES
C  T()         1-D ARRAY CONTAINING THE TIME-VARYING TORQUE LOADS
C  THIC        WALL THICKNESS AT DUCT OUTER RADIUS
C  TSTAT       STATIC TORQUE LOAD
C  V()         2-D ARRAY CONTAINING THE TIME-VARYING SHEAR LOADS -- V(1,*)
C                ARE THE V2 LOADS; V(2,*) ARE THE V3 LOADS
C  VSTAT()     1-D ARRAY CONTAINING THE STATIC SHEAR LOADS -- V(1) IS THE V2
C                LOAD; V(2) IS THE V3 LOAD
C  WOFF        WELD OFFSET

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

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

      DO 50 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)) * (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 RADII APPROPRIATE TO LOCATION
C     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
```

```
      RO2 = RO ** 2
      RDIFF = RO2 - RI2

      GEOM = 1.00 - 2.00 * LOG (RO / RI) * RO2 / RDIFF

C     TEMPERATURE STRESS

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

C     AXIAL STRESS CALCULATIONS

      STR1A = PSTAT / AREA
      STR1B = (MSTAT(1) * COS (ANGLE) + MSTAT(2) * SIN (ANGLE))
     &         * RI / MI
      STR1C = (PC - PCO) * RI2 / RDIFF

      STATIC(1) = (STR1A + STR1B + STR1C) * SKT1 * KOFF + STHMA

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

      STR2A = PC * (RI2 + RO2) / RDIFF
      STR2B = - 2.0 * PCO * RO2 / RDIFF

      STATIC(2) = (STR2A + STR2B) * SKT2 + STHMA

      STATIC(3) = - PC

C     SHEAR STRESS

      STATIC(4) = TSTAT * RI / (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,*) '    STR1A = ', STR1A, ' STR1B = ', STR1B
         WRITE(8,*) '    STR1C = ', STR1C, ' STHMA = ', STHMA
         WRITE(8,*) '    STATIC(1) = ', STATIC(1)
         WRITE(8,*) 'HOOP STRESSES'
         WRITE(8,*) '    STR2A = ', STR2A, ' STR2B = ', STR2B,
     &              ' STHMA = ', STHMA
         WRITE(8,*) '    STATIC(2) = ', STATIC(2)
         WRITE(8,*) 'RADIAL STRESS', '   STATIC(3) = ', STATIC(3)
         WRITE(8,*) 'SHEAR STRESS', '   STATIC(4) = ', STATIC(4)
         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))
     &              * RI / MI

         STRAMP(1,I) = (SIG1A(I) + SIG1B(I)) * SKT1 * KOFF

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) * RI / (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,*) '    STRAMP(1,I) = ', STRAMP(1,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


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

    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,
C             NLOAD, NUMREG, NUMSEG, SBND, PERIOD, SE, STATIC, STRAMP,
C             STRHIS, SZERO, TRUNC, ZROREG
C
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

C                   LIST OF VARIABLES
C
C  E()        1-D ARRAY CONTAINING THE STRAIN VALUES
C  EM         YOUNG'S MODULUS BEFORE YIELD
C  FATLIF     VALUE OF FATIGUE LIFE CALCULATED
C  FTY        YIELD STRENGTH
C  FTU        ULTIMATE STRENGTH
```

```
C  I            CONTROLS DO LOOP FOR RANDOM, SUPERIMPOSED SINUSOIDAL AND
C                 AERODYNAMIC LOADS
C  IOUT         OUTPUT DUMP CONTROLLER
C  J            CONTROLS DO LOOP FOR EACH POINT IN THE STRESS-TIME HISTORY
C  KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  KT           STRESS CONCENTRATION FACTOR
C  LNA()        1-D ARRAY CONTAINING VALUES OF Ln(A) = Ln(K**m) FOR EACH REGION
C  LPHIM()      1-D ARRAY CONTAINING VALUES OF Ln(PHI)m FOR EACH REGION
C  M            NUMBER OF POINTS IN STRESS-TIME HISTORY
C  MAXLD        MAXIMUM NUMBER OF TIME-VARYING LOADS ALLOWED
C  MAXM         MAXIMUM NUMBER OF POINTS ALLOWED IN STRESS-TIME HISTORY
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSEG       MAXIMUM NUMBER OF SEGMENTS ALLOWED
C  MM()         1-D ARRAY CONTAINING VALUES FOR m FOR EACH REGION
C  NEUB        NEUBER'S RULE MODEL ACCURACY FACTOR
C  NLOAD        NUMBER OF TIME-VARYING LOADS
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  NUMSEG       NUMBER OF SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
C  PERIOD       TIME IN SECONDS FOR ONE PERIOD OF STRESS-TIME HISTORY
C  RAINF2       FUNCTION WHICH CALCULATES THE TIME TO FAILURE FOR A GIVEN
C                 UNI-AXIAL STRESS-TIME HISTORY
C  S()          2-D ARRAY CONTAINING THE TOTAL COMPONENT STRESS-TIME HISTORIES
C  SBND()       1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C                 CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C                 REGION CORRECTED FOR PHI, KRATIO AND LNZ
C  SE()         1-D ARRAY CONTAINING THE STRESS-STRAIN PRODUCT
C  SEFF()       1-D ARRAY CONTAINING THE EFFECTIVE (OR UNI-AXIAL) STRESS-TIME
C                 HISTORY RESULTING FROM THE COMBINATION OF STATIC, RANDOM, AND
C                 SINUSOIDAL LOADS FOR ALL FOUR COMPONENTS
C  STATIC()     1-D ARRAY CONTAINING VALUES OF THE STATIC STRESSES -- STATIC(1)
C                 IS THE AXIAL STRESS; STATIC(2) IS THE HOOP STRESS; STATIC(3)
C                 IS THE RADIAL STRESS; STATIC(4) IS THE SHEAR STRESS
C  STRAMP()     2-D ARRAY CONTAINING VALUES OF THE TIME-VARYING STRESSES
C                 -- STRAMP(1,*) ARE THE AXIAL STRESSES; STRAMP(2,*) ARE
C                 THE HOOP STRESSES; STRAMP(3,*) ARE THE RADIAL STRESSES;
C                 STRAMP(4,*) ARE THE SHEAR STRESSES
C  SZERO        STRESS TENSILE POINT (PSI)
C  TRUNC        VALUE USED TO FILTER OUT NOISE IN THE STRESS-TIME HISTORY
C  ZROREG       ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUES -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                 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)

C           NOTE:  STRAMP(2,I) = STRAMP(3,I) = 0

  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
      ENDIF

      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,*) 'PERIOD = ', PERIOD, ' FATLIF = ', FATLIF

      RETURN
      END


C***********************************************************************
C
C  FUNCTION RAINF2 CALCULATES THE TIME (in seconds) TO FAILURE FOR
C  THE GIVEN UNI-AXIAL (OR EFFECTIVE) STRESS-TIME HISTORY
C
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20SEP89
C     VERSION:  4.2+ (CONSISTENT WITH THDUCT V3.8, V3.9, V4, & V4.1,
C                     MATCHR V8.4, V8.5)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
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, LPHIM, M, MM, NEUB, PERIOD,
C          SBND, NUMREG, NUMSEG, SE, SEFF, SZERO, TRUNC, ZROREG
C
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     ARGM, CHKFT, E(MAXSEG), EE(MAXM), EM, FTU, FTY,
     &         GTLIFE, INVLIF(MAXM), KRATIO, KT, LIFE(MAXM),
     &         LNA(0:MAXREG), LNZ, LPHIM(0:MAXREG), MM(0:MAXREG),
     &         NEUB, NEUBER, PERIOD, RAINF2, S(MAXM), SALTF(MAXM),
     &         SBND(0:MAXREG), SE(MAXSEG), SEFF(MAXM), SEFFM(2,MAXM),
     &         SEFMAX, SM, SMEANF(MAXM), SP(MAXM), STR(MAXM), SUMDAM,
     &         SZERO, TEST1(MAXM), TEST2(MAXM), TRUNC

C                      LIST OF VARIABLES
C
C  RAINF2       CYCLES TO FAILURE FOR THE GIVEN STRESS LEVELS
C
C input variables:
C
C  E()          1-D ARRAY CONTAINING THE STRAIN VALUES
C  EM()         YOUNG'S MODULUS BEFORE YIELD
C  FTU          ULTIMATE TENSILE STRENGTH (PSI)
C  FTY          YIELD TENSILE STRENGTH (PSI)
```

```
C  IOUT          OUTPUT DUMP CONTROLLER
C  KRATIO        RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  KT            FATIGUE CONCENTRATION FACTOR
C  LNA()         1-D ARRAY CONTAINING VALUES OF Ln(A) = Ln(K**M) FOR
C                  EACH REGION
C  LNZ           NORMAL (0,PVAR) GENERATED FROM RANDOM VARIATE
C  LPHIM()       1-D ARRAY CONTAINING VALUES OF Ln(PHI)*M FOR EACH
C                  REGION WHERE PHI IS A WEIBULL(BETAo, ETAo) GENERATED
C                  RANDOM VARIATE
C  M             TOTAL NUMBER OF STRESS DATA POINTS PER PERIOD
C  MM()          1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C  MAXM          MAXIMUM NUMBER OF POINTS IN STRESS TIME HISTORY
C  MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSEG        MAXIMUM NUMBER OF SEGMENTS ALLOWED
C  NUMREG        NUMBER OF REGIONS OF INTEREST
C  NUMSEG        NUMBER OF SEGMENTS OF INTEREST
C  PERIOD        TIME IN SECONDS FOR ONE PERIOD
C  SBND          1-D ARRAY CONTAINING LOWER BOUNDS FOR THE NUMREG STRESS
C                  REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE)
C                  MATERIALS S-N DATA SET
C  SE()          1-D ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
C  SEFF(M)       EFFECTIVE STRESSES BEFORE FILTERING/RAINFLOW
C  SZERO         STRESS TENSILE POINT (PSI)
C  TRUNC         VALUE USED TO FILTER OUT NOISE
C  ZROREG        ZeRO REGion -- VALUES CHOSEN TO FACILIATE REGION DO LOOPS
C                BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                REGION
C
C
C intermediate variables:
C
C  ARGM          INTERMEDIATE CALCULATION VARIABLE EQUAL TO KT/(1 - SM/FTU)
C  BIG1          VALUE OF i FOR SEFMAX
C  CHKFT         RAINFL VALUE OF LIFE
C  EE()          HOLDING ARRAY USED TO FIND CYCLES DURING RAINFLOW ANALYSIS
C  GETLIF        FUNCTION WHICH CALCULATES THE NUMBER OF CYCLES TO FAILURE
C  INDEX()       COUNTER FOR EFFECTIVE STRESSES
C  INVLIF(N)     INVLIF(I) = 1/LIFE(I); DAMAGE FRACTION
C  I,J,K         COUNTERS FOR VARIOUS DO LOOPS
C  JMAX          INDEX (LOCATION) OF SEFMAX IN SEFF()
C  LIFE(N)       LIFE(I) = CALCULATED LIFE FOR STRESS LEVEL STR(I)
C  N             NUMBER OF CYCLES FOUND DURING RAINFLOW ANALYSIS
C  NEUB          NEUBER'S RULE MODEL ACCURACY FACTOR
C  NEUBER        FUNCTION TO CALCULATE EQUIVALENT MEAN STRESS
C  NEWTOT        TOTAL NUMBER OF EFFECTIVE STRESS VALUES AFTER FILTERING
C  OVER          FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
C  RAINF2        FUNCTION WHICH CALCULATES TIME TO FAILURE FOR  A GIVEN
C                    UNI-AXIAL STRESS-TIME HISTORY
C  S(NEWTOT)     FILTERED EFFECTIVE STRESSES
C  SALTF(N)      SALTF(I) = sigma alternating,eff,i
C  SEFFM(2,N)    EFFECTIVE STRESSES AFTER RESEQUENCING/FILTERING/RAINFLOW
C                    SEFFM(1,I) = sigma max,eff,i
C                    SEFFM(2,I) = sigma min,eff,i
C  SEFMAX        LARGEST EFFECTIVE STRESS
C  SM            SM = EQUIVALENT MEAN STRESS
C  SMEANF(N)     SMEANF(I) = sigma mean,eff,i
C  SP(M+1)       RESEQUENCED EFFECTIVE STRESSES; # OF PTS = M+1
C  STR(N)        STR(I) = EQUIVALENT (COMBINED) STRESS,i
C  SUMDAM        SUM OF ALL THE DAMAGE FRACTIONS
C  TEST1()       1-D ARRAY USED IN FILTERING THE STRESSES
C  TEST2()       1-D ARRAY USED IN FILTERING THE STRESSES
C  TOHIGH        COUNTER FOR AMOUNT OF STRESSES IN EXCESS OF FTU

C  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,*)'lna(i),      lphim(i),      mm(i),      sbnd(i)'
         write(8,*) (lna(i), lphim(i), mm(i), sbnd(i), i=zroreg,numreg)
         WRITE(8,*) 'EM   :',EM,'     TRUNC    :',TRUNC,'   NEUB  :',NEUB
         WRITE(8,*) 'E():', E
```

```
         WRITE(8,*) 'SE():', SE
         WRITE(8,*) 'NUMSEG:', NUMSEG
         write(8,*)' '
      endif

C  INITIALIZE ARRAYS

      DO 50 I = 1, MAXM
         SP(I) = 0.0
         S(I) = 0.0
         EE(I) = 0.0
         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******************** B E G I N   R E S E Q U E N C E ********************
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
      SP(M+1) = SEFF(JMAX)
      if (iout.eq.20) then
         write(8,*)'sefmax:',sefmax,'    jmax:',jmax
         write(8,*)'sp(m+1):',(sp(i),i=1,m+1)
      endif

C********************* E N D   R E S Q U E N C E ************************

C********************* B E G I N   F I L T E R **************************
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
c       if (iout .eq. 20) then
c          do 305 i = 2, m
c             write(8,*) 'test1 = ', test1(i), ' test2 = ', test2(i)
c  305     continue
c       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,*)'newtot:',newtot
         write(8,*)'s(newtot):',(s(i),i=1,newtot)
      endif

C*************************** E N D    F I L T E R ****************************

C********************** B E G I N    R A I N F L O W ***********************
C  RAINFLOW ANALYSIS to identify cycles within effective stress data,
C  S(NEWTOT); places each cycle's max and min values into SEFFM(2,N)
C
C  counters: I counts # of cycles found, J counts how many S()'s counted,
C            K accumulates unmatched points

      I = 0
      J = 0
      K = 0
  400 CONTINUE

      J = J+1
      K = K+1

c   check J to avoid reading beyond end of filtered stress data
      IF ( J .GT. NEWTOT ) GOTO 499

c   read stress point into a holding array to be checked for cycles
      EE(K) = S(J)

  410 IF ( K .LT. 3 ) GOTO 400
      IF (ABS (EE(K) - EE(K-1)) .LT. ABS (EE(K-1) - EE(K-2))) GOTO 400

c    if not, then a cycle has been found, but we need to check for truncation

      IF (ABS (EE(K-1) - EE(K-2)) .GT. TRUNC) THEN

c        cycle is large enough to save

         I = I+1
         SEFFM(1,I) = AMAX1( EE(K-1), EE(K-2) )
         SEFFM(2,I) = AMIN1( EE(K-1), EE(K-2) )

      ENDIF
c   discard points K-1 and K-2, and decrement the counter of unmatched points

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

```
  499 CONTINUE
C    N equals the final number of cycles found
      N = I

      if (iout.eq.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 TOO LARGE -- NO CYCLES LEFT
         SUMDAM = 1.0E-36
         GOTO 710
      ENDIF

C
C*********************** E N D   R A I N F L O W ***************************

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) then
         write(8,*)'saltf(n) :',(saltf(i),i=1,n)
         write(8,*)'smeanf(n):',(smeanf(i),i=1,n)
      endif

C******* Determine Equivalent Mean Stress, SM(N), (two methods) ***********
C

      BIG1 = N
      OVER = 0

C  We are calculating the equivalent mean stress using neuber's rule
C  SM is the equivalent mean stress (see Rocketdyne IL2112-3139)

      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
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), MM, LNA, LPHIM, KRATIO, LNZ,
     &                     SBND, ZROREG, NUMREG, SZERO)

  600 CONTINUE

      DO 650 I=1,N
         INVLIF(I) = 1.0 / LIFE(I)

  650 CONTINUE

      if (iout.eq.20) then
         do 14 i=1,n
            write(8,*)'life(n):',life(i),'      invlif(n):',invlif(i)
   14    continue
      endif


C  Miner's Rule -- sum the damage fractions
C
      SUMDAM = 0.0
      DO 700 I=1,N
         SUMDAM = SUMDAM + INVLIF(I)

  700 CONTINUE
  710 CONTINUE

      if (iout.eq.20) write(8,*)'sumdam:',sumdam

C  calculate fatigue life (time to failure) in seconds
C
      RAINF2 = PERIOD / SUMDAM

      if (iout.eq.15) then
         chkft=period/sumdam
         write(8,*)' rainfl life',chkft
         write(8,*)
      endif

      RETURN
      END



C*************************************************************************
C  NEUBER USES NEUBERS RULE AND THE STRESS-STRAIN CURVE TO CALCULATE THE
C  THE MEAN STRESS.  PROGRAM ASSUMES THAT THE STRESS STRAIN CURVE IS
C  PIECEWISE LINEAR WITH AT MOST FIVE SECTIONS.
C
C  PROGRAMMER:  L. NEWLIN
C        DATE:  13SEP88
C     VERSION:  THDUCT V3.7, V3.8, V3.9, V4, V4.1
C*************************************************************************

      FUNCTION NEUBER (EM, SALT, SMEAN, NUMSEG, E, SE, NEUB, OVER)

C  INPUTS:   EM, SALT, SMEAN, NUMSEG, SE, E, NEUB, OVER
C  OUTPUTS:  NEUBER

C       IMPLICIT NONE

        COMMON IOUT

        INTEGER I, IOUT, MAXSEG, NUMSEG, OVER

        PARAMETER (MAXSEG = 10)

        REAL  E(MAXSEG), EM, EPSLON, NEUB, NEUBER, PRODCT, SALT,
     &        SE(MAXSEG), SMEAN, ST, TEMP

C
```

```
C                          LIST OF VARIABLES
C
C   E()            STRAIN VALUES FOR EACH SEGMENT
C   EM             YOUNG'S MODULUS BEFORE YIELD
C   EPSLON         CALCULATED STRAIN (WHERE PLASTIC=ELASTIC DEFORMATION)
C   I              CONTROLS DO LOOP FOR EACH SEGMENT
C   IOUT           OUTPUT DUMP CONTROLLER
C   MAXSEG         MAXIMUM NUMBER OF SEGMENTS ALLOWED (STRESS-STRAIN)
C   NEUB           NEUBER'S RULE MODEL ACCURACY FACTOR
C   NEUBER         TOTAL EQUIVALENT MEAN STRESS
C   NUMSEG         NUMBER OR SEGMENTS OF INTEREST IN STRESS-STRAIN CURVE
C   OVER           FLAG INDICATING THAT LIFE IS ONLY ONE CYCLE
C   PRODCT         STRESS STRAIN PRODUCT (WHERE PLASTIC=ELASTIC DEFORMAITON)
C   SALT           TOTAL ALTERNATING STRESS
C   SE()           1-DIMENSIONAL ARRAY CONTAINING THE STRESS-STRAIN PRODUCTS
C                  FOR A MULTI-SEGMENT CURVE
C   SMEAN          MEAN STRESS
C   ST             UNI-AXIAL TOTAL STRESS
C   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
  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,*)
         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.

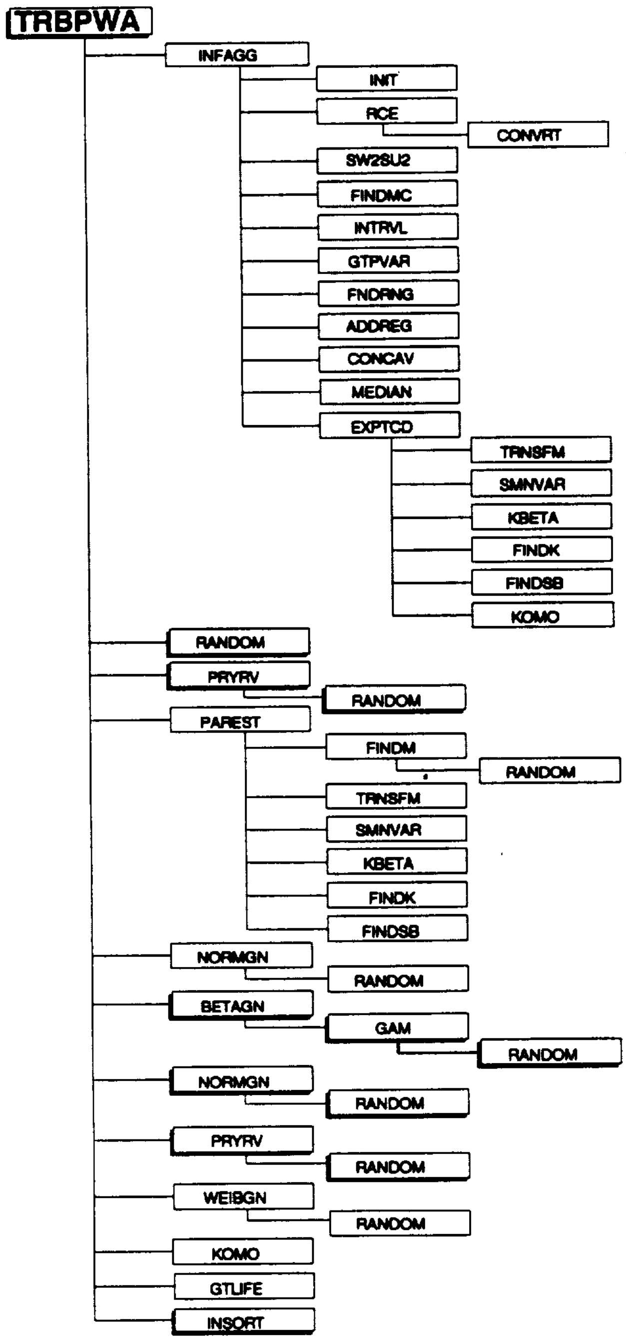

Figure 7-5 Tree Structure for Program TRBPWA for the Uniform Variation in Materials Shape Parameter *m*

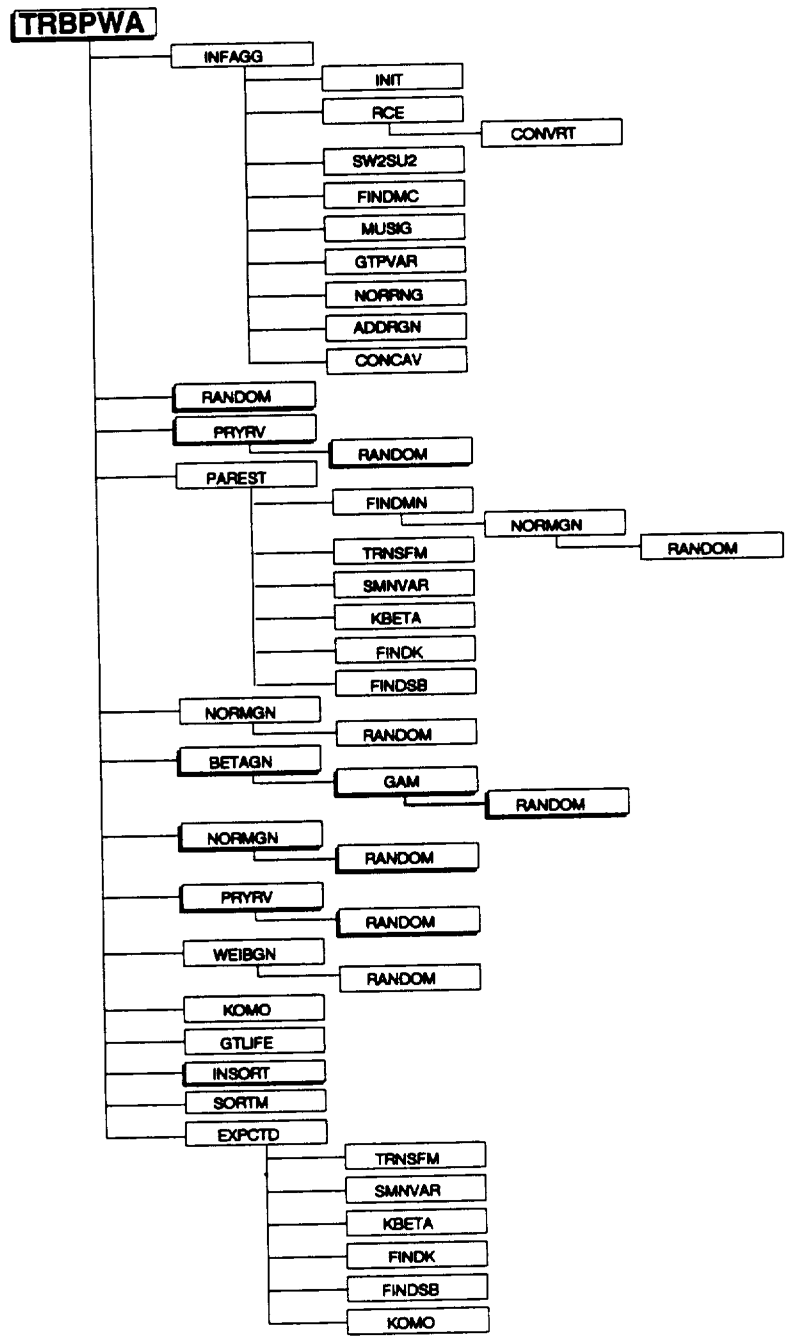

Figure 7-6 Tree Structure for Program TRBPWA for the Truncated Normal Variation in Materials Shape Parameter *m*

Table 7-5 List of Subprograms For Program TRBPWA
(Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
ADDREG[1]	4.1.3.9	Adds the *m* ranges for the non-data life regions to the right of those with data, for the Uniform distribution case.
ADDRGN[1]	4.1.3.15	Adds the *m* ranges for the non-data life regions to the right of those with data, for the truncated Normal distribution case.
BETAGN[2]	4.4.5	Generates Beta(a, b, ρ, θ) random variates.
CONCAV[3]	4.1.3.10	Adjusts the upper bound of the posterior ranges on *m* to be consistent with concavity constraints.
CONVRT[4]	4.1.3.3	Transforms stress data to equivalent zero-mean stresses with stress ratio of -1.0
EXPCTD[5]	4.1.3.12	Calculates the median S/N curve parameters from the results of the information aggregation calculations.
FINDK	4.1.5.6	Calculates the value of the location parameter *K* (where $A = K^m$) for each life region by using *Equations 2-37* and *2-41*.
FINDM[6]	4.1.5.1	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.
FINDMC	4.1.3.5	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*.
FINDMN[6]	4.1.5.2	Obtains the value of *m* for each life region by sampling from the appropriate truncated Normal distribution on *m*.
FINDSB	4.1.5.7	Calculates the life region "tie-points" or stress values which correspond to the "life boundaries" conditional on the randomly selected *m* for each region. Also calculates *K*, characterizing the specific material S/N data set, which is a function of β_o and *k*.
FNDRNG[7]	4.1.3.8	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.
GAM	4.4.4	Generates Gamma(α, 1) random variates.
GTLIFE	4.1.8	Calculates the cycles to failure for a particular stress based upon the materials characterization model S/N curve of *Equation 2-48.*
GTPVAR	4.1.3.7	Calculates σ^2, *Equation 2-49*, the extent of departures from the multiple heat median S/N curve warranted by the information available.
INFAGG[8]	4.1.3	Controls the logical flow for the information aggregation portion of the materials characterization model.
INIT	4.1.3.1	Initializes the entries of the arrays used in the information aggregation subroutine, INFAGG, to zero.

Table 7-5 List of Subprograms For Program TRBPWA (Cont'd)

NAME	SECTION	PURPOSE
INSORT	5.B	Performs an insertion sort for the lowest fifty percent of the lives calculated.
INTRVL	4.1.3.6	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*.
KBETA	4.1.5.5	Calculates *k* and β_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*.
KOMO[9]	4.1.6	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.
MEDIAN	4.1.3.11	Calculates the median values of *m* based on the posterior credibility ranges of *m* by using *Equation 2-34*.
MUSIG[10]	4.1.3.13	Calculates the posterior Normal distribution parameters: mean m_* and standard deviation σ_*, for each life region of the S/N curve.
NORMGN[11]	4.4.3	Generates Normal(μ, σ^2) random variates.
NORRNG[7]	4.1.3.14	Combines the implicit and explicit constraints on *m* to obtain the posterior credibility ranges of *m* for each life region.
PAREST[12]	4.1.5	Controls the logical flow for the parameter estimation model portion of the materials characterization model.
PRYRV[13]	7.6.6	Generates the Uniform(*a*, *b*) and Uniform(*c*, *d*) pair of independent random variates.
RANDOM[13]	4.4.2	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.
RCE	4.1.3.2	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.
SMNVAR	4.1.5.4	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*.
SORTM[14]	4.1.10	Sorts the *m* values in increasing order for each life region for the truncated Normal distribution case.
SW2SU2	4.1.3.4	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.

Table 7-5 List of Subprograms For Program TRBPWA (Cont'd)

NAME	SECTION	PURPOSE
TRBPWA	5.2.2	The main routine that controls the logical flow and performs the driver transformation calculations of the low cycle fatigue turbine disk program.
TRMNAT	4.1.11	Performs premature program termination, when required.
TRNSFM[15]	4.1.5.3	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.
WEIBGN	4.4.6	Generates Weibull(β, $\eta(\beta)$) random variates.

---

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)

VARIABLE NAME	TYPE	DESCRIPTION
ALLM(MAXMM, MAXREG)	RE	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]
BIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter *K*, *Equation 2-12*, where $A = K^m$.
BIGK1	RE	Dummy variable used during calls to subroutine EXPCTD, equal to **BIGK(1)**.
BLFPER(MAXBLF)	RE	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%.
BLFPOS(MAXBLF)	INT	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( )**.
BZERO	RE	Estimate of Weibull distribution shape parameter β_o, *Equation 2-11*, which characterizes the intrinsic variation of the S/N data set.
CG	RE	C_G (psi/°F) in *Equation 2-98*, the sensitivity of stress to the deviation from nominal thermal gradient ΔG_T, due to the deviation from nominal coolant fluid temperature ΔT_f.
CG1	RE	C_{G1} in *Equation 2-96*, the sensitivity of thermal gradient to the deviation from nominal coolant fluid temperature ΔT_f for $\Delta T_f < 0$.
CG2	RE	C_{G2} in *Equation 2-96*, the sensitivity of thermal gradient to the deviation from nominal coolant fluid temperature ΔT_f for $\Delta T_f \geq 0$.

Table 7-6 List of Variables For Program TRBPWA (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
CM	RE	C_m (psi/°F) in *Equation 2-97*, the sensitivity of stress to the deviation from nominal metal temperature ΔT_m, due to the deviation from nominal coolant fluid temperature ΔT_f.
CMF	RE	C_{mf} (psi/°F) in *Equation 2-95*, the sensitivity of metal temperature to the deviation from nominal coolant fluid temperature ΔT_f.
CS	RE	C_S in *Equation 2-94*, the rotational speed correction factor, (*selected speed / reference speed*)^2.
DELGT	RE	ΔG_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 ΔT_f.
DELTA	RE	ΔT_f lower bound of Beta distribution 1.
DELTB	RE	ΔT_f upper bound of Beta distribution 1.
DELTC	RE	ΔT_f lower bound of Beta distribution 2.
DELTD	RE	ΔT_f upper bound of Beta distribution 2.
DELTE	RE	Decimal equivalent percentage weight occuring in Beta distribution 1 of the deviation from nominal coolant fluid temperature ΔT_f.
DELTF	RE	ΔT_f (°F) in *Equation 2-95*, the randomly selected deviation from nominal coolant fluid temperature.
DELTHI	RE	Upper bound of the randomly selected Beta distribution for the deviation from nominal coolant fluid temperature ΔT_f.
DELTLO	RE	Lower bound of the randomly selected Beta distribution for the deviation from nominal coolant fluid temperature ΔT_f.
DELTM	RE	ΔT_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 ΔT_f.
DELTR	RE	Randomly selected Beta distribution location parameter ρ for the deviation from the nominal coolant fluid temperature ΔT_f.
DELTR1	RE	ρ Uniform distribution lower bound of Beta distribution 1 of ΔT_f.

Table 7-6 List of Variables For Program TRBPWA (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
DELTR2	RE	ρ Uniform distribution upper bound of Beta distribution 1 of ΔT_f.
DELTR3	RE	ρ Uniform distribution lower bound of Beta distribution 2 of ΔT_f.
DELTR4	RE	ρ Uniform distribution upper bound of Beta distribution 2 of ΔT_f.
DELTT	RE	Randomly selected Beta distribution shape parameter θ for the deviation from the nominal coolant fluid temperature ΔT_f.
DELTT1	RE	θ Uniform distribution lower bound of Beta distribution 1of ΔT_f.
DELTT2	RE	θ Uniform distribution upper bound of Beta distribution 1of ΔT_f.
DELTT3	RE	θ Uniform distribution lower bound of Beta distribution 2 of ΔT_f.
DELTT4	RE	θ Uniform distribution upper bound of Beta distribution 2 of ΔT_f.
FACTR	RE	Equal to FACTOR = **PHI * KRATIO * Z**. Used by the materials model.
FIFTY	RE	Variable used to access the fifty-percent point in the **LIFE ( )** array.
FTU	RE	Material ultimate strength (psi).
FTY	RE	Material yield strength (psi).
GTLIFE	RE	Function that calculates the fatigue cycles to failure at a given stress, given by *Equation 2-48.*
I	INT	Controls inner DO loop.
IOUT	INT	Output dump controller.
J	INT	Controls DO loop for each B-life.[2]
K	INT	Controls outer DO loop.
KD	RE	K_d in *Equation 2-102*, the stress factor to adjust for using 2-D stress analyses.
KRATIO	RE	Ratio of *MED K*/MED K* in *Equation 2-48.* **KRATIO** is constant over life regions for the materials model.

Table 7-6 List of Variables For Program TRBPWA (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
KT	RE	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.
L	INT	Controls DO loop for each life region of the S/N curve.
LAMG	RE	λ_G in *Equation 2-98*, the factor characterizing the sensitivity of stress due to thermal gradient.
LAMKD	RE	λ_{K_d} in *Equation 2-103*, the randomly selected model accuracy factor for the parameter K_d.
LAMKDA	RE	λ_{K_d} Uniform distribution lower bound.
LAMKDB	RE	λ_{K_d} Uniform distribution upper bound.
LAMKT	RE	λ_{K_t} in *Equation 2-103*, the randomly selected model accuracy factor for the parameter K_t.
LAMKTA	RE	λ_{K_t} Uniform distribution lower bound.
LAMKTB	RE	λ_{K_t} Uniform distribution upper bound.
LAMM	RE	λ_m in *Equation 2-97*, the factor characterizing the sensitivity of stress due to metal temperature.
LIFE(MAXLIF)	RE	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.
LNZ	RE	ln(Z) in *Equation 2-48*, the Normal(0, **PVAR**) random variate for the materials process variation aspect of the materials model.
M	INT	Controls symmetry DO loop.
MAXBLF	INT	Maximum number of B-lives to be obtained from the simulated failure distribution. The maximum number of B-lives allowed is 10.[2]
MAXDAT	INT	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.
MAXLIF	INT	Maximum number of fatigue lives allowed for the simulated failure distribution. The maximum number of fatigue lives to be saved is 10,000.
MAXMM	INT	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.

Table 7-6 List of Variables For Program TRBPWA (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
MAXREG	INT	Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.
MCOUNT	INT	Counts number of *m*'s to be used to calculate the median S/N curve for the truncated Normal distribution case.[1]
MEDM(MAXMM)	RE	1-D array containing the empirical median *m* for each life region of the S/N curve.[3]
MID	INT	Pointer to the median *m* values in array **SORTM( )** for the truncated Normal median S/N curve. Value of half of **MCOUNT**.
MINPHI	RE	Value of min(PHI), the minimum of **NSYM** draws of the materials scatter parameter φ.
MM(0:MAXREG)	RE	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.
MPROC	INT	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]
MU(MAXREG)	RE	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.
NBLIFE	INT	Number of B-lives to be obtained from the simulated failure distribution.[2]
NBND(0:MAXREG)	RE	$N^*_{i,i+1}$ 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.
NEWLIF	RE	Fatigue life value (cycles) returned from call to function GTLIFE.
NF(MAXDAT, MAXREG)	RE	2-D array containing values from the array **RAWNF( )** for the specific material S/N data set partitioned into life regions.
NHYPER	INT	The outer loop size.
NLIFE	INT	The inner loop size.
NLIFET	INT	Total number of lives calculated by program TRBPWA. Value of **NHYPER * NLIFE**.

Table 7-6 List of Variables For Program TRBPWA (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NMED	INT	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.
NPTS(MAXREG)	INT	1-D array containing the number of points per life region for the specific material S/N data set.
NSYM	INT	Symmetry number, usually equal to the multiplicity of the modeling unit in the component.
NUMREG	INT	R in *Equation 2-11*, the number of life regions of interest in the S/N curve.
PHI	RE	φ in *Equation 2-11*, the materials intrinsic variation, or scatter, given by a Weibull(β_o, $\eta_o(\beta_o)$) random variate.
PSIG	RE	σ in *Equation 2-48*, the value of SQRT(**PVAR**).
PVAR	RE	σ^2 in *Equation 2-48*, characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.
RAND	DRE	Random number seed.
RANGEM(2, MAXREG)	RE	2-D array containing values of the posterior credibility ranges on the materials model shape parameter m for each life region in the S/N curve. **RANGEM(1,L)** is the lower bound and **RANGEM(2,L)** is the upper bound in region L.[6]
REFSPD	RE	REFerence SPeeD ω_o (rpm) in *Equation 2-94*, the speed corresponding to the nominal mechanical stress **SMM** due to rotor speed.
SBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each life region of the S/N curve contained in array **NBND( )**.
SG	RE	S_{G_o} (psi) in *Equation 2-98*, the nominal stress due to the thermal gradient alone, at reference speed **REFSPD** and nominal coolant fluid temperature.
SIG(MAXREG)	RE	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.

Table 7-6 List of Variables For Program TRBPWA (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
SMM	RE	S_{M_o} (psi) in *Equation 2-94*, the nominal mechanical stress due to centrifugal effects only at the reference speed **REFSPD**.
SPDMU	RE	Mean μ of Normally distributed speed (rpm).
SPDSIG	RE	Standard deviation σ of Normally distributed speed (rpm).
SPEED	RE	ω (rpm) in *Equation 2-94*, the randomly selected rotational speed for rotational speed correction factor **CS**.
SR	RE	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.
STM	RE	S_{m_o} (psi) in *Equation 2-97*, the nominal stress due to the metal temperature alone, at the reference speed **REFSPD** and nominal coolant fluid temperature.
STR(MAXDAT, MAXREG)	RE	2-D array containing stress points with stress ratio = −1.0, for the specific material S/N data set partitioned into life regions.
SZERO	RE	Stress tensile test point, S_o (psi).[8]
TEST	RE	Uniform(0, 1) random variate used to determine Beta distribution for ΔT_f.
TRBIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter K consistent with the tensile point S_o.[8]
TRSBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = −1.0, corresponding to the "life boundary" values for each region of the S/N curve contained in array **NBND( )** for each **PHI** draw consistent with the tensile point S_o.[8]
VARY	INT	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.
Z	RE	Z in *Equation 2-48*, the randomly selected process variation shift factor given by a Lognormal(0, **PVAR**) random variate.

Table 7-6 List of Variables For Program TRBPWA (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
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 σ_* 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

TRBPWA Version 3

Program TRBPWA Listing Temporal Order, Uniform Distribution

Program TRBPWA Listing Temporal Order, Truncated Normal Distribution

Routine	Page

```
C*********************************************************************
C  PROGRAM TRBPWA CONTROLS THE FLOW OF LOGIC OF THE LOW CYCLE
C  FATIGUE ANALYSIS OF THE ATD-HPFTP 2ND STAGE TURBINE DISK
C  PROGRAMMER:  L. NEWLIN
C        DATE:  9SEP91
C     VERSION:  3  (MATCHR V8.5, INSORT V2.1)
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C*********************************************************************

      PROGRAM TRBPWA

C  SUBPROGRAMS:  INFAGG, PAREST, PRYRV, BETAGN, NORMGN, WEIBGN,
C                GTLIFE, INSORT, TRMNAT, SORTM, EXPCTD
C        FILES:  1:TRBPWD-OLD; 3:TRBPWO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
C                7:DUMP-NEW; 8:IOUTPR-NEW; 9:LOWLIF-NEW;
C                NOTE: 5 & 6 ARE OPENED IN 'INFAGG'

C     IMPLICIT NONE

      INTEGER  MAXBLF, MAXDAT, MAXLIF, MAXMM, MAXREG

      PARAMETER (MAXBLF= 10, MAXDAT = 50, MAXLIF = 10000,
     &           MAXMM = 20001, MAXREG = 3)

      COMMON   IOUT

      INTEGER  BLFPOS(MAXBLF), I, IOUT, J, K, L, M, MCOUNT, MID,
     &         MPROC, NBLIFE, NHYPER, NLIFE, NLIFET, NMED,
     &         NPTS(MAXREG), NSYM, NUMREG, VARY, ZROREG

      DOUBLE PRECISION  RAND

      REAL     ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1,
     &         BLFPER(MAXBLF), BZERO, CG, CG1, CG2, CM, CMF, CS,
     &         DELGT, DELTA, DELTB, DELTC, DELTD, DELTE, DELTF, DELTHI,
     &         DELTLO, DELTM, DELTR, DELTR1, DELTR2, DELTR3, DELTR4,
     &         DELTT, DELTT1, DELTT2, DELTT3, DELTT4, FACTR, FIFTY,
     &         FTU, FTY, GTLIFE, KD, KRATIO, KT, LAMG, LAMKD, LAMKDA,
     &         LAMKDB, LAMKT, LAMKTA, LAMKTB, LAMM, LIFE(MAXLIF), LNZ,
     &         MEDM(MAXMM), MINPHI, MM(0:MAXREG), MU(MAXREG),
     &         NBND(0:MAXREG), NEWLIF, NF(MAXDAT, MAXREG), PHI, PSIG,
     &         PVAR, RANGEM(2, MAXREG), REFSPD, SBND(0:MAXREG), SG,
     &         SIG(MAXREG), SMM, SPDMU, SPDSIG, 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,*) NLIFE
      WRITE(8,*) '                          INNER LOOP SIZE = ', NLIFE
      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
```

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

C ** READ DATA FROM TRBPWD

      READ(1,*) DELTA, DELTB, DELTR1, DELTR2, DELTT1, DELTT2,
     &          DELTC, DELTD, DELTR3, DELTR4, DELTT3, DELTT4,
     &          DELTE
      READ(1,*) SPDMU, SPDSIG
      READ(1,*) LAMKDA, LAMKDB
      READ(1,*) LAMKTA, LAMKTB
      READ(1,*) KD, KT, SMM, REFSPD
      READ(1,*) STM, CMF, CM
      READ(1,*) SG, CG1, CG2, CG

C ** ECHO DATA TO TRBPWO

      WRITE(3,900)
      WRITE(3,901) DELTA, DELTB, DELTR1, DELTR2, DELTT1, DELTT2,
     &             DELTC, DELTD, DELTR3, DELTR4, DELTT3, DELTT4,
     &             DELTE
      WRITE(3,902) SPDMU, SPDSIG
      WRITE(3,903) LAMKDA, LAMKDB
      WRITE(3,904) LAMKTA, LAMKTB
      WRITE(3,905) KD, KT, SMM, REFSPD
      WRITE(3,906) STM, CMF, CM
      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, ZROREG, NUMREG,
     &             NBND, STR, FTU, FTY, VARY, MPROC, KRATIO, PVAR)

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

      MCOUNT = 0

C ** INITIALIZE VARIABLES

      DO 10 K = 1, MAXLIF
         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 PRYRV (RAND,DELTR1,DELTR2,DELTT1,DELTT2,DELTR,DELTT)
            DELTLO = DELTA
            DELTHI = DELTB
         ELSE
            CALL PRYRV (RAND,DELTR3,DELTR4,DELTT3,DELTT4,DELTR,DELTT)
            DELTLO = DELTC
            DELTHI = DELTD
         ENDIF
         IF (IOUT .EQ. 15) THEN
            WRITE(8,*) 'TEST =', TEST, ' DELTE =', DELTE
            WRITE(8,*) 'DELTLO =', DELTLO, ' DELTHI =', DELTHI
         ENDIF

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

         CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG,
     &                RAND, NBND, STR, BIGK, BZERO, MM, SBND)

C ** OBTAIN MATERIALS PROCESS VARIATION IF DESIRED

         CALL NORMGN (RAND, 0.0, PSIG, LNZ)

         IF (MPROC .EQ. 1) THEN
            Z = EXP (LNZ)
         ELSE
            KRATIO = 1.0
            Z = 1.0
            LNZ = 0.0
         ENDIF

         MCOUNT = MCOUNT + 1
         DO 175 L = 1, NUMREG
            ALLM(MCOUNT,L) = MM(L)
  175    CONTINUE

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

         DO 200 I = 1, NLIFE

C ** SELECT DRIVERS FOR CALCULATING LIFE

            CALL BETAGN (RAND, DELTR, DELTT, DELTLO, DELTHI, DELTF)
            CALL NORMGN (RAND, SPDMU, SPDSIG, SPEED)
            CALL PRYRV (RAND, LAMKDA, LAMKDB,LAMKTA,LAMKTB,LAMKD,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(8,*) 'DELTF =', DELTF, ' SPEED =', SPEED
               WRITE(8,*) 'LAMKD =', LAMKD, ' LAMKT =', LAMKT
               WRITE(8,*) '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 = ',TRBIGK(L)
 250         CONTINUE
          ENDIF

C ** PERFORM DRIVER TRANSFORMATION

          CS = (SPEED / REFSPD) ** 2

          DELTM = 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,*) 'DELTM = ', DELTM, ' LAMM = ', LAMM
             WRITE(8,*) 'DELGT = ', DELGT, ' LAMG = ', LAMG
          ENDIF

          SR = LAMKD * LAMKT * (CS * SMM + LAMM * 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,
     &                                 ' NEWLIF = ', NEWLIF

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

C ** INITIALIZE VARIABLES

        DO 325 J = 1, MAXBLF
           BLFPOS(J) = 0
 325    CONTINUE

C ** PRINT EMPIRICAL BLIVES

        FIFTY = 0.50E0

        WRITE(3,910)
        DO 350 J = 1, NBLIFE
           BLFPOS(J) = NINT (BLFPER(J) * FLOAT (NLIFET))
           WRITE(3,920) BLFPER(J), LIFE(BLFPOS(J))
 350    CONTINUE
        WRITE(3,920) FIFTY, LIFE(NLIFET/2)
```

```
      ENDIF
C ** CALCULATE NORMAL MEDIAN CURVE IF DESIRED
      IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) THEN

         CALL SORTM (ALLM, NUMREG, MCOUNT)
         MID = MCOUNT / 2

         DO 400 L = 1, NUMREG
            MEDM(L) = ALLM(MID,L)
  400    CONTINUE

         CALL EXPCTD (1, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG,
     &                NBND, BIGK1, BZERO)

      ENDIF

C ** FORMAT STATEMENTS TO ECHO INPUT DATA TO TRBPWO

  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &       'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &       'NASA Contract NAS7-918 is acknowledged.',
     &       ////,33X,'INPUT DATA',
     &       ///,14X,'DRIVERS',25X,'PARAMETER DISTRIBUTIONS',
     &       //,48X,'RHO',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 'TRBPWD' INPUT FILE
C***********************************************************************
C 675                                       !RANDOM NUMBER SEED
C 0                                         !OUTPUT DUMP CONTROLLER
C 100                                       !INNER LOOP SIZE
C 200                                       !OUTER LOOP SIZE
```

```
C 50                                        !SYMMETRY NUMBER
C 2                                         !UNIFORM S/N VARIATION
C 0                                         !NORMAL MEDIAN NOT REQUIRED
C 0                                         !MATERIALS PROC. VAR. NOT REQUIRED
C 3                                         !NUMBER OF BLIVES TO BE REQUIRED
C 0.0001                                    !B.01 LIFE
C 0.001                                     !B.1 LIFE
C 0.01                                      !B1 LIFE
C -200. 200.  0.40   0.60  5.0  10.!DELTA Tf      (A,B) (R1,R2) (T1,T2)
C 200.  500.  0.50   0.50  0.1  0.5!              (C,D) (R3,R4) (T3,T4)
C 0.95                                      !TEST FOR DELTA Tf HYPER DIST.
C 37592.    507.                            !ROTATIONAL SPEED:  MU, SIGMA
C  0.80000   1.20000                        !LAMBDA Kd          (A,B)
C  0.95000   1.05000                        !LAMBDA Kt          (A,B)
C 1.41   2.18   159807.   38600.            !Kd, Kt, MECH. STRESS (PSI), SPEED (RPM)
C   1915.  0.91325   4.4435                 !THERMAL STRESS (PSI), Cmf, Cm
C  14749.  0.04   0.70   101.72             !THERM GRAD STRESS (PSI), Cg1, Cg2, Cg
C 'PWA HPFTP 2ND TURBINE DISK'              !SPECIFIC MATERIAL DESCRIPTION
C 00000.  198000.   1    9                  !YIELD & ULTIMATE STRENGTHS, NDIV, NPTS
C  9   -1.0   1                             !# PTS IN DIV, STRESS RATIO, REGION
C 160000.      600.                         !S(1) N(1)          RAW
C 160000.      700.                         !S(2) N(2)         STRESS
C 160000.     1000.                         !S(3) N(3)          LIFE
C 140000.     4800.                         !S(4) N(4)          (S/N)
C 130000.     3700.                         !S(5) N(5)          DATA
C 130000.     4300.                         !S(6) N(6)         POINTS
C 120000.     3800.                         !S(7) N(7)         FOR THE
C 120000.    11000.                         !S(8) N(8)         SPECIFIC
C 110000.    40000.                         !S(9) N(9)         MATERIAL
C 198000.                                   !TENSILE TEST POINT, So
C 1    0                                    !NUMBER OF ERGION:  W/DATA, W/O DATA
C 500.                                      !LIFE BOUNDARY REGION 0
C 1.0E+36                                   !LIFE BOUNDARY REGION 1
C 0.00                                      !CONSTRAINT ON COEFF. OF VARIATION
C 0      0.000       0.000                  !0 PTS IN RANGE, LOWER BOUND, UPPER BOUND
C 0.0     0.0        0.0                    !NORMAL DIST. PRIORS: DELTA, Mo, SIGMA2

C
C**********************************************************************
C                           LIST OF VARIABLES
C**********************************************************************
C
C ALLM()         2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C BIGK()         1-D ARRAY CONTAINING VALUES OF K WHERE A = K ** M
C BIGK1          DUMMY VARIABLE USED DURING CALLS TO EXPCTD
C BLFPER()       1-D ARRAY CONTAINING USER SPECIFIED BLIVES TO BE PROVIDED
C BLFPOS()       1-D ARRAY CONTAINING POSITION IN LIFE() OF EMPIRICAL BLIVES
C BZERO          VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S-N DATA SET
C CG             SENSITIVITY OF STRESS DUE TO VARIATION OF THERMAL GRADIENT
C                  DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE
C                  (PSI/deg F)
C CG1            SENSITIVITY OF THERMAL GRADIENT DUE TO DEVIATION FROM NOMINAL
C                  COOLANT FLUID TEMPERATURE FOR DELTA Tf < 0
C CG2            SENSITIVITY OF THERMAL GRADIENT DUE TO DEVIATION FROM NOMINAL
C                  COOLANT FLUID TEMPERATURE FOR DELTA Tf >= 0
C CM             SENSITIVITY OF STRESS DUE TO VARIATION OF METAL TEMPERATURE
C                  DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE
C                  (PSI/deg F)
C CMF            SENSITIVITY OF STRESS DUE TO VARIATION OF METAL TEMPERATURE
C                  DUE TO DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE
C                  (PSI / deg F)
C CS             ROTATIONAL SPEED CORRECTION FACTOR -- EQUAL TO (SELECTED SPEED
C                  / REFERENCE SPEED) ** 2
C DELGT          DELTA THERMAL GRADIENT (deg F) IS THE DEVIATION FROM NOMINAL
C                  THERMAL GRADIENT IN THE BLADE ATTACHMENT AREA AS A RESULT
C                  OF THE SELECTED DELTA Tf
C DELTA          DELTA Tf LOWER BOUND - HYPER-DISTRIBUTION 1
C DELTB          DELTA Tf UPPER BOUND - HYPER-DISTRIBUTION 1
C DELTC          DELTA Tf LOWER BOUND - HYPER-DISTRIBUTION 2
C DELTD          DELTA Tf UPPER BOUND - HYPER-DISTRIBUTION 2
C DELTE          PERCENTAGE OCCURRING IN HYPER-DISTRIBUTION 1
C                  -- DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE,
C                  DELTA Tf
C DELTF          SELECTED DEVIATION FROM NOMINAL COOLANT FLUID TEMPERATURE,
C                  DELTA Tf
```

```
C DELTHI         SELECTED LAMBDA UPPER BOUND FOR DEVIATION FROM NOMINAL
C                  COOLANT FLUID TEMPERATURE, DELTA Tf
C DELTLO         SELECTED LAMBDA LOWER BOUND FOR DEVIATION FROM NOMINAL
C                  COOLANT FLUID TEMPERATURE, DELTA Tf
C DELTM          DELTA Tm IS THE DEVIATION FROM THE NOMINAL METAL
C                  TEMPERATURE IN THE BLADE ATTACHMENT AREA AS A RESULT
C                  OF THE SELECTED DELTF
C DELTR          SELECTED RHO FOR DEVIATION FROM NOMINAL COOLANT FLUID
C                  TEMPERATURE, DELTA Tf
C DELTR1         DELTA Tf - RHO LOWER BOUND - HYPER-DISTRIBUTION 1
C DELTR2         DELTA Tf - RHO UPPER BOUND - HYPER-DISTRIBUTION 1
C DELTR3         DELTA Tf - RHO LOWER BOUND - HYPER-DISTRIBUTION 2
C DELTR4         DELTA Tf - RHO UPPER BOUND - HYPER-DISTRIBUTION 2
C DELTT          SELECTED THETA FOR DEVIATION FROM NOMINAL COOLANT FLUID
C                  TEMPERATURE, DELTA Tf
C DELTT1         DELTA Tf - THETA LOWER BOUND - HYPER-DISTRIBUTION 1
C DELTT2         DELTA Tf - THETA UPPER BOUND - HYPER-DISTRIBUTION 1
C DELTT3         DELTA Tf - THETA LOWER BOUND - HYPER-DISTRIBUTION 2
C DELTT4         DELTA Tf - THETA UPPER BOUND - HYPER-DISTRIBUTION 2
C FACTR          EQUAL TO FACTOR = PHI * KRATIO * Z
C FIFTY          EQUAL TO .5 -- USED TO ACCESS 50% POINT IN LIFE()
C FTU            MATERIAL ULTIMATE STRENGTH (PSI)
C FTY            MATERIAL YIELD STRENGTH (PSI)
C GTLIFE         FUNCTION WHICH CALCULATES THE CYCLES TO FAILURE AT A GIVEN
C                  STRESS
C I              CONTROLS DO LOOP FOR EACH LIFE CALCULATION
C IOUT           OUTPUT DUMP CONTROLLER
C J              CONTROLS DO LOOP FOR EACH BLIFE
C K              CONTROLS DO LOOP FOR EACH HYPER-PARAMETER SET
C KD             STRESS FACTOR TO ADJUST FOR USING 2-D ANALYSIS
C KRATIO         RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C KT             STRESS CONCENTRATION FACTOR USED IN ANALYSIS TO OBTAIN
C                  REFERENCE STRESSES
C L              CONTROLS DO LOOP FOR EACH REGION
C LAMG           THERMAL GRADIENT SENSITIVITY FACTOR
C LAMKD          SELECTED STRESS CONCENTRATION FACTOR Kd ACCURACY
C LAMKDA         Kd ACCURACY FACTOR LOWER BOUND
C LAMKDB         Kd ACCURACY FACTOR UPPER BOUND
C LAMKT          SELECTED STRESS CONCENTRATION FACTOR Kt ACCURACY
C LAMKTA         Kt ACCURACY FACTOR LOWER BOUND
C LAMKTB         Kt ACCURACY FACTOR UPPER BOUND
C LAMM           METAL TEMPERATURE SENSITIVITY FACTOR
C LIFE()         1-D ARRAY CONTAINING VALUES OF THE LIVES GENERATED BY THE PFM
C                  -- SORTED VALUES OF THE LEFT-HAND TAIL
C LNZ            NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C M              CONTROLS SYMMETRY DO LOOP
C MAXBLF         MAXIMUM NUMBER OF BLIVES TO BE CALCULATED
C MAXDAT         MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED
C MAXLIF         MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA,
C                  ALPHA CALCULATION
C MAXMM          MAXIMUM NUMBER OF M'S TO BE SORTED
C MAXREG         MAXIMUM NUMBER OF REGION ALLOWED
C MCOUNT         NUMBER OF M'S TO BE USED TO CALCULATE NORMAL MEDIAN S/N CURVE
C MEDM()         1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C MID            POINTER TO THE MEDIAN M VALUES -- EQUAL TO HALF OF MCOUNT
C MINPHI         EQUAL TO MIN(PHI) -- THE MINIMUM OF NSYM DRAWS OF PHI
C MM()           1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C MPROC          Materials PROCess variation -- CONTROLS MATERIALS PROCESS
C                  VARIATION -- 0 - NO VARIATION, 1 - VARIATION
C MU()           1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION MEAN
C                  FOR EACH REGION
C NBLIFE         NUMBER OF BLIVES TO BE CALCULATED
C NBND()         1-D ARRAY CONTAINING UPPER BOUNDS FOR THE NUMREG LIFE
C                  REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE)
C                  MATERIAL S/N DATA SET
C NEWLIF         LIFE VALUE RETURNED FROM CALL TO GTLIFE
C NF()           2-D ARRAY CONTAINING RAWNF() FOR THE SPECIFIC MATERIAL
C                  S/N DATA SET BROKEN INTO LIFE REGIONS
C NHYPER         SIZE OF OUTER LOOP
C NLIFE          SIZE OF INNER LOOP
C NLIFET         TOTAL NUMBER OF LIVES CALCULATED BY PFM
C NMED           CONTROLS MEDIAN CALCULATION FOR THE NORMAL DISTRIBUTION CASE
C                  -- 0 - NO MEDIAN CALCULATION;  1 - MEDIAN CALCULATION DESIRED
C NPTS()         1-D ARRAY CONTAINING THE NUMBER OF POINTS PER LIFE REGION
C                  FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
```

```
C NSYM          SYMMETRY NUMBER
C NUMREG        NUMBER OF REGIONS OF INTEREST
C PHI           WEIBULL (BETAo, ETAo) GENERATED RANDOM VARIATE
C PSIG          EQUAL TO SQRT(PVAR) -- MATERIALS PROCESS STANDARD DEVIATION
C PVAR          MATERIALS PROCESS VARIATION
C RAND          RANDOM NUMBER SEED
C RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND
C                 AND RANGEM(2,L) IS THE UPPER BOUND
C REFSPD        REFERENCE SPEED (RPM) CORRESPONDING TO THE NOMINAL MECHANICAL
C                 STRESS, SMM, DUE TO ROTATIONAL SPEED ONLY
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()
C SG            NOMINAL STRESS DUE TO THERMAL GRADIENT ONLY (PSI), AT
C                 REFERENCE SPEED, REFSPD, AND NOMINAL COOLANT FLUID
C                 TEMPERATURE
C SIG()         1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION
C                 STANDARD DEVIATION FOR EACH REGION
C SMM           NOMINAL MECHANICAL STRESS DUE TO ROTATION ONLY (PSI) AT
C                 REFERENCE SPEED, REFSPD
C SPDMU         MEAN SPEED, RPM (NORMAL DISTRIBUTION)
C SPDSIG        STANDARD DEVIATION OF SPEED, RPM (SIGMA, NORMAL DISTRIBUTION)
C SPEED         SELECTED ROTATIONAL SPEED, RPM, FOR ROTATIONAL SPEED
C                 CORRECTION FACTOR
C SR            VALUE OF MAXIMUM REFERENCE STRESS USED TO CALCULATE LIFE --
C                 STRESS RATIO IS ASSUMED TO BE GREATER THAN OR EQUAL TO 0
C STM           NOMINAL STRESS DUE TO METAL TEMPERATURE ONLY (PSI), AT
C                 REFERENCE SPEED, REFSPD, AND NOMINAL COOLANT FLUID
C                 TEMPERATURE
C STR()         2-D ARRAY CONTAINING STRESS POINTS (STRESS RATIO = -1.0) FOR
C                 THE SPECIFIC MATERIAL S/N DATA SET BROKEN INTO LIFE REGIONS
C SZERO         STRESS TENSILE TEST POINT, So
C TEST          UNIFORM(0,1) RANDOM VARIATE USED TO DETERMINE
C                 HYPER-DISTRIBUTION TO SELECT FROM
C TRBIGK()      1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO BRING CURVE
C                 DOWN IF SBND(0) > So FOR EACH TRIAL
C TRSBND()      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() FOR EACH TRIAL
C VARY          CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                 VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                 VARIATION;  3 - TRUNCATED NORMAL VARIATION
C Z             LOG-NORMAL(0,PVAR) RANDOM VARIATE
C ZROREG        ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                 REGION


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


C************************************************************************
C    SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C    INDEPENDENT RANDOM VARIATES
C    PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C          DATE:  9MAR87
C    SUBPROGRAM:  RANDOM
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.
C************************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, RHO1, RHO2, THE1, THE2, X, Y
```

```
      INTEGER IOUT

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

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

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &       ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      RETURN
      END

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


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

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

C  TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

      X = W * (B - A) + A
      IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X

      RETURN
      END
C***********************************************************************

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     pp. 407-409.

      REAL FUNCTION GAM (ALPHA, RAND)

C     SUBPROGRAM:  RANDOM

      COMMON  IOUT

      INTEGER IOUT
```

```
      REAL     A, ALPHA, ARG, U1, U2, V1, V2

      DOUBLE PRECISION RAND

      A = ALPHA - 1.
C     IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA

   10 CALL RANDOM (U1, RAND)
      CALL RANDOM (U2, RAND)
      V1 = - ALOG(U1)
      V2 = - ALOG(U2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =',
C    &                           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 INSORT PERFORMS AN INSERTION SORT FOR EACH LIFE CALCULATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  20JUN90
C     VERSION:  2.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INSORT (NEWLIF, LIFE, NLIFET)

C   INPUTS:  NEWLIF, LIFE, NLIFET
C  OUTPUTS:  LIFE

C     IMPLICIT NONE

      INTEGER  MAXLIF

      PARAMETER (MAXLIF = 10000)

      COMMON   IOUT

      INTEGER  I, IOUT, NLIFET, NUM, PLACE

      REAL     LIFE(MAXLIF), NEWLIF, TEMP(MAXLIF)


C                    LIST OF VARIABLES
C
C  I         CONTROLS DO LOOP FOR INSERTION
C  IOUT      OUTPUT DUMP CONTROLLER
C  LIFE()    1-D ARRAY CONTAINING TAIL VALUES OF THE LIVES GENERATED BY THE
C              PFM TO BE SORTED
C  MAXLIF    MAXIMUM NUMBER OF FATIGUE LIVES ALLOWED FOR BETA, THETA, ALPHA,
C              CALCULATION
C  NEWLIF    LIFE VALUE TO BE INSERTED INTO LIFE()
C  NLIFET    TOTAL NUMBER OF LIVES CALCULATED BY PFM
C  NUM       NUMBER OF LIFE VALUES IN LIFE()
C  PLACE     POSITION WHERE NEWLIF IS TO BE INSERTED INTO LIFE()
C  TEMP()    1-D ARRAY CONTAINING VALUES OF LIFE() TO BE SHIFTED UPON
C              INSERTION OF NEWLIF


      NUM = NLIFET / 2
```

```
C     FIND POSITION IN LIFE() FOR NEWLIF

      IF (NEWLIF .GT. LIFE(NUM)) GOTO 400

      DO 100 I = 1, NUM
         IF (NEWLIF .LT. LIFE(I)) THEN
            PLACE = I
            GOTO 110
         ENDIF
  100 CONTINUE
  110 CONTINUE

C     STORE VALUES OF LIFE() TO BE SHIFTED DUE TO NEWLIF INSERTION IN TEMP()

      DO 200 I = (PLACE + 1), NUM
         TEMP(I) = LIFE(I-1)
  200 CONTINUE

C     INSERT NEWLIF

      LIFE(PLACE) = NEWLIF

C     SHIFT VALUES OF LIFE() FOLLOWING NEWLIF

      DO 300 I = (PLACE + 1), NUM
         LIFE(I) = TEMP(I)
  300 CONTINUE

C     IF NEWLIF IS LARGER THAN ALL LIVES IN LIFE() THEN RETURN

  400 CONTINUE


      RETURN
      END

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

C  SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C  AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C  FOR THE STRESS FORMULATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  13JUL89      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.4, V8.5    MATGRM V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, 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,
C             NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C  SUBPROGRAMS:  INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDREG,
C                CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR
C        FILES:  5:RELATD-OLD; 6:RELATO-NEW

C      IMPLICIT NONE

       INTEGER MAXDAT, MAXREG, MAXSET

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

       COMMON  IOUT

       INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
     &         NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
```

```
     &          NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG
      REAL      BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
     &          FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
     &          KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &          LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
     &          MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG),
     &          MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &          PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET),
     &          RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &          SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT, MAXREG),
     &          SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(MAXREG),
     &          SXY(MAXREG), SY2(MAXREG), SZERO


C                    LIST OF VARIABLES
C
C   BIGKHT      EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
C   BZERO       VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N
C                 DATA SET
C   CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                 COEFFICIENT OF VARIATION, Co
C   DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C   DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
C                 AND SIG() CALCULATION
C   FTUZ        ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   FTYZ        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   IOUT        OUTPUT DUMP CONTROLLER
C   IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                 FOR EACH REGION
C   JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C   KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C   L           CONTROLS DO LOOP FOR EACH REGION
C   LAMN        LAMBDA-N -- RATIO OF Var(Ln N given S) / (m**2 C**2),
C                 CONSTANT OVER REGIONS AND COMPONENTS
C   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 IN 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   MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                 REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                 -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                 BOUND
C   MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                 MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   MCPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C   MEDM()      1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C   MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C   MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C   MPROC       Materials PROCess variation --CONTROLS MATERIALS PROCESS
C                 VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C   MU()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION MEAN FOR EACH REGION
C   MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C   NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C   NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                 SET IN EACH REGION
C   NPPR()      1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                 ALL DATA SETS IN A REGION (Number of Points Per Region)
C   NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
```

```
C  NUMREG        NUMBER OF REGIONS OF INTEREST
C  PVAR          MATERIALS PROCESS VARIATION
C  RANGEM()      2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                  FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                  RANGEM(2,L) IS THE UPPER BOUND
C  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) OR 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  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  STR()         2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                  S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SUHAT2()      1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                  REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT2()      1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
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  SXY()         1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                  REGION (X = Ln S, Y = Ln N)
C  SY2()         1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                  (Y = Ln N)
C  SZERO         STRESS TENSILE TEST POINT, So
C  VARY          CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                  VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                  VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  ZROREG        ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                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
C  INFORMATION

C  PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION

C  INITIALIZE PRIMARY ARRAYS

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

C  READ, CONVERT, ECHO INFORMATION

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

C  CALCULATE RESIDUAL VARIANCES

      CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, 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, MCHAT)
C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
       CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C  OBTAIN POSTERIOR RANGES ON M

      CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
     &             RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

      CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

      CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

      WRITE(7,900)

      DO 25 L = 1, NUMREG
         WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
     &                   JZERO(1, L), JZERO(2, L)
 25   CONTINUE

      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,
     &             NBND, BIGKHT, BZERO)

C  CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED

      IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
         DO 200 L = 1, NUMREG
            RANGEM(1,L) = MEDM(L)
            RANGEM(2,L) = MEDM(L)
 200     CONTINUE
```

```
         ENDIF
      ELSE
C  NORMAL VARIATION IS DESIRED
C  CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION
         CALL MUSIG (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTA, MO,
     &               SIGMA2, MCHAT, MU, SIG)
C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED
      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF
C  COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M
         CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)
C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA
         CALL ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO,
     &                MPNT, MO, SIGMA2)
C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS
         CALL CONCAV (NUMREG, RANGEM)
C  WRITE RESULTS TO FILE DUMP
         WRITE(7,975)
         DO 350 L = 1, NUMREG
            WRITE(7,980) L, MCHAT(1,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(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)
         DO 380 L = 1, NUMREG
            WRITE(7,990) L, MU(L), SIG(L)
  380 CONTINUE
      ENDIF
C  PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS
      IF (MPROC .EQ. 1) THEN
         WRITE(7,995) PVAR
      ENDIF
C  FORMAT STATEMENTS
```

```
900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &       'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &       'NASA Contract NAS7-918 is acknowledged.',////,
     &       2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
     &       ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
     &       'FOR EACH REGION',/)

905 FORMAT(7X,'REGION:  ',I1,7X,'Io = (',F12.9,',',F12.9,')',
     &       /,24X,'Jo = (',F12.9,',',F12.9,')')

910 FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
     &       //,7X,'REGION',8X,'E(C)',12X,'E(m)',/)

915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)

920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH '
     &       'REGION')

930 FORMAT(//,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

950 FORMAT(///)

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

965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')

970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

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

980 FORMAT(9X,I1,11X,F10.6)

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

990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)

995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',
     &       'MEDIAN S/N CURVE',/,2X,'WARRANTED BY THE AVAILABLE ',
     &       'INFORMATION',//,7X,E11.5)

      RETURN
      END

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


C   SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C   ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C   PROGRAMMER:  L. NEWLIN
C         DATE:  5OCT87
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRMNAT

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

```
C***************************************************************************

C   SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C   USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

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

C   INPUTS:    ---
C   OUTPUTS:   NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
C              NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

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

      COMMON  IOUT

      INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
     &        NPTS(0:MAXSET), REFNP(MAXREG)

      REAL    DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
     &        RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG)

C                     LIST OF VARIABLES
C
C   DELTA()       1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                   SIG() CALCULATION
C   I             CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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 IN 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   MO()          1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                   MEAN FOR EACH REGION
C   MPNT()        1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                   MZERO() FOR EACH REGION
C   MZERO()       2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                   EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                   IS THE UPPER BOUND
C   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
```

```
C                  S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
      DO 100 J = 0, MAXSET
         NPTS(J) = 0.0
  100 CONTINUE

      DO 200 L = 1, MAXREG
         DO 250 J = 0, MAXSET
            NP(J, L) = 0.0
  250    CONTINUE
  200 CONTINUE

      DO 300 J = 0, MAXSET
         DO 350 I = 1, MAXDAT
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            RATSTR(I,J) = 0.0
  350    CONTINUE
  300 CONTINUE

      DO 400 L = 1, MAXREG
         DO 425 K = 1, MAXDAT
            DO 450 J = 0, MAXSET
               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"
C  THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C  SPECFO AND RELATO.  RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C  SPECIFIED BY USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  21JUN88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
     &                LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
     &                NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
     &                FTUZ, FTYZ, DELTA, MO, SIGMA2, 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,
C            CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2,
C            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),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RATIO, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG), SZERO

      CHARACTER*40  DESCRP(0:MAXSET)

C                          LIST OF VARIABLES
C
C  COUNT       INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO,
C                CONVERSION, AND BREAK UP
C  CZERO       EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                COEFFICIENT OF VARIATION, Co
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                SIG() CALCULATION
C  DESCRP()    1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C  FTU         ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
C  FTUZ        ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTY         YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ        YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  I           CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LAMN        LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                CONSTANT OVER ALL REGIONS AND COMPONENTS
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  M           CONTROLS DO LOOP FOR EACH DATA DIVISION
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN 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  MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                MEAN FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MPROC       Materials PROcess variation -- CONTROLS MATERIALS PROCESS
C                VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C  NDIV        NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
C                REGION PAIRS DURING INPUT
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NNODAT      Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                IN EACH REGION
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUM         NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  RATIO       STRESS RATIO (R = -1.0 IS DESIRED)
```

```
C   RATSTR()     2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C                  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) OR 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   REG          REGION OF INTEREST IN A PARTICULAR DIVISION
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
C                  S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZERO        STRESS TENSILE TEST POINT, So
C   VARY         CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                  VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                  VARIATION;  3 - TRUNCATED NORMAL VARIATION
C   ZROREG       ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                  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), FTY, FTU, NDIV, NPTS(0)

      IF (NPTS(0) .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF

      WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
      IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)

      WRITE(3,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

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

      FTUZ = FTU
      FTYZ = FTY

C     INPUT STRESS/LIFE INFORMATION -- INCLUDING STRESS RATIO AND REGION
C     INFORMATION FROM SPECFD AND ECHO TO SPECFO

      DO 100 M = 1, NDIV

         READ (1,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE FOR RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

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

         DO 110 I = (COUNT + 1), (COUNT + NUM)
            READ(1,*) RAWSTR(I,0), RAWNF(I,0)
  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, FTY)
      ENDIF
C     ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL
      DO 130 I = (COUNT + 1), (COUNT + NUM)
         WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
     &                RATSTR(I,0), RAWNF(I,0)
         IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
     &                RATIO, REG, RATSTR(I,0), RAWNF(I,0)
 130  CONTINUE
C     BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C     EXPCTD, AND PAREST
      K = NP(0,REG)
      DO 140 I = (COUNT + 1), (COUNT + NUM)
         K = K + 1
         LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
         LNNF(K,0,REG) = ALOG(RAWNF(I,0))
         STR(K,REG) = RATSTR(I,0)
         NF(K,REG) = RAWNF(I,0)
 140  CONTINUE
      IF (K .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF
      NP(0,REG) = K
      REFNP(REG) = K
      COUNT = COUNT + NUM
 100 CONTINUE
     IF (NPTS(0) .NE. COUNT) THEN
        WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &             'INCORRECTLY SPECIFIED'
        WRITE(8,*) 'IN SPECIFIC DATA SET'
        CALL TRMNAT
     ENDIF
     READ(1,*) SZERO
     IF (NINT (SZERO) .GT. 0) THEN
        ZROREG = 0
     ELSE
        ZROREG = 1
     ENDIF
     IF (IOUT .EQ. 10)
     &  WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG
C    INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION
```

```
      READ(1,*) NUMREG, NNODAT

      IF ((NUMREG + NNODAT) .GT. MAXREG)  THEN
         WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON NUMBER OF REGIONS'
         CALL TRMNAT
      ENDIF

      DO 150 L = ZROREG, (NUMREG + NNODAT)
         READ(1,*) NBND(L)
  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
         ENDIF
  180 CONTINUE

      IF (VARY .EQ. 3) THEN
C        READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
         WRITE(3,945)
         IF (IOUT .EQ. 10) WRITE(8,945)
         DO 190 L = 1, (NUMREG + NNODAT)
            READ(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
            ENDIF
  190    CONTINUE
      ENDIF

      IF (MPROC .EQ. 1) THEN
         READ(1,*) KRATIO, LAMN
         WRITE(3,955) KRATIO, LAMN
         IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
      ENDIF

C  BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATD
C  AND THEN ECHO TO RELATO

      READ(5,*) NSETS
```

```
      IF (NSETS .GT. MAXSET) THEN
         WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF RELATED DATA SETS'
         CALL TRMNAT
      ENDIF

      WRITE(6,935) NSETS

      DO 200 J = 1, NSETS

         COUNT = 0

         IF (IOUT. EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS

         READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)

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

         WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
         IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)

         WRITE(6,905)
         IF (IOUT .EQ. 10) WRITE(8,905)

         DO 300 M = 1, NDIV

            READ(5,*) NUM, RATIO, REG

            IF (ABS(RATIO) .GT. 1.0) THEN
               WRITE(8,*) 'ERROR:  INVALID VALUE OF RATIO:   ', RATIO
               CALL TRMNAT
            ENDIF

            IF (REG .GT. MAXREG) THEN
               WRITE(8,*)
     &            'ERROR:  OVER REGION LIMIT IN RELATED MATERIAL ', J
               CALL TRMNAT
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
               WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
            ENDIF

            DO 310 I = (COUNT + 1), (COUNT + NUM)
               READ(5,*) RAWSTR(I,J), RAWNF(I,J)
 310        CONTINUE

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,I2)

905 FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
&          /,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
&          'STRESS',7X,'LIFE'/)

910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

913 FORMAT(//)

914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN So OF',
&          5X,E11.5)

915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA  ',
&          /,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
&          /,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
&          '(CYCLES): ',/)

920 FORMAT(10X,E9.3)

925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
&          'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
&          //,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
&          //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
&          5X,'UPPER BOUND',/)

930 FORMAT(6X,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**************************************************************************


C  THIS SUBROUTINE PERFORMS THE TRANSFORMATION ON STR() WHEN THE
C  STRESS RATIO, R, IS NOT -1.0
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
C
C  FTU         ULTIMATE STRENGTH OF MATERIAL (PSI)
C  FTY         YIELD STRENGTH OF MATERIAL (PSI)
C  I           CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C  IOUT        OUTPUT DUMP CONTROLLER
C  J           DATA SET OF INTEREST
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C  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 * (1 - R)/2 , TO BE COMPARED WITH FTY

C  Kt IS ASSUMED TO BE ONE

       DO 100 I = NUM1, NUM2

          TEST = STR(I,J) * (1.0 - R)/2.0

          IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST
```

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

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

C  SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C  AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C  X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  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               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   NUMREG, NSETS, NP, LNSTR, LNNF
C  OUTPUTS:  SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR

C     IMPLICIT NONE

      INTEGER 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)
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                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
C               L AND DATA SET J (X = Ln S)
C  MEANY()    1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
C               L AND DATA SET J (Y = Ln N)
C  NP()       2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C               SET IN EACH REGION
C  NPPR()     1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C               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
C               REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SWHAT2()   1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C               REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
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  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

         DO 200 J = 0, NSETS
C           FIRST CALCULATE SAMPLE X AND Y MEANS
C           FOR DATA SET J IN REGION L
            MEANY(J) = 0.0
            MEANX(J) = 0.0
            IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
     &                        ' NP =', NP(J,L)

            DO 250 K = 1, NP(J,L)
               MEANY(J) = MEANY(J) + LNNF(K,J,L)
               MEANX(J) = MEANX(J) + LNSTR(K,J,L)
               IF (IOUT .EQ. 10) WRITE(8,*) 'LNNF =', LNNF(K,J,L),
     &               ' LNSTR =', LNSTR(K,J,L)
 250        CONTINUE

            MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
            MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
            IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY(J) =', MEANY(J),
     &         ' MEANX(J) =', MEANX(J)
```

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

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

            NPPR(L) = NPPR(L) + NP(J,L) - 1
            IF (IOUT .EQ. 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
C           OUR MODEL
            WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
            CALL TRMNAT
         ENDIF

         NPPR(L) = NPPR(L) - 1

         IF (NPPR(L) .LE. 0) THEN
            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))

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,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
     &                 ' SX2(L) =', SX2(L)
            WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
     &                 ' BB(L) =', BB(L)
         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) =',
     &            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 INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, Io, ON
C   C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C   PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5OCT87   COMMENTS:  15SEP89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                   JZERO, MCHAT)

C   INPUTS:   NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C   OUTPUTS:  IZERO, JZERO, MCHAT
C   SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER CHITAB, MAXREG, TTAB

      PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)

      COMMON  IOUT

      INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG

      REAL    ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
     &        IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
     &        SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
     &        SX2(MAXREG), T, T025(TTAB)

      DATA (CHI025(I), I = 1, 75) /
     &          0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
     &          1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
     &          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,
     &          29.15,    29.95,   30.75,   31.55,   32.3574,
     &          33.16,    33.96,   34.77,   35.58,   36.39,
     &          37.21,    38.02,   38.84,   39.66,   40.4817,
     &          41.30,    42.12,   42.95,   43.77,   44.60,
     &          45.43,    46.26,   47.09,   47.92,   48.7576,
     &          49.59,    50.42,   51.26,   52.10,   52.94 /
      DATA (CHI025(I), I = 76, 150) /
     &          53.78,    54.62,   55.46,   56.30,   57.1532,
     &          57.80,    58.84,   59.69,   60.54,   61.39,
     &          62.24,    63.09,   63.94,   64.79,   65.6466,
     &          66.50,    67.35,   68.21,   69.07,   69.92,
     &          70.78,    71.64,   72.50,   73.36,   74.2219,
     &          75.08,    75.94,   76.80,   77.67,   78.53,
     &          79.40,    80.27,   81.13,   82.00,   82.87,
     &          83.73,    84.60,   85.47,   86.34,   87.21,
     &          88.08,    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 /

      DATA (CHI975(I), I = 1, 75) /
     &          5.02389,  7.37776, 9.34840, 11.1433, 12.8325,
```

```
&               14.4494,   16.0128,  17.5346,  19.0228,  20.4831,
&               21.9200,   23.3367,  24.7356,  26.1190,  27.4884,
&               28.8454,   30.1910,  31.5264,  32.8523,  34.1696,
&               35.4789,   36.7807,  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,
&               54.44,     55.67,    56.89,    58.12,    59.3417,
&               60.56,     61.77,    62.99,    64.20,    65.41,
&               66.62,     67.82,    69.02,    70.22,    71.4202,
&               72.61,     73.81,    75.00,    76.19,    77.38,
&               78.57,     79.75,    80.93,    82.12,    83.2976,
&               84.48,     85.65,    86.83,    88.00,    89.18,
&               90.35,     91.52,    92.69,    93.86,    95.0231,
&               96.19,     97.35,    98.52,    99.68,    100.84 /
  DATA (CHI975(I), I = 76, 150) /
&               102.00,    103.16,   104.31,   105.47,   106.629,
&               107.78,    108.94,   110.09,   111.24,   112.39,
&               113.54,    114.69,   115.84,   116.99,   118.136,
&               119.28,    120.43,   121.57,   122.72,   123.86,
&               125.00,    126.14,   127.28,   128.42,   129.561,
&               130.70,    131.84,   132.98,   134.11,   135.25,
&               136.38,    137.52,   138.65,   139.79,   140.92,
&               142.05,    143.18,   144.31,   145.44,   146.57,
&               147.70,    148.83,   149.96,   151.09,   152.21,
&               153.34,    154.47,   155.59,   156.72,   157.84,
&               158.97,    160.09,   161.21,   162.33,   163.46,
&               164.58,    165.70,   166.82,   167.94,   169.06,
&               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 /

C  VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:
C
C     1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719
C
C     31-39, 41-49, 51-59, 61-69, 71-79, 81-89, 91-99, 101-150
C        -- CALCULATED USING CUBE RULE APPROXIMATION

  DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
&              2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
&              2.160, 2.145, 2.131, 2.120, 2.110, 2.101,
&              2.093, 2.086, 2.080, 2.074, 2.069, 2.064,
&              2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /


C                        LIST OF VARIABLES
C
C  ARG         INTERMEDIATE CALCULATION VARIABLE
C  CHI025()    TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHI975()    TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHITAB      MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  I           CONTROLS LOOP FOR CHI025() AND CHI975()
C  IOUT        OUTPUT DUMP CONTROLLER
C  IZERO()     2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                FOR EACH REGION
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
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  NUM         EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  SUHAT       EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SUHAT2()    1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT       EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
```

```
C  SWHAT2()   1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C               REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SX         EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
C  SX2()      1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C               (X = Ln S)
C  T          VALUE OF T025() USED IN CALCULATIONS
C  T025()     TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
C  TTAB       MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C  INITIALIZE IZERO, JZERO AND MCHAT

      DO 50 L = 1, MAXREG
         IZERO(1,L) = 0.0
         IZERO(2,L) = 0.0
         JZERO(1,L) = 0.0
         JZERO(2,L) = 0.0
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
   50 CONTINUE

C  CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

      DO 75 L = 1, NUMREG
         IF (NPPR(L) .GT. CHITAB) THEN
            WRITE(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,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
     &                 IZERO(2,L)
            WRITE(8,*) 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
     &                 JZERO(2,L)
            WRITE(8,*) 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
```

```
     &                    MCHAT(2,L)
        ENDIF
 100 CONTINUE

      RETURN
      END


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


C   SUBROUTINE FINDMC CALCULATES THE CONSTRAINED M RANGES BASED UPON
C   THE Co GIVEN BY THE USER
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  8OCT87      COMMENTS:  13JUL89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

C   INPUTS:   NUMREG, CZERO, SX2, SXY, SY2
C   OUTPUTS:  MCPNT, MC

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, MCPNT(MAXREG), NUMREG

      REAL     ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
     &         SXY(MAXREG), SY2(MAXREG)


C                     LIST OF VARIABLES
C
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     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         MCPNT(L) = 0
         MC(1,L) = 0.0
         MC(2,L) = 0.0
  50 CONTINUE

C     BEGIN CALCULATIONS
```

```
      CZERO2 = CZERO ** 2

      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

      DO 100 L = 1, NUMREG

         ARG1 = SX2(L) - CZERO2
         ARG2 = 0.0

         IF (CZERO .EQ. 0.0) THEN

C           THEN NO M CONSTRAINT IS REQUIRED

            MCPNT(L) = 0

         ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

C           THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M

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

         ELSE

C           THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C           COMMON CALCULATIONS

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

            IF (SX2(L) .LT. CZERO2) THEN

C              AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

               MCPNT(L) = 1
               MC(1,L) = (- SXY(L) - ARG2) / ARG1

            ELSE

C              SX2(L) .GT. CZERO2 -- THIS TIME THE M CONSTRAINT IS A RANGE

               MCPNT(L) = 2
               MC(1,L) = (- SXY(L) - ARG2) / ARG1
               MC(2,L) = (- SXY(L) + ARG2) / ARG1

            ENDIF

         ENDIF

  100 CONTINUE

      IF (IOUT .EQ. 10) THEN

         DO 200 L = 1, NUMREG

            WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
            WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

  200    CONTINUE

      ENDIF

      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
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM 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

      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


C                        LIST OF VARIABLES
C
C   IOUT        OUTPUT DUMP CONTROLLER
C   J           CONTROLS DO LOOP FOR EACH DATA SET
C   L           CONTROLS DO LOOP FOR EACH REGION
C   LAMN        LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                 CONSTANT OVER REGIONS AND COMPONENTS
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                 MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   NP()        2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                 SET IN EACH REGION
C   NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUM()       EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
C                 NUMBER OF POINTS IN EACH DATA SET
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   PSIG2()     1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C                 VARIATION IN EACH REGION
C   PVAR        THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                 CURVE WARRANTED BY THE AVAILABLE INFORMATION
C   SUM         WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
C                 AVERAGE
C   TOTAL       SUM OF NUM() OVER ALL REGIONS


C     INITIALIZE VARIABLES

      SUM = 0.0
      TOTAL = 0.0

      DO 50 L = 1, MAXREG
         PSIG2(L) = 0.0
         NUM(L) = 0
   50 CONTINUE

      DO 100 L = 1, NUMREG
         DO 150 J = 0, NSETS
```

```
         NUM(L) = NUM(L) + NP(J,L)
150     CONTINUE
        NUM(L) = NUM(L) - 1
        TOTAL = TOTAL + NUM(L)
100 CONTINUE

    DO 200 L = 1, NUMREG
       PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
       SUM = SUM + PSIG2(L) * NUM(L)
200 CONTINUE

    IF (IOUT .EQ. 10) THEN
       WRITE(8,*) 'LAMN = ', LAMN
       DO 300 L = 1, NUMREG
          WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
          WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
300     CONTINUE
        WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
    ENDIF

    PVAR = SUM / FLOAT (TOTAL)

    RETURN
    END

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

C  SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C  M AND Co WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C  TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
     &                   MCHAT, RANGEM)

C  INPUTS:    NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
     &        MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER

C                             LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LOWER       LOWER BOUND OF INTERSECTION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                BOUND
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION -- MCHAT(1,L) = - DD(L), THE ESTIMATE
C                FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
```

```
C  MCPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  UPPER        UPPER BOUND OF INTERSECTION

C     INITIALIZE VARIABLES

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

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG

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

         IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C           THERE IS NO EXOGENOUS INFORMATION
C           ASSUME RANGE TO BE Jo

            RANGEM(1,L) = JZERO(1,L)
            RANGEM(2,L) = JZERO(2,L)

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' JZERO(1,L) = ', JZERO(1,L)
               WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

C           NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
C           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

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

C           THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
C           CORRESPONDING TO THE Co CONSTRAINT, ADJUST Jo ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MC(1,L))
```

```
         UPPER = AMIN1(JZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

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

C        THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER
C        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
C        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)
            WRITE(8,*) '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

C        THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C        CONSTRAINT, INTERSECT Jo AND Mo, ADJUSTING THE LOWER BOUND
C        BY Mc ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                 'AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF
```

```
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANEGM(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
C           INTERSECT THESE TWO RANGES WITH Jo

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                    ' RANGEM(2,L) = ', RANGEM(2,L)
            ENDIF

         ELSE

            WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
            CALL TRMNAT

         ENDIF

C        RESTRICT RANGE TO BE NON-NEGATIVE

         RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

         IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END

C***********************************************************************
```

```
C  SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C  WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  INPUTS:    RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C  OUTPUTS:   RANGEM, MCHAT, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

      REAL    MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)


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

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           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

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
            ELSE
```

```
            MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
         ENDIF
         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


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
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE CONCAV (NUMREG, RANGEM)

C  INPUTS:     NUMREG, RANGEM
C  OUTPUTS:    RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    RANGEM(2, MAXREG), TESTM


C                             LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
C  L           CONTROLS DO LOOP FOR EACH REGION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C  TESTM      UPPER BOUND OF RANGE ON M IN REGION L-1 -- USED DURING
C               CONCAVITY ADJUSTMENT


C     ADJUST RANGE TO INSURE CONCAVITY

      DO 100 L = NUMREG, 2, -1

         IF (RANGEM(2,L-1) .EQ. 0.0) THEN
C           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
C           RANGE IS AN INTERVAL IN REGION L-1
            TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
            IF (TESTM .LT. RANGEM(1,L-1)) THEN
```

```
            WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &        ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
     &        'REGION ', L-1
            CALL TRMNAT
         ELSE
            RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
         ENDIF
      ENDIF

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'RANGEM(1,L-1) =', RANGEM(1,L-1),
     &               ' RANGEM(2,L-1) =', RANGEM(2,L-1)
         WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &               ' RANGEM(2,L) =', RANGEM(2,L)
         WRITE(8,*) 'TESTM =', TESTM, ' L = ', L
      ENDIF

  100 CONTINUE

      RETURN
      END


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


C  SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C  BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5OCT87      COMMENTS:  1DEC87
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C  INPUTS:    NUMREG, RANGEM
C  IOUTPUT:   MEDM

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON    IOUT

      INTEGER  IOUT, L, NUMREG

      REAL      LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)


C                    LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LOWERM      LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
C                TO BE USED IN MEDIAN CALCULATION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEDM()      1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND

C     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
C           -- 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)
            WRITE(8,*) '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
C     VERSION:  MATCHR V8.3, V8.4, V8.5   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 EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
     &                   ZROREG, NBND, BIGK1, BZHAT)

C  INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C  OUTPUTS:  BIGK1, BZHAT
C  SUBPROGRAMS:  TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL    BIGK(0:MAXREG), BIGK1, 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)

C                    LIST OF VARIABLES
C
C   BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                EACH REGION
C   BIGK1      EQUAL TO BIGK(1)
C   BZHAT      E(BETAo)
C   FACTR      A SCALE FACTOR = PHI * KRATIO * Z
C   IOUT       OUTPUT DUMP CONTROLLER
C   KHAT       E(k)
C   L          CONTROLS DO LOOP FOR EACH REGION
C   MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C   MEDM()     1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C   MM()       1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C   NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST
C   NCOMPS     Number of COMPonents -- 1 FOR STRESS AND STRAIN WHEN DECOMPOSED
C                DATA UNAVAILABLE -- 2 FOR DECOMPOSED STRAIN DATA
C   NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                DATA SET
C   NPTS()     1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR
C                THE SPECIFIC MATERIAL S/N DATA SET
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 REGION
C                CONTAINED IN NBND()
C   STR()      2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C                DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZ2        SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C   SZERO      STRESS TENSILE TEST POINT, So
C   TRBIGK()   1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C                TRBIGK(i) = BIGK(i)
C   ZROREG     ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                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  INITIALIZE VARIABLES

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

C  CREATE MM() ARRAY FROM MEDM() ARRAY

      DO 100 L = 1, NUMREG
         MM(L) = MEDM(L)
  100 CONTINUE

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

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

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE BETAo AND k

      CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C  CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

      CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

      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), SBND(L)
             IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
     &                                      NBND(L), SBND(L)
  200     CONTINUE

          WRITE(7,920)

       ELSE

          WRITE(7,930) MM(1), BIGK(1), KHAT

       ENDIF

C   FORMAT STATEMENTS

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

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

  920 FORMAT(///)

  930 FORMAT(//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &        //,11X,'m',14X,'K',13X,'E(k)',
     &        //,7X,F8.5,5X,E12.5,6X,F7.4,/)

       RETURN
       END


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


C   SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C   MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  21JUN88       COMMENTS:  13JUL89
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

       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
C  ARG         INTERMEDIATE CALCULATION VARIABLE
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
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                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  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 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  SUMX2       EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
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     INITALIZE ARRAYS

      DO 50 L = 1, MAXREG
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
         MU(L) = 0.0
         SIG(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATION FOR EACH REGION

      DO 100 L = 1 , NUMREG

         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SQRT (SUHAT2(L))
         SUMX2 = NPPR(L) * SX2(L)
         ARG = SUMX2 + DELTA(L)

         IF (DELTA(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C           USE THE ESTIMATE OF M
            MU(L) = MCHAT(1,L)
         ELSE
C           UPDATE THE ESTIMATE OF M WITH Mo USING DELTA
            MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
         ENDIF
```

```
         IF (SIGMA2(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
C           USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
            SIG(L) = SQRT (SWHAT2(L) / ARG)
         ELSE
            SIG(L) = SQRT (SIGMA2(L) / ARG)
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
     &                MCHAT(1,L)
            WRITE(8,*) 'SUHAT2 = ', SUHAT2(L),  ' MCHAT2 = ',
     &                MCHAT(2,L)
            WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
     &                ' SUMX2 = ', SUMX2
            WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
            WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
            WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
     &                ' SIG = ', SIG(L)
         ENDIF

  100 CONTINUE

      RETURN
      END

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     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  INPUTS:   NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
     &        RANGEM(2, MAXREG), UPPER


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

```
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  UPPER       UPPER BOUND OF INTERSECTION

C     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
C           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 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. 1)) THEN

C           THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C           CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc

            LOWER = AMAX1(MZERO(1,L), MC(1,L))
            UPPER = MZERO(2,L)
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
```

```
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANEGM(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
C        INTERSECT THESE TWO RANGES

         LOWER = AMAX1(MZERO(1,L), MC(1,L))
         UPPER = AMIN1(MZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSE

         WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
         CALL TRMNAT

      ENDIF

C     RESTRICT RANGE TO BE NON-NEGATIVE

      RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

      IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &       .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &       WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                  'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END

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

C   SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C   DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  10FEB88      FORMAT/COMMENTS:  12AUG91
C      VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDRGN (RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG,
```

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


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


      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
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            MU(LL) = MO(LL)
            SIG(LL) = SQRT(SIGMA2(LL))
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
               WRITE(8,*) 'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
               WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
     &                    SIGMA2(LL)
```

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
               MU(LL) = RANGEM(1,LL)
               SIG(LL) = 0.0
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
     &         ' MU = ', MU(LL), ' SIG = ', SIG(LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END


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


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               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  INPUTS:     VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG, ZROREG, RAND,
C              NBND, STR
C  OUTPUTS:    BIGK, BZERO, MM, SBND
C  SUBPROGRAMS:  FINDM, FINDMN, TRNSFM, SMNVAR, KBETA, FINDK, FINDSB

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG

      REAL      BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG),
     &          MU(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &          RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
     &          STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

      DOUBLE PRECISION RAND

C                       LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C  IOUT      OUTPUT DUMP CONTROLLER
```

```
C  K         VALUE OF k -- PARAMETER CHARACTERIZING SPECIFIC MATERIAL DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT    MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ     SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C              DISTRIBUTION MEAN FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NF()      2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C              SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP        TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()    1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C              SPECIFIC MATERIAL S/N DATA SET
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  RANGEM()  2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C              FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C              RANGEM(2,L) IS THE UPPER BOUND
C  RAND      RANDOM NUMBER SEED
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()
C  SIG()     1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
C              STANDARD DEVIATION FOR EACH REGION
C  STR()     2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C              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;
C              1 - S/N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;
C              3 - TRUNCATED NORMAL VARIATION
C  ZROREG    ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO REGION
C  ZZ()      1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C              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, 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,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. 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

      SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C  INPUTS:   RAND, NUMREG, RANGEM
C  OUTPUTS:  MM
C  SUBPROGRAMS:  RANDOM, TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X

      DOUBLE PRECISION RAND

C                     LIST OF VARIABLES
C
C  IOUT           OUTPUT DUMP CONTROLLER
C  L              CONTROLS DO LOOP FOR EACH REGION
C  MAXREG         MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()           1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NUMREG         NUMBER OF REGIONS OF INTEREST
C  PICK()         1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C  RAND           RANDOM NUMBER SEED
C  RANGEM()       2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                   FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                   RANGEM(2,L) IS THE UPPER BOUND
C  X              UNIFORM(0,1) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                   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)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              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),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF

  100 CONTINUE

       RETURN
       END


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


C**********************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C        Random Number Generator, JPL Publication 85-98, JPL Document
C        5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C**********************************************************************
```

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

C               LIST OF VARIABLES
C
C   FRAC      UNIFORM (0,1) RANDOM VARIATE
C   IOUT      OUTPUT DUMP CONTROLLER
C   RANA      CONSTANT FOR LCG
C   RANC      CONSTANT FOR LCG
C   RAND      RANDOM NUMBER SEED
C   RANDIV    INTERNAL CALCULATION
C   RANM      CONSTANT FOR LCG
C   RANSUB    INTERNAL CALCULATION
C   RANT      INTERNAL CALCULATION
C   RANX      INTERNAL CALCULATION


C         USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

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

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

      RETURN
      END

C     NOTES:  IOUT=2 DUMPS TO SCREEN


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


C  SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13FEB89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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

C                         LIST OF VARIABLES
C
C   IOUT        OUTPUT DUMP CONTROLLER
C   L           CONTROLS DO LOOP FOR EACH REGION
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()        1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C   MU()       1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   PICK()      1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C   RAND        RANDOM NUMBER SEED
C   RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C   SIG()      1-D ARRAY CONTAINING THE STANDARD DEVIATION OF M FOR EACH
C                 REGION
C   X           NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                 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
   10       CALL NORMGN (RAND, MU(L), SIG(L), X)
            IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
            MM(L) = X
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                    ' RANGEM(2,L) =', RANGEM(2,L)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              SAMPLE ON ADJUSTED RANGE
   20          CALL NORMGN (RAND, MU(L), SIG(L), X)
               IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
               MM(L) = X
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
```

```
&                        ' RANGEM(1,L) =', RANGEM(1,L)
            WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
            WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
&                        ' MM(L) =', MM(L)
         ENDIF
      ENDIF

  100 CONTINUE

      RETURN
      END


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


C*************************************************************************
C  SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C  WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  3FEB88
C     VERSION:  MATCHR V7, V7.1, 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
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C*************************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT


C             LIST OF VARIABLES
C
C  FRAC     UNIFORM(0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  MU       MEAN OF NORMAL DISTRIBUTION
C  RAND     RANDOM NUMBER SEED
C  SIGMA    STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X        NORMAL RANDOM VARIATE
C  U1       UNIFORM RANDOM NUMBER U(0,1)
C  U2       UNIFORM RANDOM NUMBER U(0,1)
C  Z1       NORMAL RANDOM NUMBER ON N(0,1)
C  Z2       NORMAL RANDOM NUMBER ON N(0,1)


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

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

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

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

      X = SIGMA * Z1 + MU
      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 V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:    NPTS, STR, NF, NUMREG, MM, NBND
C  OUTPUTS:   NP, ZZ

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON IOUT

      INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG

      REAL    MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        STR(MAXDAT, MAXREG), ZZ(MAXDAT)


C                         LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR EACH DATA POINT
C  IOUT       OUTPUT DUMP CONTROLLER
C  K          CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LL         CONTROLS INNER DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C  MML        EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C               SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()     1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C               SPECIFIC MATERIAL S/N DATA SET
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  STR()      2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C               S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  ZZ()       1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)


C  INITIALIZE VARIABLES

      NP = 0

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

```
C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG
         MML = MM(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =',
     &        MML, ' NPTS =', NPTS(L)

         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))
     &                  * ((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) =',
     &            MM(LL), ' ZZ =', ZZ(NP)
  300       CONTINUE

  200    CONTINUE

  100 CONTINUE

      RETURN
      END

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

C  SUBROUTINE SMNVAR CALCULATES THE Sample MeaN and VARiance OF
C  Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  24AUG87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V3.3, V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
C  I           CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT        OUTPUT DUMP CONTROLLER
C  MAXDAT      MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MEANZ       SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                DATA SET
C  SZ2         SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZZ()        1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C                Z = F(STR,NF,NBND,MM)
```

```
C  INITIALIZE VARIABLES

      MEANZ = 0.0
      SZ2 = 0.0

C  CALCULATE THE MEAN OF ZZ(), MEANZ

      DO 100 I = 1, NP
         MEANZ = MEANZ + ZZ(I)
         IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I,
     &        ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
  100 CONTINUE
      MEANZ = MEANZ / FLOAT(NP)
      IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ

C  CALCULATE THE VARIANCE OF ZZ(), SZ2

      DO 200 I = 1, NP
         SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
         IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
  200 CONTINUE
      SZ2 = SZ2 / FLOAT(NP - 1)
      IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

      RETURN
      END


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


C  SUBROUTINE KBETA CALCULATES k AND BETAo FROM THE SAMPLE MEAN AND
C  VARIANCE OF Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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


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

```
C   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
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

       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)

C                        LIST OF VARIABLES
C
C   BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C               FOR EACH REGION
C   BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
C               MATERIAL DATA BASE
C   GAMMA     EULER'S CONSTANT
C   IOUT      OUTPUT DUMP CONTROLLER
C   K         VALUE OF k -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C               DATA BASE
C   L         CONTROLS DO LOOP FOR EACH REGION
C   MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C   MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C   NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C   NUMREG    NUMBER OF REGIONS OF INTEREST

C   INITIALIZE VARIABLES
```

```
      DO 50 L = 0, MAXREG
         BIGK(L) = 0.0
   50 CONTINUE

C  CALCULATE K FOR REGION ONE

      BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C      WRITE(7,*) 'REGION:  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)))
C         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
C        DATE:  22DEC88
C     VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.2, V4.3, V4.4, V4.5

      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)

C              LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  IOUT      OUTPUT DUMP CONTROLLER
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  SBND()    1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CONTAINED IN NBND()
```

```
C  ZROREG     ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO REGION


C  INITIALIZE SBND()

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

C   CALCULATE SBND(0) IF ZROREG = 0

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

C  CALCULATE THE NON-ZERO REGION STRESS BOUNDARIES

      DO 100 L = 1, NUMREG
         IF (NBND(L) .GE. 1.0E+36) THEN
            SBND(L) = 0.0
         ELSE
            SBND(L) = BIGK(L) * NBND(L) ** (-1.0 / MM(L))
         ENDIF
  100 CONTINUE

      RETURN
      END


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


C  THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
C  MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C  TRANSFORM METHOD"
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  18MAR87      COMMENTS:  15SEP89
C     VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                      V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                      V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE WEIBGN (BETA, RAND, WEIB)

C  INPUTS:    BETA, RAND
C  OUTPUTS:  WEIB
C  SUBPROGRAMS:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER IOUT

      REAL    ARG, BETA, ETA, FRAC, WEIB

      DOUBLE PRECISION RAND


C                 LIST OF VARIABLES
C
C  ARG       INTERMEDIATE CALCULATION VARIABLE
C  BETA      WEIBULL DISTRIBUTION SHAPE PARAMETER
C  ETA       WEIBULL DISTRIBUTION LOCATION PARAMETER
C  FRAC      UNIFORM (0,1) RANDOM VARIATE
C  IOUT      OUTPUT DUMP CONTROLLER
```

```
C  RAND      RANDOM NUMBER SEED
C  WEIB      WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE

C      CALCULATE CONSTRAINED ETA

       ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))

C      GENERATE WEIBULL RANDOM VARIATE

       CALL RANDOM(FRAC, RAND)
       ARG = -ALOG(1.0 - FRAC)
       WEIB = ETA * ARG**(1.0/BETA)
       IF (IOUT .EQ. 10) WRITE(8,*)'BETA = ', BETA, ' ETA =', ETA,
     &    ' FRAC =', FRAC, ' ARG =', ARG, ' WEIB =', WEIB

       RETURN
       END


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


C  SUBROUTINE KOMO CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA
C  REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
C  AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
C  PROGRAMMER :  L. NEWLIN
C          DATE:  1AUG91
C       VERSION:  MATCHR V8.5    MATGRM V4.5
C
```

```

        SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
     &                   FACTR, NUMREG)

C  INPUTS:   SZERO, BIGK, MM, NBND, TRSBND, FACTR
C  OUTPUTS:  TRBIGK, MM, TRSBND

C       IMPLICIT NONE

        INTEGER  MAXREG

        PARAMETER (MAXREG = 3)

        COMMON   IOUT

        INTEGER  IOUT, L, NUMREG

        REAL     BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1           SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)


C                        LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C              EACH REGION
C  FACTR     SCALE FACTOR = PHI * KRATIO * Z
C  IOUT      OUTPUT DUMP CONTROLLER
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS
C  SCLK      ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
C  SZERO     STRESS TENSILE TEST POINT, So
C  TRBIGK()  1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
C              SBND(0) < So FOR EACH TRIAL
C  TRSBND()  1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
```

```
C             LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
C             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, ' BIGK1 = ', TRBIGK(1)
         WRITE(8,*) 'MM1 = ', MM(1), ' MMo = ', MM(0)
      ENDIF

      RETURN
      END

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


C  FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C  BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB89
C     VERSION:  MATCHR V8.3, V8.4, V8.5   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.

      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  OUTPUTS:  GTLIFE

C     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

C                           LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C               FOR EACH REGION
C  GETLIF     VALUE TO BE ASSIGNED TO GTLIFE -- CYCLES TO FAILURE FOR
C               THE REQUIRED STRESS LEVEL
C  IOUT       OUTPUT DUMP CONTROLLER
C  KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LNZ        NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  MAXREG     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               CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
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
               TEMP = MM(L) * (ALOG(BIGK(L)) - 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

      GTLIFE = GETLIF

      RETURN
      END


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


C  SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C  M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB88
C     VERSION:  MATCHR V7, V7.1, 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
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, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

      PARAMETER (MAXMM = 20001, MAXREG = 3)

      LOGICAL INORDR

      REAL    ALLM(MAXMM, MAXREG), TEMP
```

```
C                    LIST OF VARIABLES
C
C   ALLM()      2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
C   I           CONTROLS INSERTION POINTER
C   INC         SORT INCREMENT VARIABLE
C   INORDR      FLAG TO INDICATE WHETHER SORT IS FINISHED
C   IOUT        OUTPUT DUMP CONTROLLER
C   L           CONTROLS DO LOOP FOR EACH REGION
C   MAXMM       MAXIMUM NUMBER OF M'S TO BE SORTED
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   NUM         NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   TEMP        TEMPORARY SORTING VARIABLE

      DO 400 L = 1, NUMREG

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

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

            IF (.NOT. INORDR) GOTO 20
            GOTO 10
         ENDIF

400   CONTINUE

      RETURN
      END
```

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 7-7 List of Subprograms for Program MATCHR
(Footnotes are at the end of the table)

NAME	SECTION	PURPOSE
ADDREG[1]	4.1.3.9	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.
ADDRGN[1]	4.1.3.15	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.
ADJSTM	4.1.7	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.

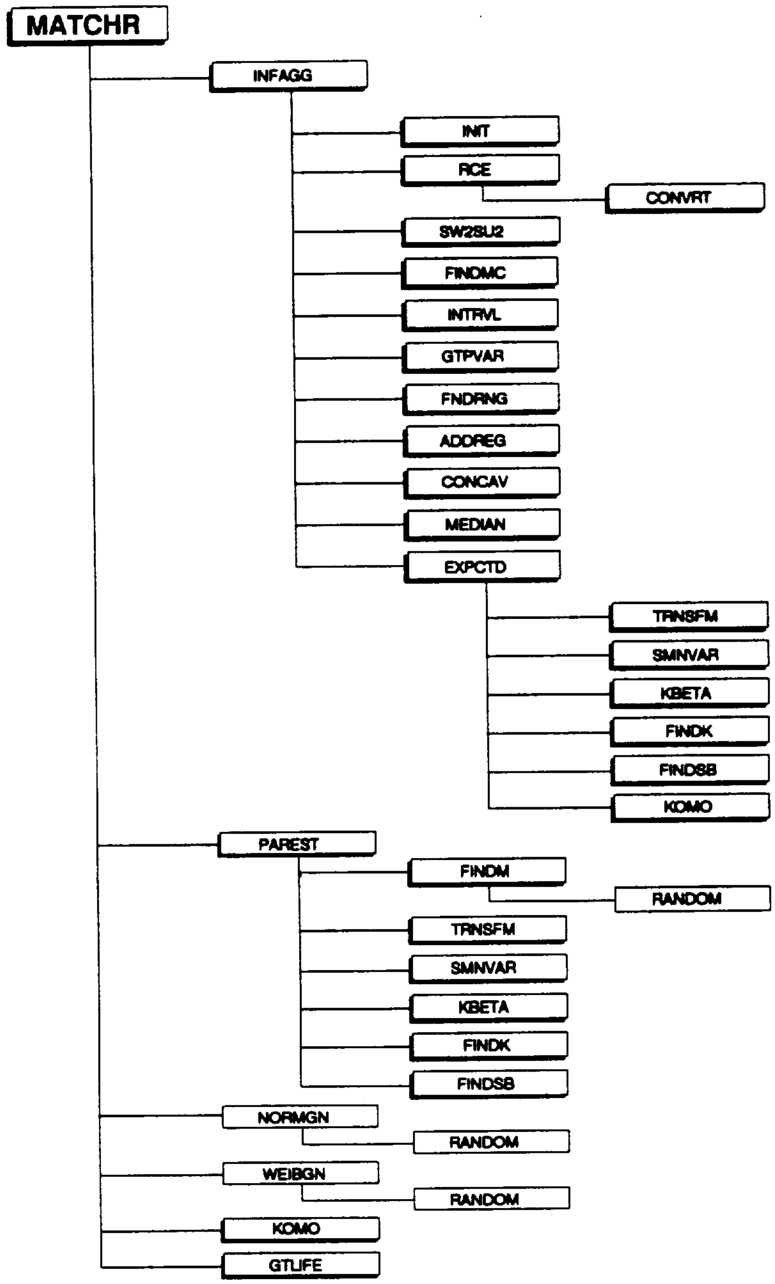

Figure 7-7 Tree Structure for the Stress Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameter *m*

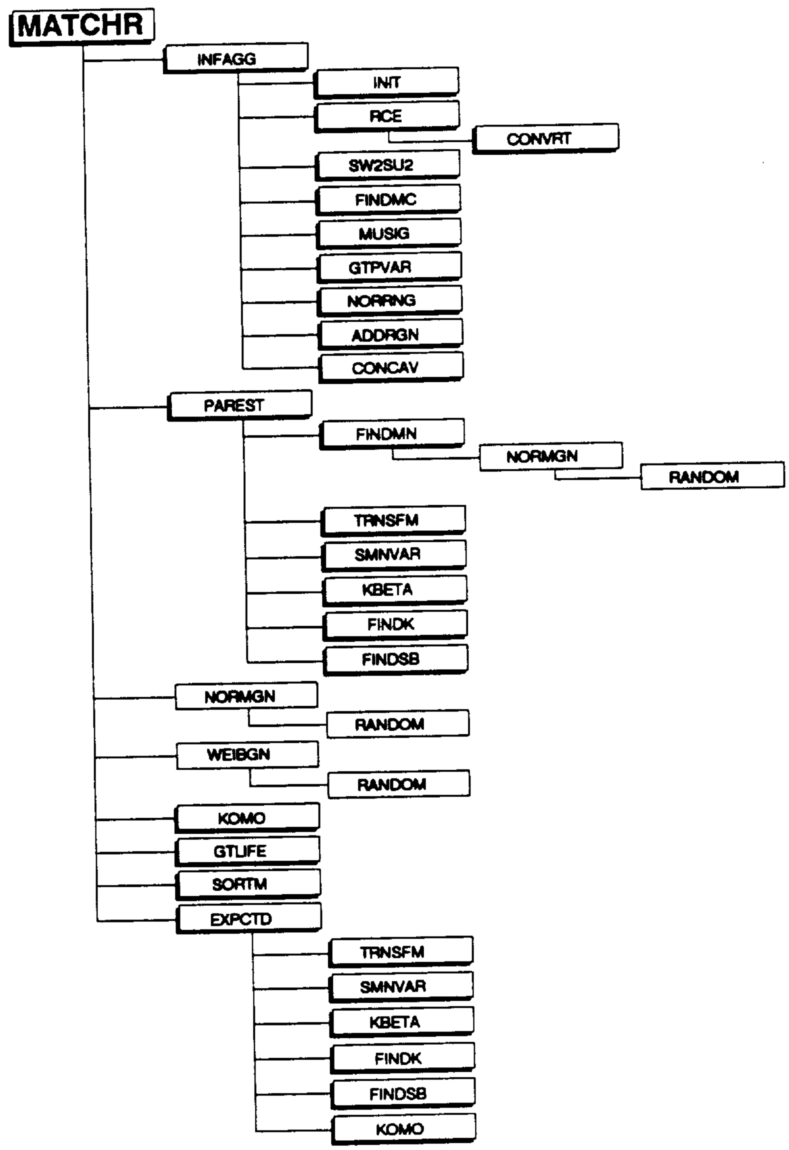

Figure 7-8 Tree Structure for the Stress Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameter *m*

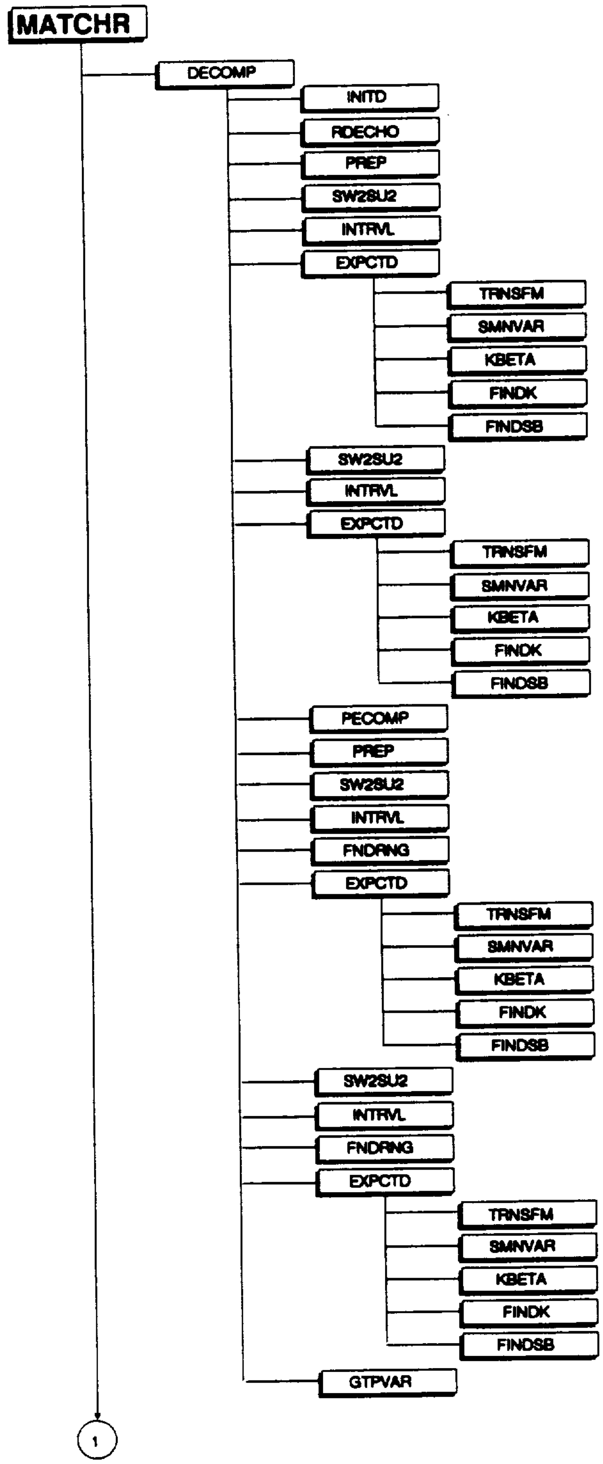

Figure 7-9 Tree Structure for the Strain Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameters m_P and m_E

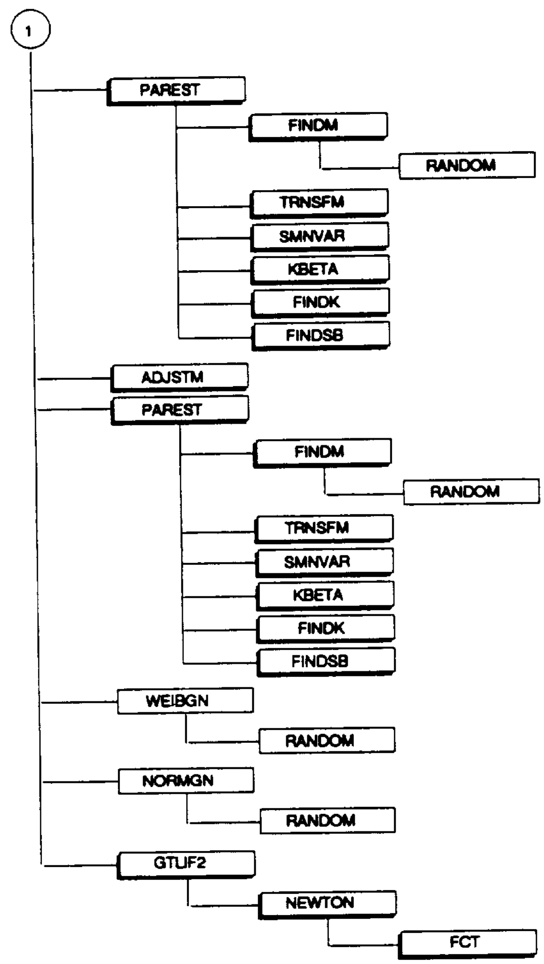

Figure 7-9 Tree Structure for the Strain Formulation of the Program MATCHR for the Uniform Variation in Materials Shape Parameters m_P and m_E (Cont'd)

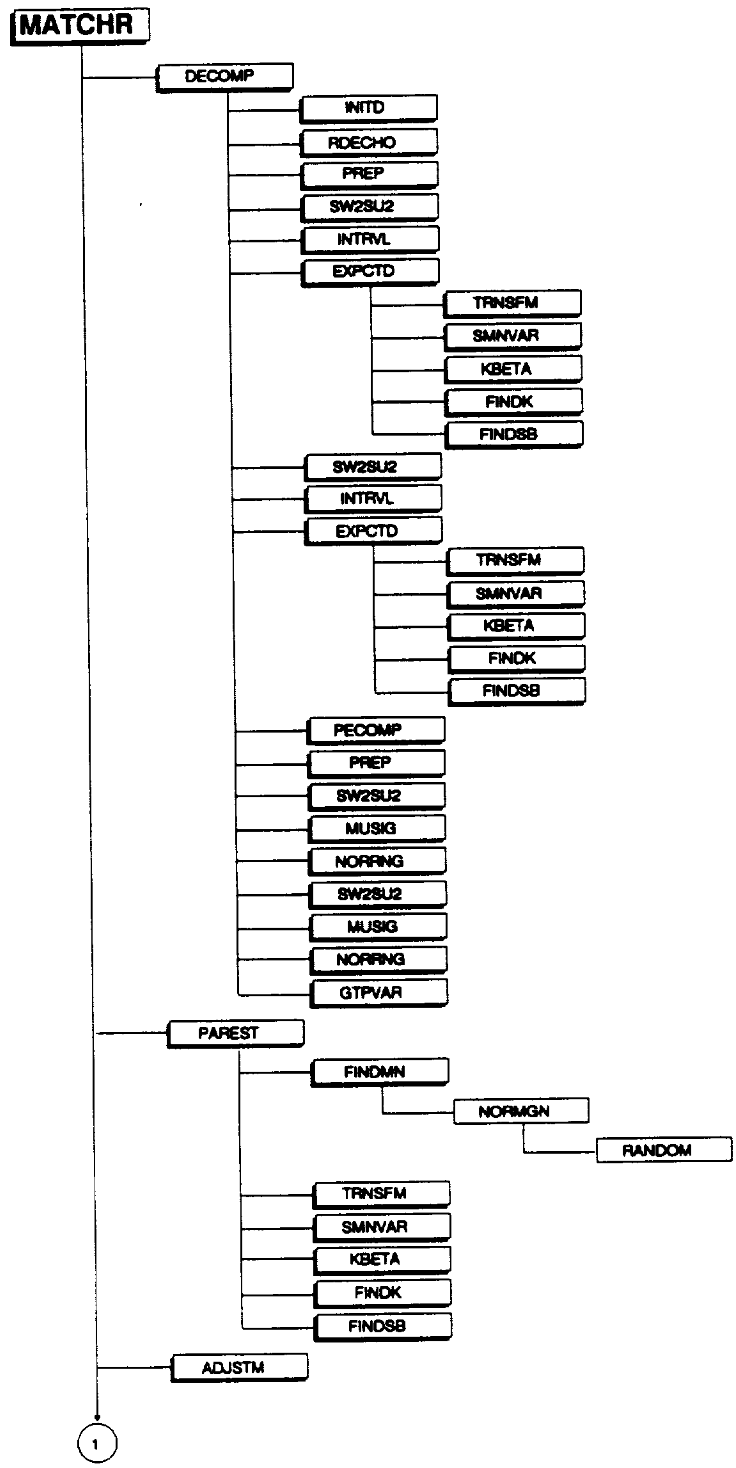

Figure 7-10 Tree Structure for the Strain Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameters m_P and m_E

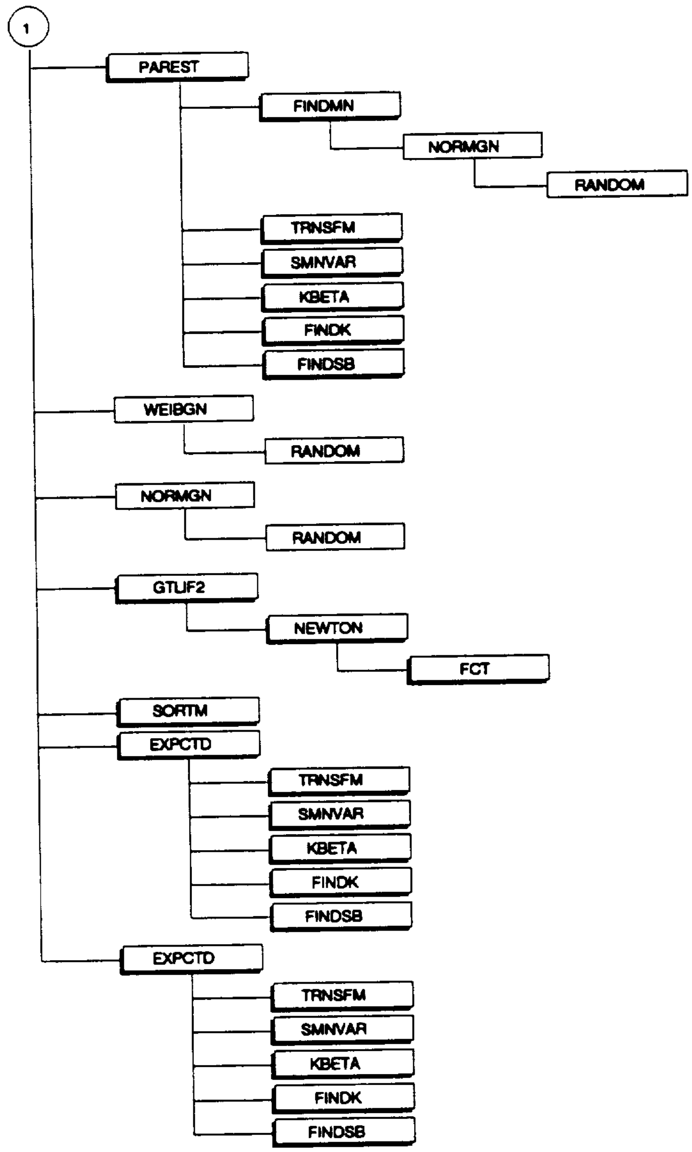

Figure 7-10 Tree Structure for the Strain Formulation of the Program MATCHR for the Truncated Normal Variation in Materials Shape Parameters m_P and m_E (Cont'd)

Table 7-7 List of Subprograms for Program MATCHR (Cont'd)

NAME	SECTION	PURPOSE
CONCAV[2]	4.1.3.10	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.
CONVRT[3]	4.1.3.3	Transforms stress data to equivalent zero-mean stresses with stress ratio of −1.0 for the stress formulation of the S/N curve.
DECOMP[4]	4.1.4	Controls the logical flow for the plastic-elastic strain decomposition and the information aggregation portion of the strain formulation materials characterization model.
EXPCTD[5]	4.1.3.12	Calculates the median S/N curve parameters from the results of the information aggregation calculations.
FCT	4.1.9.2	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*.
FINDK	4.1.5.6	Calculates the value of the location parameter K (where $A = K^m$) for each life region by using *Equations 2-37* and *2-41*.
FINDM[6]	4.1.5.1	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.
FINDMC	4.1.3.5	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.
FINDMN[6]	4.1.5.2	Obtains the value of *m* for each life region by sampling from the appropriate truncated Normal distribution on *m*.
FINDSB	4.1.5.7	Calculates the life region "tie-points" or stress values which correspond to the "life boundaries" conditional on the randomly selected *m* for each region. Also calculates K, characterizing the specific material S/N data set, which is a function of β_o and k.
FNDRNG[7]	4.1.3.8	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.
GTLIFE	4.1.8	Calculates the cycles to failure for a particular stress based upon the materials characterization model stress formulation S/N curve of *Equation 2-48*.
GTLIF2	4.1.9	Calculates the cycles to failure for a particular strain based upon the materials characterization model strain formulation S/N curve of *Equation 2-50*.
GTPVAR	4.1.3.7	Calculates σ^2, *Equation 2-49*, the extent of departures from the multiple heat median S/N curve warranted by the information available.

Table 7-7 List of Subprograms for Program MATCHR (Cont'd)

NAME	SECTION	PURPOSE
INFAGG[8]	4.1.3	Controls the logical flow for the information aggregation portion of the stress formulation materials characterization model.
INIT	4.1.3.1	Initializes the entries of the arrays used in the stress formulation information aggregation subroutine, INFAGG, to zero.
INITD	4.1.4.1	Initializes the entries of the arrays used in the strain formulation information aggregation subroutine, DECOMP, to zero.
INTRVL	4.1.3.6	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*.
KBETA	4.1.5.5	Calculates k and β_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*.
KOMO[9]	4.1.6	Calculates K_o and m_o 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.
MATCHR	4.1.2	The main routine that controls the logical flow of the calculations of the materials characterization model.
MEDIAN	4.1.3.11	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.
MUSIG[10]	4.1.3.13	Calculates the posterior Normal distribution parameters: mean m_* and standard deviation σ_*, for each life region of the S/N curve.
NEWTON	4.1.9.1	Solves the general nonlinear equations of the form $f(x) = 0$ by means of Newton's iteration method.
NORMGN[11]	4.4.3	Generates Normal(μ, σ^2) random variates.
NORRNG[7]	4.1.3.14	Combines the implicit and explicit constraints on m to obtain the posterior credibility ranges of m for each life region.
PAREST[12]	4.1.5	Controls the logical flow for the parameter estimation model portion of the materials characterization model.
PECOMP	4.1.4.4	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.
PREP	4.1.4.3	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.
RANDOM[13]	4.4.2	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0, 1) random variates.

Table 7-7 List of Subprograms for Program MATCHR (Cont'd)

NAME	SECTION	PURPOSE
RCE	4.1.3.2	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.
RDECHO	4.1.4.2	Reads the data from SPECFD and RELATD and then echoes the data to SPECFO and RELATO for the strain formulation S/N curve.
SMNVAR	4.1.5.4	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*.
SORTM[14]	4.1.10	Sorts the m values in increasing order for each life region for the truncated Normal distribution case.
SW2SU2	4.1.3.4	Calculates the residual variances from the Y on X and X on Y regressions for each life region where $Y = \ln(\textit{Endurance cycles})$ and $X = \ln(\textit{Stress})$ by using *Equations 2-20* and *2-21*; to be used in the credibility range calculations.
TRMNAT	4.1.11	Performs premature program termination, when required.
TRNSFM[15]	4.1.5.3	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.
WEIBGN	4.4.6	Generates Weibull(β, $\eta(\beta)$) random variates.

---

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)

VARIABLE NAME	TYPE	DESCRIPTION
ALLM(MAXMM, MAXREG)	RE	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]
ALPHA(MAXDAT, 0:MAXSET)	RE	2-D array containing values of the ratio of S_P to S_E, *Equation 2-47*, for each total strain point.
ARG, **ARG1**, **ARG2**	RE	Intermediate calculation variables.
BB(MAXREG)	RE	1-D array containing values of b = **SXY(L)/SY2(L)**, *Equation 2-21*, for each life region of the S/N curve.
BIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter *K*, *Equation 2-12*, where $A = K^m$.
BIGK1	RE	Dummy variable used during calls to subroutine EXPCTD, equal to **BIGK(1)**.
BIGKE(0:MAXREG)	RE	1-D array containing the value of the materials model location parameter K_E, *Equation 2-43*, for the elastic strain components.
BIGKHT	RE	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).
BIGKP(0:MAXREG)	RE	1-D array containing the value of the materials model location parameter K_P, *Equation 2-43*, for the plastic strain components.

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 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
BZERO	RE	Estimate of Weibull distribution shape parameter β_o, *Equation 2-11*, which characterizes the intrinsic variation of the S/N data set.
BZEROE	RE	Estimate of Weibull distribution shape parameter, β_{oE}, that characterizes the intrinsic variation of the elastic strain components of the S/N data set, used to estimate β_o in *Equation 2-43*.
BZEROP	RE	Estimate of Weibull distribution shape parameter, β_{oP}, that characterizes the intrinsic variation of the plastic strain components of the S/N data set, used to estimate β_o in *Equation 2-43*.
BZHAT	RE	Value of β_o corresponding to the median S/N curve.[2]
CHI025(CHITAB)	RE	Table of 0.025 percentage points for the χ^2 distribution used in *Equation 2-24*.
CHI975(CHITAB)	RE	Table of 0.975 percentage points for the χ^2 distribution used in *Equation 2-24*.
CHITAB	INT	Maximum number of degrees of freedom in **CHI025( )** and **CHI975( )**. The maximum number is 150.
COUNT	INT	Array index variable used by RCE to facilitate data input, stress ratio transformation, life region partitioning, and data echo.
CZERO	RE	C_o in *Equations 2-28* through *2-32*, the user-specified upper bound of the coefficient of variation of material fatigue strength.
CZERO2	RE	Value of **CZERO** ** 2.
DD(MAXREG)	RE	1-D array containing values of d = **SXY(L)/SX2(L)**, *Equation 2-20*, for each life region of the S/N curve.
DELTA(MAXREG)	RE	1-D array containing values of the shape parameter δ of the Bayesian prior distribution for each life region.[3]
DELTAE(MAXREG)	RE	1-D array containing the value of the shape parameter δ_E of the Bayesian prior distribution for the elastic strain components.[3]
DELTAP(MAXREG)	RE	1-D array containing the value of the shape parameter δ_P of the Bayesian prior distribution for the plastic strain components.[3]
DESCRP(0:MAXSET)	CH40	1-D array containing descriptions of each S/N data set.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
DERF	RE	DERivative of F. The resultant value of the derivative of *F* at the root during the Newton's method calculations.
DIFFX(MAXDAT, 0:MAXSET)	RE	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*.
DIFFY(MAXDAT, 0:MAXSET)	RE	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*.
ELAS(MAXDAT, 0:MAXSET)	RE	2-D array containing values of the elastic strain components S_E (%).
EPS	RE	Value of the maximum error allowed in the result of the Newton's method calculations.
F	RE	Resultant value of the function *F* at the root during the Newton's method calculations.
FACTR	RE	Value of **(PHI * KRATIO * Z)** in *Equation 2-48*.
FTU	RE	Material ultimate strength (psi).
FTUZ	RE	Material ultimate strength (psi) for specific material.
FTY	RE	Material yield strength (psi).
FTYZ	RE	Material yield strength (psi) for specific material.
GAMMA	RE	Euler's constant.
1) **GETLIF**	RE	Value to be assigned to **GTLIFE**, the fatigue cycles to failure at the required stress level.
2) **GETLIF**	RE	Value to be assigned to **GTLIF2**, the fatigue cycles to failure at the required strain level.
GTLIFE	RE	Function that calculates the fatigue cycles to failure at a given stress, given by *Equation 2-48*.
GTLIF2	RE	Function that calculates the fatigue cycles to failure at a given strain, given by *Equation 2-50*.
1) **I**	INT	Controls DO loop for each data point in an S/N data set.
2) **I**	INT	Controls the DATA statements for **CHI025( )** and **CHI975( )**.
3) **I**	INT	Controls insertion sort DO loop.
IEND	INT	Maximum number of iteration steps allowed to find the solution during the Newton's method calculations.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
IER	INT	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.
II	INT	Controls DO loop for each trial when the empirical median S/N curve for the truncated Normal case has be requested.[4]
INC	INT	Increment variable used during insertion sort.
INORDR	LOG	Logical flag to indicate whether the insertion sort has been completed.
INVME	RE	INVerse of ME, the value of 1 / **ME**.[5]
INVMP	RE	INVerse of MP, the value of 1 / **MP**.[5]
IOUT	INT	Output dump controller.
IZERO(2, MAXREG)	RE	2-D array containing values of the 95% confidence interval for C, I_0, *Equation 2-24*, for each life region.
1) **J**	INT	Controls DO loop for each S/N data set.
2) **J**	INT	S/N data set of interest during stress ratio transformation calculations.
JZERO(2, MAXREG)	RE	2-D array containing values of the 95% confidence interval for m, J_0, *Equation 2-26*, for each life region.
1) **K**	INT	Controls DO loop for each point in a life region.
2) **K**	RE	k in *Equations 2-40* through *2-42*, a parameter that characterizes the specific material S/N data base for a set of $\{m_j\}$.
1) **KE**	RE	K_E in *Equation 2-47*, the median K_E for the user-provided elastic strain components. (See **KHATE** in DECOMP)
2) **KE**	RE	Scalar variable used by function GTLIF2, equal to **BIGKE(1)**.
KEPROD	RE	Value of **KE * PHI * KRATIO * Z**.[6]
KHAT	RE	Value of k corresponding to the median S/N curve.[7]
KHATE	RE	K_E in *Equation 2-47*, the median K_E for the user-provided elastic strain components.
KHATP	RE	K_P in *Equation 2-47*, the median K_P for the user-provided plastic strain components.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
1) KP	RE	K_P in *Equation 2-47*, the median K_P for the user-provided plastic strain components. (See **KHATP** in DECOMP)
2) KP	RE	Scalar variable used by function GTLIF2, equal to **BIGKP(1)**.
KPPROD	RE	Value of KP * **PHI** * **KRATIO** * Z^6
KRATIO	RE	Ratio of *MED K*/MED K* in *Equation 2-48*. **KRATIO** is constant over life regions and strain components for the materials model.
KTERM	RE	Value of **KP** / **KE**.
L	INT	Controls DO loop for each life region of the S/N curve.
LAMBDA(MAXDAT, 0:MAXSET)	RE	2-D array containing values of λ_i, *Equations 2-46* and *2-47*, the plastic fraction of the total strain.
LAMN	RE	λ_N, the ratio of V (ln N | S, *multiple heats*) / V (ln N | S, *single heat*) or $(1 + \sigma^2 / C^2)$.[8] **LAMN** is constant over life regions and strain components.
1) LIFE	RE	The fatigue cycles to failure at a given stress or strain given by *Equation 2-48* or *2-50*.
2) LIFE	RE	Value of N for an iteration during the Newton's method calculations.
1) LL	INT	Value of **NUMREG** for a set of calculations.
2) LL	INT	Controls inner DO loop for each life region of the S/N curve.
LNA(0:MAXREG)	RE	1-D array containing values of ln(A) = ln(**BIGK**) * **MM** for each life region of the S/N curve.
LNNF(MAXDAT, 0:MAXSET, MAXREG)	RE	3-D array containing values of ln(**RAWNF( )**) indexed for life regions.
LNSTR(MAXDAT, 0:MAXSET, MAXREG)	RE	3-D array containing values of ln(**RATSTR( )**) indexed for life regions.
LNSTRE(MAXDAT, 0:MAXSET, MAXREG)	RE	3-D array containing values of ln(**SE( )**) or ln(**STRE( )**) for the elastic strain components.
LNSTRP(MAXDAT, 0:MAXSET, MAXREG)	RE	3-D array containing values of ln(**SP( )**) or ln(**STRP( )**) for the plastic strain components.
LNZ	RE	ln(Z) in *Equation 2-48*, the Normal(0, **PVAR**) random variate for the materials process variation aspect of the materials model.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LOWER	RE	The lower bound of the intersection to find $\pi(m)$ in a particular region.[9]
LOWERM	RE	max (m_l, L_{l+1}) in *Equation 2-33*, the lower bound of the *m* range, due to concavity considerations, to be used in the median calculation.[10]
LPHIM(0:MAXREG)	RE	1-D array containing values of ln(PHI) * **MM** for each life region of the S/N curve.
1) **M**	INT	Controls DO loop for each of the **NDIV** data divisions during data input, stress formulation.
2) **M**	INT	Controls DO loop for each tensile test point in an S/N data set, strain formulation.
MAXDAT	INT	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.
MAXMM	INT	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.
MAXREG	INT	Maximum number of life regions allowed for the S/N curve. The maximum number of regions is 3.
MAXSET	INT	Maximum number of related S/N data sets allowed. The maximum number of related data sets is 5.
MAXTNS	INT	Maximum number of tensile test points per data set allowed for the strain formulation. The maximum number of tensile points is 5.
MC(2, MAXREG)	RE	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**.
MCE(2, MAXREG)	RE	2-D array containing values of the range on the materials shape parameter m_E 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.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
1) **MCHAT(2, MAXREG)**	RE	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 $\hat{C}$.
2) **MCHAT(2, MAXREG)**	RE	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)**.
MCHATE(2, MAXREG)	RE	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 $\hat{C}_E$.
MCHATP(2, MAXREG)	RE	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 $\hat{C}_P$.
MCP(2, MAXREG)	RE	2-D array containing values of the range on the materials shape parameter m_P for the plastic 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.
MCPNT(MAXREG)	INT	1-D array containing the number of points, 0, 1, or 2, in **MC( )** for each life region.
MCPNTE(MAXREG)	INT	1-D array containing the number of points, 0, 1, or 2, in **MCE( )**. All entries are zero for the strain formulation.
MCPNTP(MAXREG)	INT	1-D array containing the number of points, 0, 1, or 2, in **MCP( )**. All entries are zero for the strain formulation.
1) **ME(0:MAXREG)**	RE	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.
2) **ME**	RE	m_E in *Equation 2-47*, the median m_E for the user-provided elastic strain components.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
3) ME	RE	Scalar variable used by function GTLIF2, equal to **ME(1)**.
MEANX(0:MAXSET)	RE	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 $X = \ln S$.
MEANY(0:MAXSET)	RE	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 $Y = \ln N$.
MEANZ	RE	$\bar{Z}$ in *Equation 2-42*, the sample mean of the transformed data of *Equations 2-39* through *2-41*.
MEBND(2, MAXREG)	RE	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.
MEDM(MAXMM)	RE	1-D array containing the median m for each life region of the S/N curve.[11]
MEDME(MAXREG)	RE	1-D array containing the median m_E for the elastic strain component of the S/N curve.[11]
MEDMP(MAXREG)	RE	1-D array containing the median m_P for the plastic strain component of the S/N curve.[11]
MID	INT	Pointer to the median m values in array **SORTM( )** for the truncated Normal median S/N curve. Value of half of **NTRIAL**.
MM(0:MAXREG)	RE	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.
MML	RE	Value of **MM(L)** for a set of calculations.
MO(MAXREG)	RE	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.[3]
MOE(MAXREG)	RE	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.[3]
MOP(MAXREG)	RE	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.[3]

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
1) **MP(0:MAXREG)**	RE	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.
2) **MP**	RE	m_P in *Equation 2-47*, the median m_P for the user-provided plastic strain components.
3) **MP**	RE	Scalar variable used by function GTLIF2, equal to **MP(1)**.
MPNT(MAXREG)	RE	1-D array containing the number of points, 0, 1, or 2, in **MZERO( )** for each life region.
MPNTE(MAXREG)	RE	1-D array containing the number of points, 0, 1, or 2, in **MZEROE( )** for the elastic strain component.
MPNTP(MAXREG)	RE	1-D array containing the number of points, 0, 1, or 2, in **MZEROP( )** for the plastic strain component.
MPROC	INT	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.[12]
MTERM	RE	Value of − [(1 / **MP**) − (1 / **ME**)].[13]
MU(MAXREG)	RE	1-D array containing the posterior Normal distribution mean[14] of the materials shape parameter *m* for each life region of the truncated Normal S/N curve.
MUE(MAXREG)	RE	1-D array containing the posterior Normal distribution mean[15] of the materials shape parameter m_E for the elastic strain components of the truncated Normal S/N curve.
MUP(MAXREG)	RE	1-D array containing the posterior Normal distribution mean[16] of the materials shape parameter m_P for the plastic strain components of the truncated Normal S/N curve.
MZERO(2, MAXREG)	RE	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.
MZEROE(2, MAXREG)	RE	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.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
MZEROP(2, MAXREG)	RE	2-D array containing values of the user-specified explicit constraints on the shape parameter m_p for the plastic strain components. **MZEROP(1,1)** is the lower bound and **MZEROP(2,1)** is the upper bound.
NBND(0:MAXREG)	RE	$N^*_{i,j+1}$ 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.
NCOMPS	INT	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.
NDC(0:MAXSET)	INT	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.
NDIV	INT	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.
NF(MAXDAT, MAXREG)	RE	2-D array containing values from the array **RAWNF( )** for the specific material S/N data set partitioned into life regions.
NMED	INT	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.
NNODAT	INT	Number of NO DATa regions, the number of life regions to the right with no S/N data.[17]
1) **NP(0:MAXSET, MAXREG)**	INT	2-D array containing number of points of each S/N data set in each life region.
2) **NP**	INT	The total number of points in the specific material S/N data set.
NPPR(MAXREG)	INT	Number of Points Per Region. 1-D array containing values of N in *Equation 2-20* for each life region.
1) **NPTS(MAXREG)**	INT	1-D array containing the number of points per life region for the specific material S/N data set.
2) **NPTS(0:MAXSET)**	INT	1-D array containing the total number of points in each S/N data set.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
NSETS	INT	*P* in *Equation 2-20*, the number of related material S/N data sets.
NTENS(0:MAXSET)	INT	1-D array containing the number of tensile test data points for each S/N data set for the strain formulation.
NTRIAL	INT	Number of *m*'s to be used to calculate the empirical median S/N curve for the truncated Normal case.[4]
1) **NUM**	INT	Number of S/N data points in a particular data division during data input.
2) **NUM**	INT	Value of **NPPR(L)** for a set of calculations.
3) **NUM(MAXREG)**	INT	1-D array containing values of $[(\Sigma N_j) - 1]$ for each life region.
4) **NUM(0:MAXSET)**	INT	1-D array containing the number of points in each S/N data set.
5) **NUM**	INT	Number of elements in array **ALLM( )** to be sorted.
NUM1	INT	Index of first point in **STR( )** to have the stress ratio transformation performed.
NUM2	INT	Index of last point in **STR( )** to have the stress ratio transformation performed.
NUMREG	INT	*R* in *Equation 2-11*, the number of life regions of interest in the S/N curve.
NZERO	RE	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.
PHI	RE	φ in *Equation 2-11*, the materials intrinsic variation, or scatter, given by a Weibull(β_o, η_o (β_o)) random variate.
PI	RE	π, constant equal to 3.1415926536.
PICK(2)	RE	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.
PLAS(MAXDAT, 0:MAXSET)	RE	2-D array containing values of the plastic strain components S_P (%).
PSIG	RE	σ in *Equation 2-48*, the value of SQRT(**PVAR**).
PSIG2(MAXREG)	RE	1-D array containing the estimates of σ^2, *Equation 2-49*, for each life region.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
PVAR	RE	σ^2 in *Equation 2-48*, characterizes the extent of departure from the multiple heat median S/N curve warranted by the available information.
R	RE	S/N data stress ratio, a stress ratio of −1.0 is desired.
RAND	DRE	Random number seed.
RANGEM(2, MAXREG)	RE	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.[18]
RANGME(2, MAXREG)	RE	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.[19]
RANGMP(2, MAXREG)	RE	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.[20]
RATIO	RE	S/N data stress ratio, a stress ratio of −1.0 is desired.
RATSTR(MAXDAT, 0:MAXSET)	RE	2-D array containing stress points (psi) with stress ratio = −1.0, or strain total points (%) for all S/N data sets.
RAWNF(MAXDAT, 0:MAXSET)	RE	2-D array containing the cycles to failure data for all S/N data sets.
RAWSTR(MAXDAT, 0:MAXSET)	RE	2-D array containing the user-provided stress data (psi) or total strain data (%) for all S/N data sets.
REFNP(MAXREG)	INT	1-D array containing the number of points per life region for the specific material S/N data set.
REG	INT	Life region of interest for a particular data division during data input for the stress formulation.
RFNP(MAXREG)	INT	1-D array containing the number of points for the specific material S/N data set, **RFNF(1)** = **NP(0,1)**. Note: for the strain formulation case only region 1 is used.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
RFSTRE(MAXDAT, MAXREG)	RE	2-D array containing elastic strain component points (%), **SE( )** or **STRE( )**, for the specific material S/N data set, strain formulation.
RFSTRP(MAXDAT, MAXREG)	RE	2-D array containing plastic strain component points (%), **SP( )** or **STRP( )**, for the specific materials S/N data set, strain formulation.
RSTR(MAXDAT, 0:MAXSET)	RE	2-D array containing stress points (psi) with stress ratio = −1.0, or strain total points (%) for all S/N data sets.
S	RE	Value of stress S (psi) for which a value of fatigue cycles to failure is required.
SBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = − 1.0, corresponding to the "life boundary" values for each life region of the S/N curve contained in array **NBND( )**.
SCLK	RE	Adjustment factor for **BIGK( )** if **TRSBND(0)** > **SZERO**.[21]
SE(MAXDAT, 0:MAXSET)	RE	2-D array containing values of the user-provided elastic strain components S_E (%).
SIG(MAXREG)	RE	1-D array containing the posterior Normal distribution standard deviation[22] of the materials model shape parameter m for each life region of the truncated Normal S/N curve.
SIG2E(MAXREG)	RE	1-D array containing value of $V(\ln N \mid \ln S_E)$, the shape parameter σ_E^2 of the original distribution of the shape parameter m_E for the elastic strain components.[3]
SIG2P(MAXREG)	RE	1-D array containing value of $V(\ln N \mid \ln S_P)$, the shape parameter σ_P^2 of the original distribution of the shape parameter m_p for the plastic strain components.[3]
SIGE(MAXREG)	RE	1-D array containing the posterior Normal distribution standard deviation[23] of the materials model shape parameter m_E for the elastic strain components.
SIGMA2(MAXREG)	RE	1-D array containing values of $V(\ln N \mid \ln S)$, the shape parameter σ^2 of the original distribution of the shape parameter m for each life region.[3]

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
SIGP(MAXREG)	RE	1-D array containing the posterior Normal distribution standard deviation[24] of the materials model shape parameter m_p for the plastic strain components.
SP(MAXDAT, 0:MAXSET)	RE	2-D array containing values of the user-provided plastic strain components S_p (%).
1) STR(MAXDAT, MAXREG)	RE	2-D array containing stress points with stress ratio $= -1.0$, for the specific material S/N data set partitioned into life regions, stress formulation.
2) STR(MAXDAT, 0:MAXSET)	RE	2-D array containing the user-provided stress data (psi) or total strain data (%) for all S/N data sets.
3) STR	RE	Value of total strain S (%) for which a value of fatigue cycles to failure is required.
STRAIN	RE	Value of total strain S (%) for which fatigue cycles to failure is desired.
STRE(MAXDAT, 0:MAXSET)	RE	2-D array containing values of the user-provided and the calculated elastic strain components S_E (%).
STRESS	RE	Value of stress (psi) for which fatigue cycles to failure is desired.
STRP(MAXDAT, 0:MAXSET)	RE	2-D array containing values of the user-provided and the calculated plastic strain components S_p (%).
SUHAT	RE	Value of **SUHAT2(L)** ** 0.5 for a set of calculations.
SUHAT2(MAXREG)	RE	1-D array containing values of the residual variances $S_{\hat{u}}^2$, *Equation 2-21*, from the X on Y regression for each life region, where $X = \ln S$, $Y = \ln N$.
SUM	RE	Value of Σ(**PSIG2(L)** * **NUM(L)**), a weighted sum of the **PSIG2**'s, used to calculate a weighted average to obtain σ^2 in *Equation 2-49*.[25]
SUMX2	RE	Value of (**NPPR( )** * **SX2( )**) for a particular life region.
SWHAT	RE	Value of **SWHAT2(L)** ** 0.5 for a set of calculations.
SWHAT2(MAXREG)	RE	1-D array containing values of the residual variances $S_{\hat{w}}^2$, *Equation 2-20*, from the Y on X regression for each life region, where $X = \ln S$, $Y = \ln N$.
SX	RE	Value of (**NPPR(L)** * **SX2(L)**) ** 0.5 for a set of calculations.

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
SX2(MAXREG)	RE	1-D array containing values of the sample variance S_X^2, *Equation 2-20*, for each life region, where $X = \ln S$.
SXY(MAXREG)	RE	1-D array containing values of the sample covariance S_{XY}, *Equation 2-20*, for each life region, where $X = \ln S$, $Y = \ln N$.
SY2(MAXREG)	RE	1-D array containing values of the sample variance S_Y^2, *Equation 2-21*, for each life region, where $Y = \ln N$.
SZ	RE	Value of **SZ2** ** 0.5.
SZ2	RE	$S_Z{}^2$ in *Equation 2-42*, the sample variance of the transformed data of *Equations 2-39* through *2-41*.
SZERO	RE	Stress tensile test point, S_o (psi).[21]
T	RE	Value of **T025( )** to be used in calculations for *Equation 2-26*.
T025(TTAB)	RE	Table of 0.025 percentage points for the Student's *t*-distribution.
1) **TEMP**	RE	Temporary variable used to prevent arithmetic underflows and overflows during fatigue life calculation.
2) **TEMP**	RE	Temporary variable used during insertion sort.
TEST	RE	Value of $K_t S_{max} (1 - R) / 2$, 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.
TESTM	RE	Upper bound of the posterior credibility range on *m* in region (L − 1), used during concavity adjustment calculations.[26]
TNSILE(0:MAXSET, MAXTNS)	RE	2-D array containing tensile test data points for each S/N data set, strain formulation.
TOTAL	INT	Sum of **NUM( )** over all life regions.[25]
TRBIGK(0:MAXREG)	RE	1-D array containing values of the materials model location parameter *K* consistent with the tensile point S_o.[21]
TRSBND(0:MAXREG)	RE	1-D array containing the stress values (psi) with stress ratio = − 1.0, corresponding to the "life boundary" values for each region of the S/N curve contained in array **NBND( )** for each **PHI** draw consistent with the tensile point S_o.21

Table 7-8 List of Variables for Program MATCHR (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
TTAB	INT	Maximum number of degrees of freedom in **T025( )**. The maximum number is 31.
UPPER	RE	The upper bound of the intersection to find $\pi(m)$ in a particular region.[9]
VARY	INT	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.
1) X	RE	Uniform(0, 1) random variate used to obtain a randomly selected value for *m*.
2) X	RE	Normal(μ, σ^2) random variate used to obtain a randomly selected value for *m*.
3) X	RE	Value of the root of the equation $f(x) = 0$ for a particular iteration of the Newton's method calculations.
XST	RE	Value of the initial guess of the root *x* for the Newton's method calculations.
Z	RE	*Z* in *Equation 2-48*, the randomly selected process variation shift factor given by a Lognormal(0, **PVAR**) random variate.
ZROREG	INT	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]
ZZ(MAXDAT)	RE	*Z* in the data transformation of *Equations 2-39* through *2-41*.

1 The need for saving *m*'s is discussed on *Page 2-15*.

2 See variable **BZERO** for a description of β_o.

3 Specification of the Bayesian prior distribution for the truncated Normal case is discussed on *Page 2-14*.

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 λ_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 σ_* of the posterior density of m is discussed on *Page 2-14*.

23 σ_{*E} of the posterior density of m_E is discussed on *Page 2-14*.

24 σ_{*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 7-9 Routine/Variable Chart

	ALLM	ALPHA	ARG	ARG1	ARG2	BB	BIGK	BIGK1	BIGKE	BIGKHT	BIGKP	BZERO	BZEROE	BZEROP	BZHAT	CHI025	CHI975	CHITAB	COUNT	CZERO	CZERO2	DD	DELTA	DELTAE	DELTAP	DESCRP	DERF	DIFFX	DIFFY	ELAS	EPS
ADDREG																															
ADDRGN																															
ADJSTM																															
CONCAV																															
CONVRT																															
DECOMP												X	X	X								X		X	X	X					
EXPCTD							X	X							X																
FCT																											X				
FINDK							X					X																			
FINDM																															
FINDMC				X	X															X	X										
FINDMN																															
FINDSB							X																								
FNDRNG																															
GTLIFE							X																								
GTLIFEJ																															
GTLIF2																											X				X
GTPVAR																															
INFAGG										X		X								X		X	X								
INIT																							X								
INITD																								X	X						
INTRVL			X													X	X	X				X									
KBETA												X																			
KOMO							X																								
MATCHR	X						X	X	X		X	X	X	X																	
MEDIAN																															
MUSIG			X																			X	X								
NEWTON																											X				X
NORRNG																															
PAREST							X					X																			
PECOMP		X																								X					
PREP																														X	
RCE																			X	X			X			X					
RDECHO																								X	X	X					
SMNVAR																															
SORTM	X																														
SW2SU2						X																X						X	X		
TRNSFM																															

Table 7-9 Routine/Variable Chart (Cont'd)

	F	FACTR	FTU	FTUZ	FTY	FTYZ	GAMMA	GETLIF	GTLIFE	GTLIF2	I	IEND	IER	II	INC	INORDR	INVME	INVMP	IOUT	IZERO	J	JZERO	K	KE	KEPROD	KHAT	KHATE	KHATP	KP	KPPROD	KRATIO
ADDREG																			X												
ADDRGN																			X												
ADJSTM																			X												
CONCAV																			X												
CONVRT			X		X						1								X		2										
DECOMP																			X	X	1	X					X	X			X
EXPCTD		X																	X							X					
FCT	X																X	X	X					2	X				2	X	X
FINDK							X												X				2								
FINDM																			X												
FINDMC																			X												
FINDMN																			X												
FINDSB																			X												
FNDRNG																			X			X									
GTLIFE								1	X										X												X
GTLIFEJ								1	X										X												X
GTLIF2	X							2		X		X	X						X					2					2		X
GTPVAR																			X		1										
INFAGG				X		X													X	X		X									X
INIT											1								X		1		1								
INITD											1								X		1		1								
INTRVL											2								X	X		X									
KBETA																			X				2								
KOMO		X																	X												
MATCHR		X		X		X			X	X				X					X												X
MEDIAN																			X												
MUSIG																			X												
NEWTON	X											X	X						X					2					2		X
NORRNG																			X												
PAREST																			X				2								
PECOMP											1								X		1			1					1		
PREP																			X		1		1								
RCE			X	X	X	X					1								X		1		1								X
RDECHO											1								X		1										X
SMNVAR											1								X												
SORTM											3				X	X			X												
SW2SU2																			X		1		1								
TRNSFM											1								X				1								

Table 7-9 Routine/Variable Chart (Cont'd)

	KTERM	L	LAMBDA	LAMN	LIFE	LL	LNA	LNNF	LNSTR	LNSTRE	LNSTRP	LNZ	LOWER	LOWERM	LPHIM	M	MAXDAT	MAXMM	MAXREG	MAXSET	MAXTNS	MC	MCE	MCHAT	MCHATE	MCHATP	MCP	MCPNT	MCPNTE	MCPNTP	ME
ADDREG		X				1													X					1							
ADDRGN		X				1													X					1							
ADJSTM		X																	X												
CONCAV		X																	X												
CONVRT																	X			X											
DECOMP				X				X		X	X						X		X	X	X		X	2	X	X	X		X	X	
EXPCTD		X															X		X												
FCT					2																										3
FINDK		X																	X												
FINDM		X																	X												
FINDMC		X																	X			X						X			
FINDMN		X																	X												
FINDSB		X																	X												
FNDRNG		X											X						X			X		1				X			
GTLIFE		X										X							X												
GTLIFEJ		X					X					X			X				X												
GTLIF2																															3
GTPVAR		X		X															X	X				1							
INFAGG		X		X				X	X								X		X	X		X		1				X			
INIT		X						X	X								X		X	X											
INITD		X						X		X	X					2	X		X	X	X		X	2			X		X	X	
INTRVL		X																	X					1							
KBETA																															
KOMO		X																	X												
MATCHR		X			1							X					X	X	X												1
MEDIAN		X												X					X												
MUSIG		X																	X					1							
NEWTON																															3
NORRNG		X											X						X			X		1				X			
PAREST		X															X		X												
PECOMP	X		X													2	X			X	X										2
PREP								X		X	X						X		X	X											
RCE		X		X				X	X							1	X		X	X											
RDECHO				X												2	X		X	X	X										
SMNVAR																	X														
SORTM		X																X	X												
SW2SU2		X						X	X								X		X	X											
TRNSFM		X				2											X		X												

Table 7-9 Routine/Variable Chart (Cont'd)

	MEANX	MEANY	MEANZ	MEBND	MEDM	MEDME	MEDMP	MID	MM	MML	MO	MOE	MOP	MP	MPNT	MPNTE	MPNTP	MPROC	MTERM	MU	MUE	MUP	MZERO	MZEROE	MZEROP	NBND	NCOMPS	NDC	NDIV	NF	NMED
ADDREG															X								X								
ADDRGN											X				X					X			X								
ADJSTM				X										1																	
CONCAV																															
CONVRT																															
DECOMP						X	X					X	X			X	X	X			X	X		X	X	X		X		X	
EXPCTD			X		X				X																	X	X			X	
FCT														3																	
FINDK									X																	X					
FINDM									X																						
FINDMC																															
FINDMN									X											X											
FINDSB									X																	X					
FNDRNG															X								X								
GTLIFE									X																						
GTLIFEJ									X																						
GTLIF2														3																	
GTPVAR																															
INFAGG					X						X				X			X		X			X			X				X	
INIT											X				X								X							X	
INITD												X	X			X	X							X	X	X		X		X	
INTRVL																															
KBETA			X																												
KOMO									X																	X					
MATCHR				X	X			X	X					1				X		X	X	X				X	X			X	X
MEDIAN					X																										
MUSIG											X									X											
NEWTON														3																	
NORRNG															X								X								
PAREST			X						X											X						X				X	
PECOMP														2					X									X			
PREP																														X	
RCE											X				X			X					X			X			X	X	
RDECHO												X	X			X	X	X						X	X			X			
SMNVAR			X																												
SORTM																															
SW2SU2	X	X																													
TRNSFM									X	X																X				X	

Table 7-9 Routine/Variable Chart (Cont'd)

	NNODAT	NP	NPPR	NPTS	NSETS	NTENS	NTRIAL	NUM	NUM1	NUM2	NUMREG	NZERO	PHI	PI	PICK	PLAS	PSIG	PSIG2	PVAR	R	RAND	RANGEM	RANGME	RANGMP	RATIO	RATSTR	RAWNF	RAWSTR	REFNP	REG	RFNP
ADDREG	X										X											X									
ADDRGN	X										X											X									
ADJSTM																							X								
CONCAV											X											X									
CONVRT									X	X										X											
DECOMP		1	X	2	X	X													X				X	X			X	X			X
EXPCTD		2		1							X																				
FCT													X																		
FINDK											X																				
FINDM											X				X						X	X									
FINDMC											X																				
FINDMN											X				X						X	X									
FINDSB											X																				
FNDRNG											X											X									
GTLIFE											X		X																		
GTLIFEJ											X																				
GTLIF2												X	X																		
GTPVAR		1			X			3			X							X	X												
INFAGG	X	1	X	2	X						X								X			X				X	X	X	X		
INIT		1		2																						X	X	X	X		
INITD		1		2		X																					X	X			X
INTRVL			X					2			X																				
KBETA														X																	
KOMO											X																				
MATCHR				1			X				X		X				X		X		X	X	X	X							
MEDIAN											X											X									
MUSIG			X								X																				
NEWTON													X																		
NORRNG											X											X									
PAREST		2		1							X										X	X									
PECOMP				2	X	X																					X	X			
PREP		1			X			4								X											X				X
RCE	X	1		2	X			1			X														X	X	X	X	X	X	
RDECHO				2	X	X																					X	X			
SMNVAR		2																													
SORTM								5			X																				
SW2SU2		1	X		X						X																				
TRNSFM		2		1							X																				

Table 7-9 Routine/Variable Chart (Cont'd)

	RFSTRE	RFSTRP	RSTR	S	SBND	SCLK	SE	SIG	SIG2E	SIG2P	SIGE	SIGMA2	SIGP	SP	STR	STRAIN	STRE	STRESS	STRP	SUHAT	SUHAT2	SUM	SUMX2	SWHAT	SWHAT2	SX	SX2	SXY	SY2	SZ	SZ2
ADDREG																															
ADDRGN								X				X																			
ADJSTM																															
CONCAV																															
CONVRT			X												2																
DECOMP	X	X					X		X	X	X		X	X			X		X		X				X		X	X	X		
EXPCTD					X										1																X
FCT															3																
FINDK																															
FINDM																															
FINDMC																											X	X	X		
FINDMN								X																							
FINDSB					X																										
FNDRNG																															
GTLIFE				X	X																										
GTLIFEJ				X	X																										
GTLIF2															3																
GTPVAR																						X									
INFAGG								X				X			1						X				X		X	X	X		
INIT												X			1																
INITD	X	X					X		X	X				X			X		X												
INTRVL																				X	X			X	X	X	X				
KBETA																														X	X
KOMO						X																									
MATCHR	X	X			X			X			X		X		1	X		X													
MEDIAN																															
MUSIG								X				X									X		X		X		X				
NEWTON															3																
NORRNG																															
PAREST					X			X							1																X
PECOMP							X							X			X		X												
PREP	X	X																													
RCE												X			1																
RDECHO							X		X	X				X																	
SMNVAR																															X
SORTM																															
SW2SU2																					X				X		X	X	X		
TRNSFM															1																

Table 7-9 Routine/Variable Chart (Cont'd)

	SZERO	T	T025	TEMP	TEST	TESTM	TNSILE	TOTAL	TRBIGK	TRSBND	TTAB	UPPER	VARY	X	XST	Z	ZROREG	ZZ
ADDREG																		
ADDRGN																		
ADJSTM																		
CONCAV						X												
CONVRT					X													
DECOMP							X						X					
EXPCTD	X								X								X	X
FCT																X		
FINDK																		
FINDM														1				
FINDMC																		
FINDMN														2				
FINDSB																	X	
FNDRNG												X						
GTLIFE	X			1													X	
GTLIFEJ	X			1													X	
GTLIF2																X		
GTPVAR								X										
INFAGG	X												X				X	
INIT																		
INITD							X											
INTRVL		X	X								X							
KBETA																		
KOMO	X								X	X								
MATCHR	X								X	X			X			X	X	
MEDIAN																		
MUSIG																		
NEWTON														3	X	X		
NORRNG												X						
PAREST													X				X	X
PECOMP							X											
PREP																		
RCE	X												X				X	
RDECHO							X						X					
SMNVAR																		X
SORTM				2														
SW2SU2																		
TRNSFM																		X

7.3.1.4 Program MATCHR Listing

MATCHR Version 8.5

Program MATCHR Listing Temporal Order, Stress Formulation, Uniform Distribution

Program MATCHR Listing Temporal Order, Stress Formulation, Truncated Normal Distribution

Program MATCHR Listing Temporal Order, Strain Formulation, Uniform Distribution

Program MATCHR Listing Temporal Order, Strain Formulation, Truncated Normal Distribution

```
C  PROGRAM MATCHR PERFORMS THE OVERALL CONTROL OF THE STAND-ALONE
C  MATERIALS CHARACTERIZATION CALCULATIONS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  1AUG91
C     VERSION:  8.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      PROGRAM MATCHR

C  SUBPROGRAMS:  TRMNAT, INFAGG, PAREST, NORMGN, WEIBGN, KOMO, GTLIFE,
C                DECOMP, ADJSTM, GTLIF2, SORTM, EXPCTD
C        FILES:  1:SPECFD-OLD; 3:SPECFO-NEW; 5:RELATD-OLD; 6:RELATO-NEW;
C                7:DUMP-NEW; 8:IOUTPR-NEW
C                NOTE: 5 & 6 ARE OPENED IN INFAGG OR DECOMP

C     IMPLICIT NONE

      INTEGER  MAXDAT, MAXMM, MAXREG

      PARAMETER (MAXDAT = 50, MAXMM = 20001, MAXREG = 3)

      COMMON   IOUT

      INTEGER  II, IOUT, L, MID, MPROC, NCOMPS, NMED, NPTS(MAXREG),
     &         NTRIAL, NUMREG, VARY, ZROREG

      REAL     ALLM(MAXMM, MAXREG), BIGK(0:MAXREG), BIGK1,
     &         BIGKE(0:MAXREG), BIGKP(0:MAXREG), BZERO, BZEROE, BZEROP,
     &         FACTR, FTUZ, FTYZ, GTLIF2, GTLIFE, KRATIO, LIFE, LNZ,
     &         ME(0:MAXREG), MEBND(2, MAXREG), MEDM(MAXREG),
     &         MM(0:MAXREG), MP(0:MAXREG), MU(MAXREG), MUE(MAXREG),
     &         MUP(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG), PHI,
     &         PSIG, PVAR, RANGEM(2, MAXREG), RANGME(2, MAXREG),
     &         RANGMP(2, MAXREG), RFSTRE(MAXDAT, MAXREG),
     &         RFSTRP(MAXDAT, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
     &         SIGE(MAXREG), SIGP(MAXREG), STR(MAXDAT, MAXREG), STRAIN,
     &         STRESS, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG), Z

      DOUBLE PRECISION  RAND


C                       LIST OF VARIABLES
C
C  ALLM()        2-D ARRAY CONTAINING M VALUES TO BE SORTED FOR EACH REGION
C  BIGK()        1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C                  EACH REGION
C  BIGK1         EQUAL TO BIGK(1) -- DUMMY PARAMETER FOR CALLS TO SUBROUTINE
C                  EXPCTD
C  BIGKE()       1-D ARRAY CONTAINING VALUE OF Ke FOR ELASTIC COMPONENTS
C  BIGKP()       1-D ARRAY CONTAINING VALUE OF Kp FOR PLASTIC COMPONENTS
C  BZERO         VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA
C                  SET
C  BZEROE        VALUE OF WEIBULL PARAMETER, BETAoe, CHARACTERIZING ELASTIC
C                  COMPONENTS OF S/N DATA SET
C  BZEROP        VALUE OF WEIBULL PARAMETER, BETAop, CHARACTERIZING PLASTIC
C                  COMPONENTS OF S/N DATA SET
C  FACTR         IT IS FACTOR = PHI * KRATIO * Z
C  FTUZ          ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ          YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  GTLIF2        FUNCTION WHICH CALCULATES THE CYCLES TO FAILURE AT A
C                  GIVEN STRAIN
C  GTLIFE        FUNCTION WHICH CALCULATES THE CYCLES TO FAILURE AT A
C                  GIVEN STRESS OR STRAIN WHEN NCOMPS = 1
C  II            CONTROLS DO LOOP FOR EACH TRIAL
C  IOUT          OUTPUT DUMP CONTROLLER
C  KRATIO        RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L             CONTROLS DO LOOP FOR EACH REGION
C  LIFE          CYCLES TO FAILURE AT A GIVEN STRESS OR STRAIN
C  LNZ           NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  MAXDAT        MAXIMUM NUMBER OF POINTS PER DATA SET (PER REGION) ALLOWED
```

```
C  MAXMM       MAXIMUM NUMBER OF M'S TO BE SORTED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  ME()        1-D ARRAY CONTAINING SELECTED VALUE OF Me FOR ELASTIC
C                COMPONENTS
C  MEBND()     2-D ARRAY CONTAINING ALLOWABLE RANGE ON Me, FOR ELASTIC
C                COMPONENTS, AFTER SELECTION OF Mp
C  MEDM()      1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C  MID         POINTER TO THE MEDIAN M VALUES -- EQUAL TO HALF OF NTRIAL
C  MM()        1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MP()        1-D ARRAY CONTAINING SELECTED VALUE OF Mp FOR PLASTIC
C                COMPONENTS
C  MPROC       Materials PROCess variation -- CONTROLS MATERIALS PROCESS
C                VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MU()        1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION MEAN FOR EACH REGION
C  MUE()       1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION MEAN
C                FOR ELASTIC COMPONENTS
C  MUP()       1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION MEAN
C                FOR PLASTIC COMPONENTS
C  NBND()      1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                REGIONS OF INTEREST FOR THE SPECIFIC (REFERENCE) MATERIAL
C                S/N DATA SET
C  NCOMPS      Number of COMPonents -- 1 FOR STRESS AND STRAIN WHEN
C                DECOMPOSED DATA UNAVAILABLE -- 2 FOR DECOMPOSED STRAIN
C                DATA
C  NF()        2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                SPECIFIC MATERIAL S-N DATA SET BROKEN INTO REGIONS
C  NMED        CONTROLS MEDIAN CALCULATION FOR THE NORMAL DISTRIBUTION
C                CASE -- 0 - NO MEDIAN CALCULATION;  1 - MEDIAN CALCULATION
C                DESIRED
C  NPTS()      1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION
C                FOR THE SPECIFIC (REFERENCE) MATERIAL S/N DATA SET
C  NTRIAL      NUMBER OF M'S TO BE USED TO CALCULATE MEDIAN
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  PHI         WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIATE
C  PSIG        EQUAL TO SQRT(PVAR)
C  PVAR        CHARACTERIZES THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT
C                MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  RAND        RANDOM NUMBER SEED
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON
C                M FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND
C                AND RANGEM(2,L) IS THE UPPER BOUND
C  RANGME()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON
C                Me FOR ELASTIC COMPONENTS -- RANGME(1,1) IS THE LOWER
C                BOUND AND RANGME(2,1) IS THE UPPER BOUND
C  RANGMP()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON
C                Mp FOR PLASTIC COMPONENTS -- RANGMP(1,1) IS THE LOWER
C                BOUND AND RANGMP(2,1) IS THE UPPER BOUND
C  RFSTRE()    2-D ARRAY CONTAINING ELASTIC STRAIN POINTS (%), SE() OR
C                STRE(), FOR THE SPECIFIC MATERIAL S/N DATA SET
C  RFSTRP()    2-D ARRAY CONTAINING PLASTIC STRAIN POINTS (%), SP() OR
C                STRP(), FOR THE SPECIFIC MATERIAL S/N DATA SET
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()
C  SIG()       1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                DISTRIBUTION STANDARD DEVIATION FOR EACH REGION
C  SIGE()      1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION
C                STANDARD DEVIATION FOR ELASTIC COMPONENTS
C  SIGP()      1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION
C                STANDARD DEVIATION FOR PLASTIC COMPONENTS
C  STR()       2-D ARRAY CONTAINING STRESS OR STRAIN POINTS FOR THE
C                SPECIFIC MATERIAL S/N DATA SET (STRESS RATIO = -1.0)
C                BROKEN INTO REGIONS (PSI OR %)
C  STRAIN      VALUE OF TOTAL STRAIN (%) FOR WHICH CYCLES TO FAILURE IS
C                DESIRED
C  STRESS      VALUE OF STRESS (PSI) FOR WHICH CYCLES TO FAILURE IS DESIRED
C  SZERO       STRESS TENSILE TEST POINT, So
C  TRBIGK()    1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO BRING
C                CURVE DOWN IF SBND(0) > So FOR EACH TRIAL
C  TRSBND()    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() ADJUSTED BY VARIATION PARAMETERS
C                FOR EACH TRIAL
C  VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
```

```
C                VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                VARIATION;  3 - TRUNCATED NORMAL VARIATION
C  Z           LOG-NORMAL(0,PVAR) RANDOM VARIATE
C  ZROREG      ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                BEGINNING VALUE -- 0 - ZERO REGION EXISTS, 1 - NO ZERO
C                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 ',
     &              '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

      NTRIAL = 1
      IF ((VARY .EQ. 3) .AND. (NMED .EQ. 1)) NTRIAL = 2000


      IF (NCOMPS .EQ. 1) THEN

         READ(1,*) STRESS
         WRITE(8,*) '                                VALUE ',
     &              'OF STRESS: ', STRESS

C        PERFORM CALCULATIONS ON STRESS OR TOTAL STRAIN DATA

         CALL INFAGG (RANGEM, MU, SIG, NF, NPTS, SZERO, ZROREG, NUMREG,
     &                NBND, STR, FTUZ, FTYZ, VARY, MPROC, KRATIO, PVAR)
```

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

      DO 100 II = 1, NTRIAL

         CALL PAREST (VARY, RANGEM, MU, SIG, NF, NPTS, NUMREG,
     &                ZROREG, RAND, NBND, 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

         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, MM, 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
         DO 50 L = ZROREG, NUMREG
            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(II,L) = MM(L)
 150     CONTINUE

 100  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,*) '                       VALUE ',
     &           'OF TOTAL STRAIN: ', STRAIN
```

```
C        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, BZEROP, 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 / BZEROP) ** 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)

 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,
     &                   BIGK1, BZEROP)

            WRITE(7,*) 'MEDIAN ELASTIC S/N CURVE'
            MEDM(1) = ALLM(MID,2)

            CALL EXPCTD (2, MEDM, NPTS, RFSTRE, NF, 0.0, 1, 1, NBND,
     &                   BIGK1, BZEROE)

            BZERO = 1.0 / SQRT (0.5 * ((1.0 / BZEROP) ** 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*************************************************************************


C  SUBROUTINE TRMNAT HANDLES THE TERMINATION OF THE PROGRAM RUN WHEN
C  ONE OF THE PROGRAM'S ASSUMPTIONS HAVE BEEN VIOLATED
C  PROGRAMMER:  L. NEWLIN
C        DATE:  5OCT87
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE TRMNAT

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


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



C  SUBROUTINE INFAGG CONTROLS THE CALCULATIONS FOR THE INFORMATION
C  AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C  FOR THE STRESS FORMULATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  13JUL89      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.4, V8.5    MATGRM V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE INFAGG (RANGEM, MU, SIG, NF, 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,
C              NBND, STR, FTUZ, FTYZ, KRATIO, PVAR
C  SUBPROGRAMS:  INIT, RCE, SW2SU2, FINDMC, INTRVL, FNDRNG, ADDREG,
C                CONCAV, MEDIAN, EXPCTD, MUSIG, NORRNG, ADDRGN, GTPVAR
C        FILES:  5:RELATD-OLD; 6:RELATO-NEW

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET

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

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), MPROC, NNODAT,
     &        NP(0:MAXSET, MAXREG), NPPR(MAXREG), NPTS(0:MAXSET),
     &        NSETS, NUMREG, REFNP(MAXREG), VARY, ZROREG

      REAL    BIGKHT, BZERO, CZERO, DD(MAXREG), DELTA(MAXREG),
```

```
     &          FTUZ, FTYZ, IZERO(2, MAXREG), JZERO(2, MAXREG),
     &          KRATIO, LAMN, LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &          LNSTR(MAXDAT, 0:MAXSET, MAXREG), MC(2, MAXREG),
     &          MCHAT(2, MAXREG), MEDM(MAXREG), MO(MAXREG), MU(MAXREG),
     &          MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &          PVAR, RANGEM(2, MAXREG), RATSTR(MAXDAT, 0:MAXSET),
     &          RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &          SIG(MAXREG), SIGMA2(MAXREG), STR(MAXDAT, MAXREG),
     &          SUHAT2(MAXREG), SWHAT2(MAXREG), SX2(MAXREG),
     &          SXY(MAXREG), SY2(MAXREG), SZERO


C                    LIST OF VARIABLES
C
C  BIGKHT       EQUAL TO THE MEDIAN VALUE OF K IN REGION 1
C  BZERO        VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING THE S/N
C                 DATA SET
C  CZERO        EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                 COEFFICIENT OF VARIATION, Co
C  DD()         1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()      1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU()
C                 AND SIG() CALCULATION
C  FTUZ         ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  FTYZ         YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C  IOUT         OUTPUT DUMP CONTROLLER
C  IZERO()      2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                 FOR EACH REGION
C  JZERO()      2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C  KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L            CONTROLS DO LOOP FOR EACH REGION
C  LAMN         LAMBDA-N -- RATIO OF Var(Ln N given S) / (m**2 C**2),
C                 CONSTANT OVER REGIONS AND COMPONENTS
C  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 IN 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  MC()         2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                 REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                 -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                 BOUND
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                 MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MC() FOR EACH REGION
C  MEDM()       1-D ARRAY CONTAINING THE MEDIAN M FOR EACH REGION
C  MO()         1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C  MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C  MPROC        Materials PROcess variation --CONTROLS MATERIALS PROCESS
C                 VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MU()         1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C                 DISTRIBUTION MEAN FOR EACH REGION
C  MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C  NBND()       1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C  NF()         2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                 SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NNODAT       Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C  NP()         2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                 SET IN EACH REGION
C  NPPR()       1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                 ALL DATA SETS IN A REGION (Number of Points Per Region)
C  NPTS()       1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  PVAR         MATERIALS PROCESS VARIATION
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
```

```
C                     FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                     RANGEM(2,L) IS THE UPPER BOUND
C    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) OR 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    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    STR()          2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C                     S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C    SUHAT2()       1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                     REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C    SWHAT2()       1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
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    SXY()          1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                     REGION (X = Ln S, Y = Ln N)
C    SY2()          1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                     (Y = Ln N)
C    SZERO          STRESS TENSILE TEST POINT, So
C    VARY           CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                     VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                     VARIATION;  3 - TRUNCATED NORMAL VARIATION
C    ZROREG         ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                   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
C  INFORMATION

C  PERFORM CALCULATIONS COMMON TO BOTH UNIFORM AND NORMAL TYPE OF VARIATION

C  INITIALIZE PRIMARY ARRAYS

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

C  READ, CONVERT, ECHO INFORMATION

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

C  CALCULATE RESIDUAL VARIANCES

      CALL SW2SU2 (NUMREG, NSETS, NP, LNSTR, LNNF, 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, MCHAT)
```

```
C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION TO
C  OBTAIN POSTERIOR RANGES ON M

         CALL FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT,
     &                RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

         CALL ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

         CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

         WRITE(7,900)

         DO 25 L = 1, NUMREG
            WRITE(7,905) L, IZERO(1, L), IZERO(2, L),
     &                      JZERO(1, L), JZERO(2, L)
   25    CONTINUE

         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,
     &                NBND, BIGKHT, BZERO)

C  CHECK TYPE OF S/N VARIATION DESIRED AND FIX M AT MEDIAN IF DESIRED

         IF ((VARY .EQ. 0) .OR. (VARY .EQ. 1)) THEN
            DO 200 L = 1, NUMREG
               RANGEM(1,L) = MEDM(L)
               RANGEM(2,L) = MEDM(L)
  200       CONTINUE
         ENDIF

      ELSE
```

```
C  NORMAL VARIATION IS DESIRED

C  CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION FOR EACH REGION

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

C  CALCULATE MATERIALS PROCESS VARIATION IF DESIRED

      IF (MPROC .EQ. 1) THEN
         CALL GTPVAR (NSETS, NP, NUMREG, LAMN, MCHAT, PVAR)
      ENDIF

C  COMBINE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGES ON M

         CALL NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  ADD INFORMATION ON RANGE FOR REGIONS WITHOUT DATA

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

C  ADJUST UPPER BOUNDS OF POSTERIOR RANGES FOR CONCAVITY CONSTRAINTS

         CALL CONCAV (NUMREG, RANGEM)

C  WRITE RESULTS TO FILE DUMP

         WRITE(7,975)

         DO 350 L = 1, NUMREG
            WRITE(7,980) L, MCHAT(1,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(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)
         DO 380 L = 1, NUMREG
            WRITE(7,990) L, MU(L), SIG(L)
  380 CONTINUE

      ENDIF

C  PRINT RESULTS OF MATERIALS PROCESS VARIATION CALCULATIONS

      IF (MPROC .EQ. 1) THEN
         WRITE(7,995) PVAR
      ENDIF

C  FORMAT STATEMENTS

  900 FORMAT(2X,'Copyright (C) 1990, California Institute of ',
     &       'Technology.  U.S. Government',/,2X,'Sponsorship under ',
     &       'NASA Contract NAS7-918 is acknowledged.',////,
     &       2X,'RESULTS OF INFORMATION AGGREGATION CALCULATIONS',
```

```
     &         ///,2X,'95% CONFIDENCE INTERVALS ON C AND m ',
     &         'FOR EACH REGION',/)

 905 FORMAT(7X,'REGION:  ',I1,7X,'Io = (',F12.9,',',F12.9,')',
     &         /,24X,'Jo = (',F12.9,',',F12.9,')')

 910 FORMAT(//,2X,'POINT ESTIMATES OF C AND m FOR EACH REGION',
     &         //,7X,'REGION',8X,'E(C)',12X,'E(m)',/)

 915 FORMAT(9X,I1,8X,F11.9,5X,F9.6)

 920 FORMAT(///,2X,'POSTERIOR CREDIBILITY RANGE ON m FOR EACH '
     &         'REGION')

 930 FORMAT(//,2X,'REGION',5X,'LOWER BOUND',5X,'UPPER BOUND',/)

 940 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

 950 FORMAT(///)

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

 965 FORMAT(6X,I1,8X,F8.4,8X,'INFINITY')

 970 FORMAT(6X,I1,8X,F8.4,8X,F8.4)

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

 980 FORMAT(9X,I1,11X,F10.6)

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

 990 FORMAT(5X,I1,5X,F7.4,5X,E11.5)

 995 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',
     &         'MEDIAN S/N CURVE',/,2X,'WARRANTED BY THE AVAILABLE ',
     &         'INFORMATION',//,7X,E11.5)

      RETURN
      END


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


C  SUBROUTINE INIT PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C  USED IN THE INFORMATION AGGREGATION SUBROUTINE INFAGG
C  PROGRAMMER:  L. NEWLIN
C       DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   ---
C  OUTPUTS:  NPTS, RAWNF, RAWSTR, RATSTR, NP, LNNF, LNSTR, REFNP,
C            NF, STR, MPNT, MZERO, DELTA, MO, SIGMA2

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG, MAXSET
```

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

      COMMON  IOUT

      INTEGER I, IOUT, J, K, L, MPNT(MAXREG), NP(0:MAXSET, MAXREG),
     &        NPTS(0:MAXSET), REFNP(MAXREG)

      REAL    DELTA(MAXREG), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &        LNSTR(MAXDAT, 0:MAXSET, MAXREG), MO(MAXREG),
     &        MZERO(2, MAXREG), NF(MAXDAT, MAXREG),
     &        RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG)
C
C                  LIST OF VARIABLES
C
C   DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                 SIG() CALCULATION
C   I           CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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 IN 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   MO()        1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                 MEAN FOR EACH REGION
C   MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZERO() FOR EACH REGION
C   MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                 EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                 IS THE UPPER BOUND
C   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
C                 S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C
      DO 100 J = 0, MAXSET
         NPTS(J) = 0.0
  100 CONTINUE

      DO 200 L = 1, MAXREG
         DO 250 J = 0, MAXSET
            NP(J, L) = 0.0
  250    CONTINUE
  200 CONTINUE

      DO 300 J = 0, MAXSET
         DO 350 I = 1, MAXDAT
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            RATSTR(I,J) = 0.0
  350    CONTINUE
  300 CONTINUE

      DO 400 L = 1, MAXREG
         DO 425 K = 1, MAXDAT
            DO 450 J = 0, MAXSET
               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"
C  THE STRESS DATA TO A STRESS RATIO OF -1.0; AND "ECHOES" THE DATA TO
C  SPECFO AND RELATO.  RCE ALSO BREAKS S/N DATA SETS INTO REGIONS AS
C  SPECIFIED BY USER
C  PROGRAMMER:  L. NEWLIN
C        DATE:  21JUN88       FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE RCE (VARY, MPROC, NPTS, RAWNF, RAWSTR, RATSTR, NP,
     &                LNSTR, LNNF, REFNP, STR, NF, SZERO, ZROREG,
     &                NUMREG, NNODAT, NSETS, NBND, CZERO, MPNT, MZERO,
     &                FTUZ, FTYZ, DELTA, MO, SIGMA2, 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,
C             CZERO, MPNT, MZERO, FTUZ, FTYZ, DELTA, MO, SIGMA2,
C             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),
     &        MZERO(2, MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        RATIO, RATSTR(MAXDAT, 0:MAXSET), RAWNF(MAXDAT, 0:MAXSET),
     &        RAWSTR(MAXDAT, 0:MAXSET), SIGMA2(MAXREG),
     &        STR(MAXDAT, MAXREG), SZERO

      CHARACTER*40  DESCRP(0:MAXSET)
```

```
C                         LIST OF VARIABLES
C
C   COUNT        INDEX THAT KEEPS TRACK OF DATA DURING INPUT, ECHO,
C                  CONVERSION, AND BREAK UP
C   CZERO        EXOGENOUS INFORMATION IN THE FORM OF A CONSTRAINT ON THE
C                  COEFFICIENT OF VARIATION, Co
C   DELTA()      1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
C                  SIG() CALCULATION
C   DESCRP()     1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C   FTU          ULTIMATE STRENGTH (PSI) OF MATERIAL DATA SET
C   FTUZ         ULTIMATE STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   FTY          YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   FTYZ         YIELD STRENGTH (PSI) FOR SPECIFIC MATERIAL
C   I            CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
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   KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C   L            CONTROLS DO LOOP FOR EACH REGION
C   LAMN         LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                  CONSTANT OVER ALL REGIONS AND COMPONENTS
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   M            CONTROLS DO LOOP FOR EACH DATA DIVISION
C   MAXDAT       MAXIMUM NUMBER OF POINTS IN 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   MO()         1-D ARRAY CONTAINING VALUES OF THE PRIOR NORMAL DISTRIBUTION
C                  MEAN FOR EACH REGION
C   MPNT()       1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                  MZERO() FOR EACH REGION
C   MPROC        Materials PROcess variation -- CONTROLS MATERIALS PROCESS
C                  VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C   MZERO()      2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                  EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                  IS THE UPPER BOUND
C   NBND()       1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                  REGIONS OF INTEREST
C   NDIV         NUMBER OF DIVISIONS DATA SET IS BROKEN INTO BY RATIO,
C                  REGION PAIRS DURING INPUT
C   NF()         2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C                  SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C   NNODAT       Number of NO DATa regions (REGIONS WITHOUT ANY S/N DATA)
C   NP()         2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA SET
C                  IN EACH REGION
C   NPTS()       1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUM          NUMBER OF DATA POINTS IN A PARTICULAR DIVISION
C   NUMREG       NUMBER OF REGIONS OF INTEREST
C   RATIO        STRESS RATIO (R = -1.0 IS DESIRED)
C   RATSTR()     2-D ARRAY CONTAINING STRESS DATA (PSI) CORRECTED FOR STRESS
C                  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) OR 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   REG          REGION OF INTEREST IN A PARTICULAR DIVISION
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
C                  S/N DATA SET BROKEN INTO REGIONS (PSI OR %)
C   SZERO        STRESS TENSILE TEST POINT, So
C   VARY         CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                  VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                  VARIATION;  3 - TRUNCATED NORMAL VARIATION
C   ZROREG       ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                  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), FTY, FTU, NDIV, NPTS(0)

      IF (NPTS(0) .GT. MAXDAT) THEN
         WRITE(8,*) 'ERROR:  OVER NUMBER OF POINTS LIMIT IN ',
     &              'SPECIFIC MATERIAL'
         CALL TRMNAT
      ENDIF

      WRITE(3,900) DESCRP(0), FTY, FTU, NPTS(0)
      IF (IOUT .EQ. 10) WRITE(8,900) DESCRP(0), FTY, FTU, NPTS(0)

      WRITE(3,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

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

      FTUZ = FTU
      FTYZ = FTY

C     INPUT STRESS/LIFE INFORMATION -- INCLUDING STRESS RATIO AND REGION
C     INFORMATION FROM SPECFD AND ECHO TO SPECFO

      DO 100 M = 1, NDIV

         READ (1,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE FOR RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

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

         DO 110 I = (COUNT + 1), (COUNT + NUM)
            READ(1,*) RAWSTR(I,0), RAWNF(I,0)
  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, FTY)

         ENDIF

C        ECHO STRESS/LIFE DATA ON SPECIFIC MATERIAL

         DO 130 I = (COUNT + 1), (COUNT + NUM)

            WRITE(3,910) RAWSTR(I,0), RAWNF(I,0), RATIO, REG,
     &                   RATSTR(I,0), RAWNF(I,0)
```

```
          IF (IOUT .EQ. 10) WRITE(8,910) RAWSTR(I,0), RAWNF(I,0),
     &                  RATIO, REG, RATSTR(I,0), RAWNF(I,0)

 130    CONTINUE

C       BREAK UP DATA ACCORDING TO SPECIFIED REGIONS FOR USE BY SW2SU2,
C       EXPCTD, AND PAREST

        K = NP(0,REG)

        DO 140 I = (COUNT + 1), (COUNT + NUM)

           K = K + 1
           LNSTR(K,0,REG) = ALOG(RATSTR(I,0))
           LNNF(K,0,REG) = ALOG(RAWNF(I,0))
           STR(K,REG) = RATSTR(I,0)
           NF(K,REG) = RAWNF(I,0)

 140    CONTINUE

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

        NP(0,REG) = K
        REFNP(REG) = K
        COUNT = COUNT + NUM

 100 CONTINUE

     IF (NPTS(0) .NE. COUNT) THEN
        WRITE(8,*) 'ERROR:  NUMBER OF POINTS PER DIVISION ',
     &             'INCORRECTLY SPECIFIED'
        WRITE(8,*) 'IN SPECIFIC DATA SET'
        CALL TRMNAT
     ENDIF

     READ(1,*) SZERO
     IF (NINT (SZERO) .GT. 0) THEN
        ZROREG = 0
     ELSE
        ZROREG = 1
     ENDIF
     IF (IOUT .EQ. 10)
     &  WRITE(8,*) 'SZERO = ', SZERO, ' ZROREG = ', ZROREG

C    INPUT OTHER REGION INFORMATION AND EXOGENOUS INFORMATION

     READ(1,*) NUMREG, NNODAT

     IF ((NUMREG + NNODAT) .GT. MAXREG)  THEN
        WRITE(8,*) 'ERROR:  EXCEEDED LIMIT ON NUMBER OF REGIONS'
        CALL TRMNAT
     ENDIF

     DO 150 L = ZROREG, (NUMREG + NNODAT)
        READ(1,*) NBND(L)
 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
         ENDIF
  180 CONTINUE

      IF (VARY .EQ. 3) THEN
C        READ PRIOR INFORMATION ON NORMAL DISTRIBUTION
         WRITE(3,945)
         IF (IOUT .EQ. 10) WRITE(8,945)
         DO 190 L = 1, (NUMREG + NNODAT)
            READ(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
            ENDIF
  190    CONTINUE
      ENDIF

      IF (MPROC .EQ. 1) THEN
         READ(1,*) KRATIO, LAMN
         WRITE(3,955) KRATIO, LAMN
         IF (IOUT .EQ. 10) WRITE(8,955) KRATIO, LAMN
      ENDIF


C  BEGIN INPUT OF RELATED MATERIAL INFORMATION FROM RELATD
C  AND THEN ECHO TO RELATO

      READ(5,*) NSETS

      IF (NSETS .GT. MAXSET) THEN
         WRITE(8,*) 'ERROR:  OVER LIMIT ON NUMBER OF RELATED DATA SETS'
         CALL TRMNAT
      ENDIF

      WRITE(6,935) NSETS

      DO 200 J = 1, NSETS

         COUNT = 0

         IF (IOUT. EQ. 10) WRITE(8,*)'J =', J, ' NSETS =', NSETS

         READ(5,*) DESCRP(J), FTU, FTY, NDIV, NPTS(J)

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

         WRITE(6,940) DESCRP(J), FTU, FTY, NPTS(J)
         IF (IOUT .EQ. 10) WRITE(8,940) DESCRP(J), FTU, FTY, NPTS(J)
```

```
      WRITE(6,905)
      IF (IOUT .EQ. 10) WRITE(8,905)

      DO 300 M = 1, NDIV

         READ(5,*) NUM, RATIO, REG

         IF (ABS(RATIO) .GT. 1.0) THEN
            WRITE(8,*) 'ERROR:  INVALID VALUE OF RATIO:  ', RATIO
            CALL TRMNAT
         ENDIF

         IF (REG .GT. MAXREG) THEN
            WRITE(8,*)
     &         'ERROR:  OVER REGION LIMIT IN RELATED MATERIAL ', J
            CALL TRMNAT
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'NUM = ', NUM, ' COUNT = ', COUNT
            WRITE(8,*) 'RATIO = ', RATIO, ' REG = ', REG
         ENDIF

         DO 310 I = (COUNT + 1), (COUNT + NUM)
            READ(5,*) RAWSTR(I,J), RAWNF(I,J)
  310    CONTINUE

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,I2)

  905 FORMAT(//,7X,'ORIGINAL S/N',9X,'STRESS',15X,'TRANSFORMED S/N',
     &       /,5X,'STRESS',7X,'LIFE',7X,'RATIO',3X,'REGION',5X,
     &       'STRESS',7X,'LIFE'/)

  910 FORMAT(2X,E11.5,2X,F9.0,5X,F5.2,5X,I1,5X,E11.5,2X,F9.0)

  913 FORMAT(//)

  914 FORMAT(2X,'THERE IS A NO DATA REGION TO THE LEFT WITH AN So OF',
     &       5X,E11.5)

  915 FORMAT(2X,'THERE IS ',I2,' REGION(S) WITH DATA  ',
     &       /,2X,'AND ',I2,' REGION(S) TO THE RIGHT WITHOUT DATA',
     &       /,2X,'THE UPPER BOUND(S) OF THE REGION(S) ARE ',
     &       '(CYCLES): ',/)

  920 FORMAT(10X,E9.3)

  925 FORMAT(///,2X,'EXOGENOUS INFORMATION',///,2X,
     &       'CONSTRAINT ON COEFFICIENT OF VARIATION, C:',2X,F6.4,
     &       //,2X,'EXPLICIT CONSTRAINT ON m FOR EACH REGION:',
     &       //,2X,'REGION',5X,'# OF POINTS',5X,'LOWER BOUND',
     &       5X,'UPPER BOUND',/)

  930 FORMAT(6X,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***********************************************************************


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

```
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:   6OCT87      COMMENTS:  13JUL89
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                       V8.3, V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
C
C   FTU         ULTIMATE STRENGTH OF MATERIAL (PSI)
C   FTY         YIELD STRENGTH OF MATERIAL (PSI)
C   I           CONTROLS DO LOOP FOR EACH POINT IN THE DATA SET
C   IOUT        OUTPUT DUMP CONTROLLER
C   J           DATA SET OF INTEREST
C   MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C   MAXSET      MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   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 * (1 - R)/2 , TO BE COMPARED WITH FTY

C   Kt IS ASSUMED TO BE ONE

      DO 100 I = NUM1, NUM2

         TEST = STR(I,J) * (1.0 - R)/2.0

         IF (IOUT.EQ.10) WRITE(8,*) 'I =',I,' J =',J,' TEST =',TEST

         IF (TEST .GE. FTY) THEN

            RSTR(I,J) = TEST

            IF (IOUT.EQ.10) WRITE(8,*)'1:RSTR() =',RSTR(I,J)

         ELSE IF ((TEST .LT. FTY) .AND. (STR(I,J) .GT. FTY)) THEN

            RSTR(I,J) = TEST/(1.0 -((FTY - TEST)/FTU))

            IF (IOUT.EQ.10) WRITE(8,*)'2:RSTR() =',RSTR(I,J)

         ELSE

            RSTR(I,J) = TEST/(1.0 - ((1.0 + R) * STR(I,J)
     &                  /(2.0 * FTU)))

            IF (IOUT.EQ.10) WRITE(8,*)'3:RSTR() =',RSTR(I,J)

         END IF

  100 CONTINUE

      RETURN
```

```
      END

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

C  SUBROUTINE SW2SU2 CALCULATES, SWHAT2, THE RESIDUAL VARIANCES OF Y ON X
C  AND, SUHAT2, THE X ON Y REGRESSIONS FOR EACH REGION WHERE Y = LN(NF) AND
C  X = LN(STR); TO BE USED IN THE CONFIDENCE INTERVAL CALCULATIONS
C  PROGRAMMER:  L. NEWLIN
C        DATE:  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               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:   NUMREG, NSETS, NP, LNSTR, LNNF
C  OUTPUTS:  SX2, SXY, SY2, DD, SWHAT2, SUHAT2, NPPR

C     IMPLICIT NONE

      INTEGER 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)
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                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
C               L AND DATA SET J (X = Ln S)
C  MEANY()    1-D ARRAY CONTAINING SAMPLE Y MEAN FOR POINTS FROM REGION
C               L AND DATA SET J (Y = Ln N)
C  NP()       2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C               SET IN EACH REGION
C  NPPR()     1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1))-1) OVER
C                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
C               REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
C  SWHAT2()   1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C               REGRESSION FOR THE BEST FIT LINE FOR EACH REGION
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   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

         DO 200 J = 0, NSETS
C           FIRST CALCULATE SAMPLE X AND Y MEANS
C           FOR DATA SET J IN REGION L
            MEANY(J) = 0.0
            MEANX(J) = 0.0
            IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' J =', J,
     &                        ' NP =', NP(J,L)

            DO 250 K = 1, NP(J,L)
               MEANY(J) = MEANY(J) + LNNF(K,J,L)
               MEANX(J) = MEANX(J) + LNSTR(K,J,L)
               IF (IOUT .EQ. 10) WRITE(8,*) 'LNNF =', LNNF(K,J,L),
     &               ' LNSTR =', LNSTR(K,J,L)
  250       CONTINUE

            MEANY(J) = MEANY(J)/FLOAT(NP(J,L))
            MEANX(J) = MEANX(J)/FLOAT(NP(J,L))
            IF (IOUT .EQ. 10) WRITE(8,*) 'MEANY(J) =', MEANY(J),
     &         ' MEANX(J) =', MEANX(J)

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

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

            NPPR(L) = NPPR(L) + NP(J,L) - 1
            IF (IOUT .EQ. 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
C            OUR MODEL
             WRITE(8,*) 'ERROR: SXY >= 0 IN REGION', L
             CALL TRMNAT
          ENDIF

          NPPR(L) = NPPR(L) - 1

          IF (NPPR(L) .LE. 0) THEN
             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))

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,*) 'NPPR(L) =', NPPR(L), ' SY2(L) =', SY2(L),
     &                  ' SX2(L) =', SX2(L)
             WRITE(8,*) 'SXY(L) =', SXY(L), ' DD(L) =', DD(L),
     &                  ' BB(L) =', BB(L)
          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) =',
     &             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:  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               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FINDMC (NUMREG, CZERO, SX2, SXY, SY2, MCPNT, MC)

C  INPUTS:   NUMREG, CZERO, SX2, SXY, SY2
C  OUTPUTS:  MCPNT, MC

C     IMPLICIT NONE
```

```
      INTEGER  MAXREG
      PARAMETER (MAXREG = 3)
      COMMON   IOUT
      INTEGER  IOUT, L, MCPNT(MAXREG), NUMREG
      REAL     ARG1, ARG2, CZERO, CZERO2, MC(2, MAXREG), SX2(MAXREG),
     &         SXY(MAXREG), SY2(MAXREG)

C                  LIST OF VARIABLES
C
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     INITIALIZE VARIABLES

      DO 50 L = 1, MAXREG
         MCPNT(L) = 0
         MC(1,L) = 0.0
         MC(2,L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATIONS

      CZERO2 = CZERO ** 2

      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'CZERO = ', CZERO, ' CZERO2 = ', CZERO2

      DO 100 L = 1, NUMREG

         ARG1 = SX2(L) - CZERO2
         ARG2 = 0.0

         IF (CZERO .EQ. 0.0) THEN

C           THEN NO M CONSTRAINT IS REQUIRED

            MCPNT(L) = 0

         ELSEIF (ABS(ARG1) .LT. 1.0E-6) THEN

C           THEN THE CONSTRAINT WILL BE ON THE LOWER BOUND OF M

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

         ELSE

C           THE OTHER TWO POSSIBLE CONSTRAINTS REQUIRE SOME
C           COMMON CALCULATIONS
```

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

          IF (SX2(L) .LT. CZERO2) THEN

C            AGAIN THE M CONSTRAINT IS JUST ON THE LOWER BOUND OF M

             MCPNT(L) = 1
             MC(1,L) = (- SXY(L) - ARG2) / ARG1

          ELSE

C            SX2(L) .GT. CZERO2 -- THIS TIME THE M CONSTRAINT IS A RANGE

             MCPNT(L) = 2
             MC(1,L) = (- SXY(L) - ARG2) / ARG1
             MC(2,L) = (- SXY(L) + ARG2) / ARG1

          ENDIF

       ENDIF

  100 CONTINUE

    IF (IOUT .EQ. 10) THEN

       DO 200 L = 1, NUMREG

          WRITE(8,*) 'L = ', L, ' MCPNT = ', MCPNT(L)
          WRITE(8,*) 'ARG1 = ', ARG1, ' ARG2 = ', ARG2
          WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)

  200    CONTINUE

    ENDIF

    RETURN
    END


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


C  SUBROUTINE INTRVL CALCULATES THE 95% CONFIDENCE INTERVAL, Io, ON
C  C; AND THE 95% CONFIDENCE INTERVAL, Jo, ON M
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  5OCT87    COMMENTS:  15SEP89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INTRVL (NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO,
     &                   JZERO, MCHAT)

C  INPUTS:   NUMREG, SX2, DD, SWHAT2, SUHAT2, NPPR
C  OUTPUTS:  IZERO, JZERO, MCHAT
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER CHITAB, MAXREG, TTAB

      PARAMETER (CHITAB = 150, MAXREG = 3, TTAB = 31)
```

```
      COMMON  IOUT

      INTEGER I, IOUT, L, NPPR(MAXREG), NUM, NUMREG

      REAL    ARG, CHI025(CHITAB), CHI975(CHITAB), DD(MAXREG),
     &        IZERO(2, MAXREG), JZERO(2, MAXREG), MCHAT(2, MAXREG),
     &        SUHAT, SUHAT2(MAXREG), SWHAT, SWHAT2(MAXREG), SX,
     &        SX2(MAXREG), T, T025(TTAB)

      DATA (CHI025(I), I = 1, 75) /
     &             0.000982069, 0.506356, 0.215795, 0.484419, 0.831211,
     &             1.237347, 1.68987, 2.17973, 2.70039, 3.24697,
     &             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,
     &             29.15,    29.95,   30.75,   31.55,   32.3574,
     &             33.16,    33.96,   34.77,   35.58,   36.39,
     &             37.21,    38.02,   38.84,   39.66,   40.4817,
     &             41.30,    42.12,   42.95,   43.77,   44.60,
     &             45.43,    46.26,   47.09,   47.92,   48.7576,
     &             49.59,    50.42,   51.26,   52.10,   52.94 /
      DATA (CHI025(I), I = 76, 150) /
     &             53.78,    54.62,   55.46,   56.30,   57.1532,
     &             57.80,    58.84,   59.69,   60.54,   61.39,
     &             62.24,    63.09,   63.94,   64.79,   65.6466,
     &             66.50,    67.35,   68.21,   69.07,   69.92,
     &             70.78,    71.64,   72.50,   73.36,   74.2219,
     &             75.08,    75.94,   76.80,   77.67,   78.53,
     &             79.40,    80.27,   81.13,   82.00,   82.87,
     &             83.73,    84.60,   85.47,   86.34,   87.21,
     &             88.08,    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 /

      DATA (CHI975(I), I = 1, 75) /
     &             5.02389,  7.37776, 9.34840, 11.1433, 12.8325,
     &             14.4494,  16.0128, 17.5346, 19.0228, 20.4831,
     &             21.9200,  23.3367, 24.7356, 26.1190, 27.4884,
     &             28.8454,  30.1910, 31.5264, 32.8523, 34.1696,
     &             35.4789,  36.7807, 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,
     &             54.44,    55.67,   56.89,   58.12,   59.3417,
     &             60.56,    61.77,   62.99,   64.20,   65.41,
     &             66.62,    67.82,   69.02,   70.22,   71.4202,
     &             72.61,    73.81,   75.00,   76.19,   77.38,
     &             78.57,    79.75,   80.93,   82.12,   83.2976,
     &             84.48,    85.65,   86.83,   88.00,   89.18,
     &             90.35,    91.52,   92.69,   93.86,   95.0231,
     &             96.19,    97.35,   98.52,   99.68,   100.84 /
      DATA (CHI975(I), I = 76, 150) /
     &             102.00,   103.16,  104.31,  105.47,  106.629,
     &             107.78,   108.94,  110.09,  111.24,  112.39,
     &             113.54,   114.69,  115.84,  116.99,  118.136,
     &             119.28,   120.43,  121.57,  122.72,  123.86,
     &             125.00,   126.14,  127.28,  128.42,  129.561,
     &             130.70,   131.84,  132.98,  134.11,  135.25,
     &             136.38,   137.52,  138.65,  139.79,  140.92,
     &             142.05,   143.18,  144.31,  145.44,  146.57,
     &             147.70,   148.83,  149.96,  151.09,  152.21,
     &             153.34,   154.47,  155.59,  156.72,  157.84,
     &             158.97,   160.09,  161.21,  162.33,  163.46,
     &             164.58,   165.70,  166.82,  167.94,  169.06,
     &             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 /
```

```
C  VALUES FOR THE TABLES ABOVE WERE OBTAINED IN THE FOLLOWING MANNER:
C
C     1 - 30, 40, 50, 60, 70, 80, 90, 100 -- Theil, pp. 718-719
C
C     31-39, 41-49, 51-59, 61-69, 71-79, 81-89, 91-99, 101-150
C        -- CALCULATED USING CUBE RULE APPROXIMATION

      DATA T025 / 12.706, 4.303, 3.182, 2.776, 2.571, 2.447,
     &             2.365, 2.306, 2.262, 2.228, 2.201, 2.179,
     &             2.160, 2.145, 2.131, 2.120, 2.110, 2.101,
     &             2.093, 2.086, 2.080, 2.074, 2.069, 2.064,
     &             2.060, 2.056, 2.052, 2.048, 2.045, 2.042, 1.960 /

C                         LIST OF VARIABLES
C
C  ARG          INTERMEDIATE CALCULATION VARIABLE
C  CHI025()     TABLE OF 0.025 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHI975()     TABLE OF 0.975 PERCENTAGE POINTS, CHI-SQUARE DISTRIBUTION
C  CHITAB       MAXIMUM NUMBER OF DEGREES OF FREEDOM IN CHI025 AND CHI975
C  DD()         1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  I            CONTROLS LOOP FOR CHI025() AND CHI975()
C  IOUT         OUTPUT DUMP CONTROLLER
C  IZERO()      2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                 FOR EACH REGION
C  JZERO()      2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C  L            CONTROLS DO LOOP FOR EACH REGION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                 FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                 MCHAT(1,L) = -DD, THE ESTIMATE FOR M AND
C                 MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
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  NUM          EQUAL TO NPPR(L) FOR A SET OF CALCULATIONS
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  SUHAT        EQUAL TO SUHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SUHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT        EQUAL TO SWHAT2(L)**0.5 FOR A SET OF CALCULATIONS
C  SWHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SX           EQUAL TO (NPPR(L)*SX2(L))**0.5 FOR A SET OF CALCULATIONS
C  SX2()        1-D ARRAY CONTAINING SAMPLE X VARIANCE FOR EACH REGION
C                 (X = Ln S)
C  T            VALUE OF T025() USED IN CALCULATIONS
C  T025()       TABLE OF 0.025 PERCENTAGE POINTS, T DISTRIBUTION
C  TTAB         MAXIMUM NUMBER OF DEGREES OF FREEDOM IN T025

C  INITIALIZE IZERO, JZERO AND MCHAT

      DO 50 L = 1, MAXREG
         IZERO(1,L) = 0.0
         IZERO(2,L) = 0.0
         JZERO(1,L) = 0.0
         JZERO(2,L) = 0.0
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
   50 CONTINUE

C  CHECK THAT ALLOWABLE DEGREES OF FREEDOM HAVE NOT BEEN EXCEEDED

      DO 75 L = 1, NUMREG
         IF (NPPR(L) .GT. CHITAB) THEN
            WRITE(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,*) 'IZERO(1,L) =', IZERO(1,L), ' IZERO(2,L) =',
     &                  IZERO(2,L)
            WRITE(8,*) 'JZERO(1,L) =', JZERO(1,L), ' JZERO(2,L) =',
     &                  JZERO(2,L)
            WRITE(8,*) 'MCHAT(1,L) =', MCHAT(1,L), ' MCHAT(2,L) =',
     &                  MCHAT(2,L)
         ENDIF
  100 CONTINUE

      RETURN
      END


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


C  SUBROUTINE GTPVAR CALCULATES THE EXTENT OF DEPARTURE FROM THE MULTIPLE
C  HEAT MEDIAN S/N CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM 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

      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


C                        LIST OF VARIABLES
C
C   IOUT          OUTPUT DUMP CONTROLLER
C   J             CONTROLS DO LOOP FOR EACH DATA SET
C   L             CONTROLS DO LOOP FOR EACH REGION
C   LAMN          LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                   CONSTANT OVER REGIONS AND COMPONENTS
C   MAXREG        MAXIMUM NUMBER OF REGIONS ALLOWED
C   MAXSET        MAXIMUM NUMBER OF S/N DATA SETS ALLOWED
C   MCHAT()       2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                   FOR EACH REGION, BASED ON MATERIALS DATA ONLY --
C                   MCHAT(1,L) = -DD(L), THE ESTIMATE FOR M AND
C                   MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C   NP()          2-D ARRAY CONTAINING NUMBER OF POINTS OF EACH S/N DATA
C                   SET IN EACH REGION
C   NSETS         NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUM()         EQUAL TO Nj-1 FOR EACH REGION WHERE Nj IS THE SUM OF THE
C                   NUMBER OF POINTS IN EACH DATA SET
C   NUMREG        NUMBER OF REGIONS OF INTEREST
C   PSIG2()       1-D ARRAY CONTAINING ESTIMATES OF THE MATERIALS PROCESS
C                   VARIATION IN EACH REGION
C   PVAR          THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                   CURVE WARRANTED BY THE AVAILABLE INFORMATION
C   SUM           WEIGHTED SUM OF THE PSIG2s -- USED TO CALCULATE A WEIGHTED
C                   AVERAGE
C   TOTAL         SUM OF NUM() OVER ALL REGIONS

C     INITIALIZE VARIABLES

      SUM = 0.0
      TOTAL = 0.0

      DO 50 L = 1, MAXREG
         PSIG2(L) = 0.0
         NUM(L) = 0
   50 CONTINUE

      DO 100 L = 1, NUMREG
         DO 150 J = 0, NSETS
            NUM(L) = NUM(L) + NP(J,L)
  150    CONTINUE
         NUM(L) = NUM(L) - 1
         TOTAL = TOTAL + NUM(L)
  100 CONTINUE

      DO 200 L = 1, NUMREG
         PSIG2(L) = (LAMN - 1.0) * MCHAT(2,L) ** 2
         SUM = SUM + PSIG2(L) * NUM(L)
  200 CONTINUE

      IF (IOUT .EQ. 10) THEN
         WRITE(8,*) 'LAMN = ', LAMN
         DO 300 L = 1, NUMREG
            WRITE(8,*) 'L = ', L, ' NUM = ', NUM(L)
            WRITE(8,*) 'MCHAT = ', MCHAT(2,L), ' PSIG2 = ', PSIG2(L)
  300    CONTINUE
         WRITE(8,*) 'TOTAL = ', TOTAL, ' SUM = ', SUM
      ENDIF

      PVAR = SUM / FLOAT (TOTAL)

      RETURN
      END
```

```
C*****************************************************************************

C  SUBROUTINE FNDRNG COMBINES THE PRIOR ENGINEERING KNOWLEDGE ON BOTH
C  M AND Co WITH THE 95% CONFIDENCE INTERVALS (JZERO FROM INTRVL)
C  TO OBTAIN POSTERIOR CREDIBILITY RANGES ON M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FNDRNG (NUMREG, MPNT, MZERO, MCPNT, MC, JZERO,
     &                   MCHAT, RANGEM)

C  INPUTS:    NUMREG, MPNT, MZERO, MCPNT, MC, JZERO, MCHAT
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:   TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    JZERO(2, MAXREG), LOWER, MC(2, MAXREG), MCHAT(2, MAXREG),
     &        MZERO(2, MAXREG), RANGEM(2, MAXREG), UPPER


C                            LIST OF VARIABLES
C
C  IOUT        OUTPUT DUMP CONTROLLER
C  JZERO()     2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                FOR EACH REGION
C  L           CONTROLS DO LOOP FOR EACH REGION
C  LOWER       LOWER BOUND OF INTERSECTION
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MC()        2-D ARRAY CONTAINING VALUES OF THE RANGES ON M FOR EACH
C                REGION CONSISTENT WITH GIVEN VALUE OF Co AND THE DATA
C                -- MC(1,L) IS THE LOWER BOUND AND MC(2,L) IS THE UPPER
C                BOUND
C  MCHAT()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF M AND C
C                FOR EACH REGION -- MCHAT(1,L) = - DD(L), THE ESTIMATE
C                FOR M AND MCHAT(2,L) = SUHAT, THE ESTIMATE FOR C
C  MCPNT()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MC() FOR EACH REGION
C  MPNT()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MZERO() FOR EACH REGION
C  MZERO()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGES ON M FOR
C                EACH REGION -- MZERO(1,L) IS THE LOWER BOUND AND MZERO(2,L)
C                IS THE UPPER BOUND
C  NUMREG      NUMBER OF REGIONS OF INTEREST
C  RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                RANGEM(2,L) IS THE UPPER BOUND
C  UPPER       UPPER BOUND OF INTERSECTION


C     INITIALIZE VARIABLES

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

C  PERFORM CALCULATIONS FOR EACH REGION OF INTEREST

      DO 100 L = 1, NUMREG
```

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

      IF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 0)) THEN

C        THERE IS NO EXOGENOUS INFORMATION
C        ASSUME RANGE TO BE Jo

         RANGEM(1,L) = JZERO(1,L)
         RANGEM(2,L) = JZERO(2,L)

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' JZERO(1,L) = ', JZERO(1,L)
            WRITE(8,*) 'RANGEM(2,L) = ', RANGEM(2,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 1)) THEN

C        NO PRIOR RANGE ON M, BUT THERE IS A LOWER BOUND ON M DUE
C        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

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

      ELSEIF ((MPNT(L) .EQ. 0) .AND. (MCPNT(L) .EQ. 2)) THEN

C        THERE IS NO PRIOR RANGE ON M, BUT THERE IS A RANGE
C        CORRESPONDING TO THE Co CONSTRAINT, ADJUST Jo ACCORDINGLY

         LOWER = AMAX1(JZERO(1,L), MC(1,L))
         UPPER = AMIN1(JZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                 ' JZERO(2,L) = ', JZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L), ' MC(2,L) = ', MC(2,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
         ENDIF

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

C        THERE IS A POINT PRIOR ON M -- THIS OVERRIDES ALL OTHER
C        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
C           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)
               WRITE(8,*) '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

C           THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C           CONSTRAINT, INTERSECT Jo AND Mo, ADJUSTING THE LOWER BOUND
C           BY Mc ACCORDINGLY

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
            ENDIF

            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                    ' JZERO(2,L) = ', JZERO(2,L)
               WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                    ' MZERO(2,L) = ', MZERO(2,L)
               WRITE(8,*) 'MC(1,L) = ', MC(1,L)
               WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
               WRITE(8,*) 'RANEGM(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
C           INTERSECT THESE TWO RANGES WITH Jo

            LOWER = AMAX1(JZERO(1,L), MZERO(1,L), MC(1,L))
            UPPER = AMIN1(JZERO(2,L), MZERO(2,L), MC(2,L))
            IF (UPPER .LT. LOWER) THEN
               WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Jo, Mo, ',
     &                    'AND Mc'
               CALL TRMNAT
            ELSE
               RANGEM(1,L) = LOWER
               RANGEM(2,L) = UPPER
```

```
          ENDIF

          IF (IOUT .EQ. 10) THEN
             WRITE(8,*) 'JZERO(1,L) = ', JZERO(1,L),
     &                  ' JZERO(2,L) = ', JZERO(2,L)
             WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                  ' MZERO(2,L) = ', MZERO(2,L)
             WRITE(8,*) 'MC(1,L) = ', MC(1,L)
             WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
             WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                  ' RANGEM(2,L) = ', RANGEM(2,L)
          ENDIF

       ELSE

          WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
          CALL TRMNAT

       ENDIF

C         RESTRICT RANGE TO BE NON-NEGATIVE

       RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

       IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C      CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

         IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &        .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &      WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &                 'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END


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


C  SUBROUTINE ADDREG ADDS THE INFORMATION ON M RANGES FOR REGIONS
C  WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  2FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE ADDREG (RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT)

C  INPUTS:   RANGEM, MCHAT, NNODAT, NUMREG, MZERO, MPNT
C  OUTPUTS:  RANGEM, MCHAT, NUMREG

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, LL, MPNT(MAXREG), NNODAT, NUMREG

      REAL    MCHAT(2, MAXREG), MZERO(2, MAXREG), RANGEM(2, MAXREG)
```

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

      IF (IOUT .EQ. 10) WRITE(8,*) 'NUMREG =', NUMREG

      DO 100 L = 1, NNODAT
         NUMREG = NUMREG + 1
         LL = NUMREG
         IF (IOUT .EQ. 10) WRITE(8,*) 'L =', L, ' NUMREG =', NUMREG,
     &      ' LL =', LL, ' MPNT(LL) =', MPNT(LL)

         IF ((MPNT(LL) .EQ. 1) .OR. (MPNT(LL) .EQ. 2)) THEN
C           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

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            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


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
C     VERSION:  MATCHR V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
```

```
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
      SUBROUTINE CONCAV (NUMREG, RANGEM)

C  INPUTS:    NUMREG, RANGEM
C  OUTPUTS:   RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    RANGEM(2, MAXREG), TESTM

C                           LIST OF VARIABLES
C
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  RANGEM()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C               FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C               RANGEM(2,L) IS THE UPPER BOUND
C  TESTM     UPPER BOUND OF RANGE ON M IN REGION L-1 -- USED DURING
C              CONCAVITY ADJUSTMENT

C     ADJUST RANGE TO INSURE CONCAVITY

      DO 100 L = NUMREG, 2, -1

         IF (RANGEM(2,L-1) .EQ. 0.0) THEN
C           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
C           RANGE IS AN INTERVAL IN REGION L-1
            TESTM = AMAX1(RANGEM(1,L), RANGEM(2,L))
            IF (TESTM .LT. RANGEM(1,L-1)) THEN
               WRITE(8,*) 'ERROR:  POSTERIOR INTERVAL IN REGION ', L,
     &            ' IS INCONSISTENT WITH THE POSTERIOR INTERVAL IN ',
     &            'REGION ', L-1
               CALL TRMNAT
            ELSE
               RANGEM(2,L-1) = AMIN1(RANGEM(2,L-1), TESTM)
            ENDIF
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'RANGEM(1,L-1) =', RANGEM(1,L-1),
     &                 ' RANGEM(2,L-1) =', RANGEM(2,L-1)
            WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                 ' RANGEM(2,L) =', RANGEM(2,L)
            WRITE(8,*) 'TESTM =', TESTM, ' L = ', L
         ENDIF

  100 CONTINUE

      RETURN
      END

C*************************************************************************
```

```
C   SUBROUTINE MEDIAN CALCULATES THE MEDIAN VALUES OF M AFTER Jo HAS
C   BEEN ADJUSTED BECAUSE OF PRIOR INFORMATION ON M OR Co
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  5OCT87      COMMENTS:  1DEC87
C      VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                       V8.4, V8.5
C                MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE MEDIAN (NUMREG, RANGEM, MEDM)

C   INPUTS:    NUMREG, RANGEM
C   IOUTPUT:   MEDM

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     LOWERM, MEDM(MAXREG), RANGEM(2, MAXREG)


C                    LIST OF VARIABLES
C
C   IOUT        OUTPUT DUMP CONTROLLER
C   L           CONTROLS DO LOOP FOR EACH REGION
C   LOWERM      LOWER BOUND OF M RANGE (DUE TO CONCAVITY CONSIDERATION)
C                 TO BE USED IN MEDIAN CALCULATION
C   MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C   MEDM()      1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C   NUMREG      NUMBER OF REGIONS OF INTEREST
C   RANGEM()    2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND

C     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
C           -- 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)
         WRITE(8,*) '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
C     VERSION:  MATCHR V8.3, V8.4, V8.5   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 EXPCTD (NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG,
     &                   ZROREG, NBND, BIGK1, BZHAT)

C  INPUTS:  NCOMPS, MEDM, NPTS, STR, NF, SZERO, NUMREG, ZROREG, NBND
C  OUTPUTS:  BIGK1, BZHAT
C  SUBPROGRAMS:  TRNSFM, SMNVAR, KBETA, FINDK, FINDSB, KOMO

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NCOMPS, NP, NPTS(MAXREG), NUMREG, ZROREG

      REAL     BIGK(0:MAXREG), BIGK1, 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)


C                     LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  BIGK1      EQUAL TO BIGK(1)
C  BZHAT      E(BETAo)
C  FACTR      A SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  KHAT       E(k)
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MEDM()     1-D ARRAY CONTAINING VALUES OF THE MEDIAN M FOR EACH REGION
C  MM()       1-D ARRAY CONTAINING VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NCOMPS     Number of COMPonents -- 1 FOR STRESS AND STRAIN WHEN DECOMPOSED
C               DATA UNAVAILABLE -- 2 FOR DECOMPOSED STRAIN DATA
C  NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C               SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
```

```
C  NP          TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C                 DATA SET
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN EACH REGION FOR
C                 THE SPECIFIC MATERIAL S/N DATA SET
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 REGION
C                CONTAINED IN NBND()
C  STR()       2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C                DATA SET BROKEN INTO REGIONS (PSI OR %)
C  SZ2         SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  SZERO       STRESS TENSILE TEST POINT, So
C  TRBIGK()    1-D ARRAY CONTAINING VALUES OF K. IN THIS ROUTINE
C                TRBIGK(i) = BIGK(i)
C  ZROREG      ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                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  INITIALIZE VARIABLES

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

C  CREATE MM() ARRAY FROM MEDM() ARRAY

      DO 100 L = 1, NUMREG
         MM(L) = MEDM(L)
  100 CONTINUE

C  TRANSFORM THE S/N DATA INTO THE VARIABLE Z = Ln(X)

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

C  CALCULATE THE SAMPLE MEAN AND VARIANCE OF Z = Ln(X)

      CALL SMNVAR (NP, ZZ, MEANZ, SZ2)

C  CALCULATE BETAo AND k

      CALL KBETA (MEANZ, SZ2, KHAT, BZHAT)

C  CALCULATE THE VALUES OF K, WHERE A = K ** M FOR EACH REGION

      CALL FINDK (BZHAT, KHAT, MM, NBND, NUMREG, BIGK)

      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), SBND(L)
            IF (IOUT .EQ. 10) WRITE(8,910) L, MM(L), TRBIGK(L),
```

```
     &                                    NBND(L), SBND(L)
  200     CONTINUE
          WRITE(7,920)
        ELSE
          WRITE(7,930) MM(1), BIGK(1), KHAT
        ENDIF
C  FORMAT STATEMENTS
  900 FORMAT(///,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',//,2X,
     &        'NUMBER OF REGIONS:',I4,5X,'E(BETAo) =',F8.4,5X,'E(k) =',
     &        F8.4,//,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',7X,
     &        'STRESS BOUND',/)
  910 FORMAT(5X,I1,5X,F9.5,5X,E12.5,5X,E9.3,9X,E11.5)
  920 FORMAT(///)
  930 FORMAT(//,2X,'PARAMETER VALUES FOR MEDIAN S/N CURVE',
     &        //,11X,'m',14X,'K',13X,'E(k)',
     &        //,7X,F8.5,5X,E12.5,6X,F7.4,/)

      RETURN
      END

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

C  SUBROUTINE MUSIG CALCULATES THE POSTERIOR NORMAL DISTRIBUTION PARAMETERS:
C  MEAN, MU, AND STANDARD DEVIATION, SIG; FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      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
C  ARG         INTERMEDIATE CALCULATION VARIABLE
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
C  DELTA()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MU() AND
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                   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     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 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     SUMX2       EQUAL TO NPPR() * SX2() FOR A PARTICULAR REGION
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     INITALIZE ARRAYS

      DO 50 L = 1, MAXREG
         MCHAT(1,L) = 0.0
         MCHAT(2,L) = 0.0
         MU(L) = 0.0
         SIG(L) = 0.0
   50 CONTINUE

C     BEGIN CALCULATION FOR EACH REGION

      DO 100 L = 1 , NUMREG

         MCHAT(1,L) = - DD(L)
         MCHAT(2,L) = SQRT (SUHAT2(L))
         SUMX2 = NPPR(L) * SX2(L)
         ARG = SUMX2 + DELTA(L)

         IF (DELTA(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE MEAN WAS SUPPLIED
C           USE THE ESTIMATE OF M
            MU(L) = MCHAT(1,L)
         ELSE
C           UPDATE THE ESTIMATE OF M WITH Mo USING DELTA
            MU(L) = (MCHAT(1,L) * SUMX2 + MO(L) * DELTA(L)) / ARG
         ENDIF

         IF (SIGMA2(L) .EQ. 0.0) THEN
C           THEN NO PRIOR VALUE OF THE VARIANCE WAS SUPPLIED
C           USE SWHAT2 AS AN ESTIMATE OF SIGMA-HAT-2
            SIG(L) = SQRT (SWHAT2(L) / ARG)
         ELSE
            SIG(L) = SQRT (SIGMA2(L) / ARG)
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'L = ', L, ' DD = ', DD(L), ' MCHAT1 = ',
     &                  MCHAT(1,L)
            WRITE(8,*) 'SUHAT2 = ', SUHAT2(L),   ' MCHAT2 = ',
     &                  MCHAT(2,L)
            WRITE(8,*) 'NPPR = ', NPPR(L), ' SX2 = ', SX2(L),
     &                 ' SUMX2 = ', SUMX2
            WRITE(8,*) 'DELTA = ', DELTA(L), ' ARG = ', ARG
            WRITE(8,*) 'MO = ', MO(L), ' MU = ', MU(L)
            WRITE(8,*) 'SWHAT2 = ', SWHAT2(L), ' SIGMA2 = ', SIGMA2(L),
     &                 ' SIG = ', SIG(L)
         ENDIF

  100 CONTINUE

      RETURN
```

```
      END

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     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE NORRNG (NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT, RANGEM)

C  INPUTS:   NUMREG, MPNT, MZERO, MCPNT, MC, MCHAT
C  OUTPUTS:  RANGEM
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, MCPNT(MAXREG), MPNT(MAXREG), NUMREG

      REAL    LOWER, MC(2, MAXREG), MCHAT(2, MAXREG), MZERO(2, MAXREG),
     &        RANGEM(2, MAXREG), UPPER

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

C     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
C        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 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. 1)) THEN

C        THERE IS A PRIOR RANGE ON M AND A LOWER BOUND DUE TO Co
C        CONSTRAINT ADJUST THE LOWER BOUND OF Mo BY Mc

         LOWER = AMAX1(MZERO(1,L), MC(1,L))
         UPPER = MZERO(2,L)
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
            WRITE(8,*) 'RANEGM(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
C        INTERSECT THESE TWO RANGES

         LOWER = AMAX1(MZERO(1,L), MC(1,L))
         UPPER = AMIN1(MZERO(2,L), MC(2,L))
         IF (UPPER .LT. LOWER) THEN
            WRITE(8,*) 'ERROR:  NO INTERSECTION BETWEEN Mo AND Mc'
            CALL TRMNAT
         ELSE
            RANGEM(1,L) = LOWER
            RANGEM(2,L) = UPPER
         ENDIF

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'MZERO(1,L) = ', MZERO(1,L),
     &                 ' MZERO(2,L) = ', MZERO(2,L)
            WRITE(8,*) 'MC(1,L) = ', MC(1,L)
            WRITE(8,*) 'LOWER = ', LOWER, ' UPPER = ', UPPER
```

```
            WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L),
     &                 ' RANGEM(2,L) = ', RANGEM(2,L)
          ENDIF

        ELSE

          WRITE(8,*) 'ERROR:  PRIOR ON M INCORRECTLY SPECIFIED IN ', L
          CALL TRMNAT

        ENDIF

C       RESTRICT RANGE TO BE NON-NEGATIVE

        RANGEM(1,L) = AMAX1(RANGEM(1,L), 0.0)

        IF (IOUT .EQ. 10) WRITE(8,*) 'RANGEM(1,L) = ', RANGEM(1,L)

  100 CONTINUE

C     CHECK TO SEE IF E(m) IS IN POSTERIOR RANGE

      DO 300 L = 1, NUMREG

        IF ((MCHAT(1,L) .LT. RANGEM(1,L))
     &      .OR. (MCHAT(1,L) .GT. RANGEM(2,L)))
     &    WRITE(8,*) 'NOTE:  E(m) IS NOT IN THE POSTERIOR RANGE ',
     &               'ON m IN REGION ', L

  300 CONTINUE

      RETURN
      END

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

C  SUBROUTINE ADDRGN ADDS THE INFORMATION ON M RANGES AND NORMAL
C  DISTRIBUTION PARAMETERS FOR REGIONS WITHOUT DATA
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  10FEB88      FORMAT/COMMENTS:  12AUG91
C     VERSION:  MATCHR V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

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

C  INPUTS:    RANGEM, MCHAT, MU, SIG, NNODAT, NUMREG, MZERO, MPNT,
C             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)


C                  LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
```

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

      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
C           POSTERIOR ON M IS SAME AS PRIOR ON M
            RANGEM(1,LL) = MZERO(1,LL)
            RANGEM(2,LL) = MZERO(2,LL)
            MU(LL) = MO(LL)
            SIG(LL) = SQRT(SIGMA2(LL))
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,LL) =', RANGEM(1,LL),
     &                    ' MZERO(1,LL) = ', MZERO(1,LL)
               WRITE(8,*) 'RANGEM(2,LL) =', RANGEM(2,LL),
     &                    ' MZERO(2,LL) = ', MZERO(2,LL)
               WRITE(8,*) 'MU(LL) = ', MU(LL), ' MO(LL) = ', MO(LL)
               WRITE(8,*) 'SIG(LL) = ', SIG(LL), ' SIGMA2(LL) = ',
     &                    SIGMA2(LL)
            ENDIF

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

            IF (RANGEM(2,LL) .EQ. 0.0) THEN
               MCHAT(1,LL) = RANGEM(1,LL)
               MU(LL) = RANGEM(1,LL)
               SIG(LL) = 0.0
            ELSE
               MCHAT(1,LL) = (RANGEM(1,LL) + RANGEM(2,LL)) / 2.0
            ENDIF
            IF (IOUT .EQ. 10) WRITE(8,*) 'MCHAT =', MCHAT(1,LL),
     &         ' MU = ', MU(LL), ' SIG = ', SIG(LL)
         ELSE
            WRITE(8,*) 'ERROR:  OVERALL PRIOR RANGE INCORRECTLY ',
     &                 'SPECIFIED IN REGION WITHOUT DATA'
            CALL TRMNAT
         ENDIF
  100 CONTINUE

      RETURN
      END
```

```
C***********************************************************************


C  SUBROUTINE DECOMP CONTROLS THE CALCULATIONS FOR THE INFORMATION
C  AGGREGATION MODEL PORTION OF THE MATERIALS CHARACTERIZATION MODEL
C  FOR THE STRAIN FORMULATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88      FORMAT/COMMENTS:  15SEP89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, 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 DECOMP (RANGMP, RANGME, MUP, SIGP, MUE, SIGE, NF,
     &                   RFNP, NBND, RFSTRP, RFSTRE, VARY, MPROC,
     &                   KRATIO, PVAR)

C  INPUTS:    READS DATA FROM SPECFD AND RELATD; VARY, MPROC
C  OUTPUTS:   RANGMP, RANGME, MUP, SIGP, MUE, SIGE, NF, RFNP,
C             NBND, RFSTRP, RFSTRE, KRATIO, PVAR
C  SUBPROGRAMS:  INITD, RDECHO, PREP, SW2SU2, INTRVL, FNDRNG,
C                EXPCTD, PECOMP, MUSIG, NORRNG, GTPVAR
C        FILES:  5:RELATD-OLD; 6:RELATO-NEW

C     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), MPROC, 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, KHATP, KRATIO, LAMN,
     &         LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &         LNSTRE(MAXDAT, 0:MAXSET, MAXREG),
     &         LNSTRP(MAXDAT, 0:MAXSET, MAXREG), MCE(2, MAXREG),
     &         MCHAT(2, MAXREG), MCHATE(2, MAXREG), MCHATP(2, MAXREG),
     &         MCP(2, MAXREG), MEDME(MAXREG), MEDMP(MAXREG),
     &         MOE(MAXREG), MOP(MAXREG), MUE(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),
     &         RAWSTR(MAXDAT, 0:MAXSET), RFSTRE(MAXDAT, MAXREG),
     &         RFSTRP(MAXDAT, MAXREG), SE(MAXDAT, 0:MAXSET),
     &         SIG2E(MAXREG), SIG2P(MAXREG), SIGE(MAXREG), SIGP(MAXREG),
     &         SP(MAXDAT, 0:MAXSET), STRE(MAXDAT, 0:MAXSET),
     &         STRP(MAXDAT, 0:MAXSET), SUHAT2(MAXREG), SWHAT2(MAXREG),
     &         SX2(MAXREG), SXY(MAXREG), SY2(MAXREG),
     &         TNSILE(0:MAXSET, MAXTNS)

      CHARACTER*40  DESCRP(0:MAXSET)


C                     LIST OF VARIABLES
C
C  BZERO       VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C  BZEROE      VALUE OF WEIBULL PARAMETER, BETAoe, CHARACTERIZING ELASTIC
C                COMPONENTS OF S/N DATA SET
C  BZEROP      VALUE OF WEIBULL PARAMETER, BETAop, CHARACTERIZING PLASTIC
C                COMPONENTS OF S/N DATA SET
C  DD()        1-D ARRAY CONTAINING SXY(L)/SX2(L) FOR EACH REGION
```

```
C  DELTAE()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MUE() AND
C                 SIGE() CALCULATION
C  DELTAP()     1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MUP() AND
C                 SIGP() CALCULATION
C  DESCRP()     1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C  IOUT         OUTPUT DUMP CONTROLLER
C  IZERO()      2-D ARRAY CONTAINING Io, THE 95% CONFIDENCE INTERVALS ON C
C                 FOR EACH REGION
C  J            CONTROLS DO LOOP FOR EACH DATA SET
C  JZERO()      2-D ARRAY CONTAINING Jo, THE 95% CONFIDENCE INTERVALS ON M
C                 FOR EACH REGION
C  KHATE        E(Ke), THE MEDIAN Ke FOR ELASTIC COMPONENTS
C  KHATP        E(Kp), THE MEDIAN Kp FOR PLASTIC COMPONENTS
C  KRATIO       RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  LAMN         LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                 CONSTANT OVER REGIONS AND COMPONENTS
C  LNNF()       3-D ARRAY CONTAINING LN(RAWNF())
C  LNSTRE()     3-D ARRAY CONTAINING LN(SE()) OR LN(STRE())
C  LNSTRP()     3-D ARRAY CONTAINING LN(SP()) OR LN(STRP())
C  MAXDAT       MAXIMUM NUMBER OF POINTS IN 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  MAXTNS       MAXIMUM NUMBER OF TENSILE TESTS PER DATA SET ALLOWED
C  MCE()        2-D ARRAY CONTAINING VALUES OF THE RANGE ON Me CONSISTENT
C                 WITH GIVEN VALUE OF Co AND THE DATA -- NOTE:  THE Co
C                 CONSTRAINT IS NOT APPLICABLE TO THE STRAIN FORMULATION,
C                 BUT IS A NECESSARY PARAMETER TO SOME OF THE SUBROUTINES
C                 SHARED WITH THE STRESS CASE
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF Cp AND Ce,
C                 BASED ON MATERIALS DATA ONLY -- MCHAT(2,1) = MCHATP(2,1),
C                 MCHAT(2,2) = MCHATE(2,1)
C  MCHATE()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF Me AND
C                 Ce FOR ELASTIC COMPONENTS, BASED ON MATERIALS DATA ONLY
C                 -- MCHATE(1,1) = - DD(L), THE ESTIMATE FOR Me AND
C                 MCHATE(2,1) = SUHAT, THE ESTIMATE FOR Ce
C  MCHATP()     2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF Mp AND
C                 Cp FOR PLASTIC COMPONENTS, BASED ON MATERIALS DATA ONLY
C                 -- MCHATP(1,1) = - DD(L), THE ESTIMATE FOR Mp AND
C                 MCHATP(2,1) = SUHAT, THE ESTIMATE FOR Cp
C  MCP()        2-D ARRAY CONTAINING VALUES OF THE RANGE ON Mp CONSISTENT
C                 WITH GIVEN VALUE OF Co AND THE DATA -- NOTE:  THE Co
C                 CONSTRAINT IS NOT APPLICABLE TO THE STRAIN FORMULATION,
C                 BUT IS A NECESSARY PARAMETER TO SOME OF THE SUBROUTINES
C                 SHARED WITH THE STRESS CASE
C  MCPNTE()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MCE() (ALWAYS 0 FOR STRAIN CASE)
C  MCPNTP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MCP() (ALWAYS 0 FOR STRAIN CASE)
C  MEDME()      1-D ARRAY CONTAINING THE MEDIAN Me VALUE
C  MEDMP()      1-D ARRAY CONTAINING THE MEDIAN Mp VALUE
C  MOE()        1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
C                 ELASTIC COMPONENTS
C  MOP()        1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
C                 PLASTIC COMPONENTS
C  MPNTE()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZEROE()
C  MPNTP()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZEROP()
C  MPROC        Materials PROCess variation -- CONTROLS MATERIALS PROCESS
C                 VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MUE()        1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR
C                 ELASTIC COMPONENTS
C  MUP()        1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION MEAN FOR
C                 PLASTIC COMPONENTS
C  MZEROE()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Me FOR
C                 ELASTIC COMPONENTS -- MZEROE(1,1) IS THE LOWER BOUND AND
C                 MZEROE(2,1) IS THE UPPER BOUND
C  MZEROP()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Mp FOR
C                 PLASTIC COMPONENTS -- MZEROP(1,1) IS THE LOWER BOUND AND
C                 MZEROP(2,1) IS THE UPPER BOUND
C  NBND()       1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST

C  NDC()        1-D ARRAY CONTAINING Number of given DeComposed STRAIN
C                 POINTS FOR EACH DATA SET
```

```
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
C                 SET IN EACH REGION
C  NPPR()       1-D ARRAY CONTAINING VALUES OF ((SUM OF (NP()-1) -1) OVER
C                 ALL DATA SETS IN A REGION (Number of Points Per Region)
C  NPTS()       1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NTENS()      1-D ARRAY CONTAINING NUMBER OF TENSILE TEST DATA POINTS
C                 FOR EACH DATA SET
C  PVAR         THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT MEDIAN S/N
C                 CURVE WARRANTED BY THE AVAILABLE INFORMATION
C  RANGME()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON Me
C                 FOR ELASTIC COMPONENTS -- RANGME(1,1) IS THE LOWER
C                 BOUND AND RANGME(2,1) IS THE UPPER BOUND
C  RANGMP()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON Mp
C                 FOR PLASTIC COMPONENTS -- RANGMP(1,1) IS THE LOWER
C                 BOUND AND RANGMP(2,1) IS THE UPPER BOUND
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) OR RAW TOTAL
C                 STRAIN DATA (%) FOR ALL S/N DATA SETS
C  RFNP()       1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET IN EACH REGION
C  RFSTRE()     2-D ARRAY CONTAINING ELASTIC STRAIN POINTS (%), SE() OR
C                 STRE(), FOR THE SPECIFIC MATERIAL S/N DATA
C  RFSTRP()     2-D ARRAY CONTAINING PLASTIC STRAIN POINTS (%), SP() OR
C                 STRP(), FOR THE SPECIFIC MATERIAL S/N DATA
C  SE()         2-D ARRAY CONTAINING GIVEN ELASTIC STRAIN COMPONENTS (%)
C  SIG2E()      1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE
C                 FOR ELASTIC COMPONENTS
C  SIG2P()      1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE
C                 FOR PLASTIC COMPONENTS
C  SIGE()       1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION
C                 STANDARD DEVIATION FOR ELASTIC COMPONENTS
C  SIGP()       1-D ARRAY CONTAINING THE POSTERIOR NORMAL DISTRIBUTION
C                 STANDARD DEVIATION FOR PLASTIC COMPONENTS
C  SP()         2-D ARRAY CONTAINING GIVEN PLASTIC STRAIN COMPONENTS (%)
C  STRE()       2-D ARRAY CONTAINING CALCULATED ELASTIC STRAIN COMPONENTS (%)
C  STRP()       2-D ARRAY CONTAINING CALCULATED PLASTIC STRAIN COMPONENTS (%)
C  SUHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM X ON Y
C                 REGRESSION FOR EACH REGION (X = Ln S, Y = Ln N)
C  SWHAT2()     1-D ARRAY CONTAINING RESIDUAL VARIANCES FROM Y ON X
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  SXY()        1-D ARRAY CONTAINING SAMPLE X, SAMPLE Y COVARIANCE FOR EACH
C                 REGION (X = Ln S, Y = Ln N)
C  SY2()        1-D ARRAY CONTAINING SAMPLE Y VARIANCE FOR EACH REGION
C                 (Y = Ln N)
C  TNSILE()     2-D ARRAY CONTAINING TENSILE TEST DATA -- INDEXED FOR EACH
C                 DATA SET
C  VARY         CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                 VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                 VARIATION;  3 - TRUNCATED NORMAL VARIATION


      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,
     &            NP, LNNF, LNSTRP, LNSTRE, STRP, STRE, MZEROP,
     &            MZEROE, MPNTP, MPNTE, MCP, MCE, MCPNTP, MCPNTE,
     &            RFNP, NF, 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, MZEROP, MPNTP, MZEROE,
     &             MPNTE, DELTAP, DELTAE, MOP, MOE, SIG2P, SIG2E,
     &             KRATIO, LAMN)

C  PREPARE GIVEN DECOMPOSED STRAIN DATA FOR INFORMATION
C  AGGREGATION PROCESSING

      CALL PREP (0, NDC, SP, SE, RAWNF, NP, LNSTRP, LNSTRE, LNNF,
     &           RFNP, RFSTRP, RFSTRE, NF)

C  BEGIN INFORMATION AGGREGATION PROCESSING FOR THE GIVEN PLASTIC
C  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, KHATP,
     &             BZEROP)

C  BEGIN INFORMATION AGGREGATION PROCESSING FOR THE GIVEN ELASTIC
C  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), MCHATE(1,1)
      WRITE(7,925)

C  CALCULATE MEDIAN VALUES FOR ELASTIC S/N CURVE PARAMETERS
C  BASED ON GIVEN DECOMPOSITION DATA

      MEDME(1) = MCHATE(1,1)

      CALL EXPCTD (2, MEDME, RFNP, RFSTRE, NF, 0.0, 1, 1, NBND, KHATE,
     &             BZEROE)

C  DECOMPOSE TOTAL STRAIN DATA BASED ON MEDIAN S/N CURVE PARAMETERS
C  FOR THE GIVEN PLASTIC AND ELASTIC STRAIN COMPONENT DATA TO OBTAIN
C  ESTIMATED PLASTIC AND ELASTIC STRAIN COMPONENTS

      CALL PECOMP (NSETS, DESCRP, NDC, SP, SE, NPTS, RAWSTR,
     &             RAWNF, NTENS, TNSILE, KHATP, KHATE, MCHATP(1,1),
     &             MCHATE(1,1), STRP, STRE)

C  PREPARE ESTIMATED DECOMPOSED STRAIN DATA FOR INFORMATION
C  AGGREGATION PROCESSING
```

```
      CALL PREP (NSETS, NPTS, STRP, STRE, RAWNF, NP, LNSTRP, LNSTRE,
     &           LNNF, RFNP, RFSTRP, RFSTRE, NF)

      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 VARATION SAVE NORMAL

C CALCULATE RESIDUAL VARIANCES

         CALL SW2SU2 (1, NSETS, 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 COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION
C TO OBTAIN POSTERIOR RANGE ON Mp

         CALL FNDRNG (1, MPNTP, MZEROP, MCPNTP, MCP, JZERO, MCHATP,
     &                RANGMP)

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
C           RANGE ON Mp IS A POINT
            MEDMP(1) = RANGMP(1,1)
         ELSE
C           RANGE ON Mp IS AN INTERVAL
            MEDMP(1) = (RANGMP(1,1) + RANGMP(2,1)) / 2.0
         ENDIF

         CALL EXPCTD (2, MEDMP, RFNP, RFSTRP, NF, 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, SXY, SY2, DD,
     &                SWHAT2, SUHAT2, NPPR)

C CALCULATE BOUNDS FOR CONFIDENCE INTERVALS

         CALL INTRVL (1, SX2, DD, SWHAT2, SUHAT2, NPPR, IZERO, JZERO,
     &                MCHATE)

C COMBINE CONFIDENCE INTERVALS AND EXOGENOUS INFORMATION
C TO OBTAIN POSTERIOR RANGE ON Me

         CALL FNDRNG (1, MPNTE, MZEROE, MCPNTE, MCE, 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', RANGME(1,1), RANGME(2,1)
         WRITE(7,960)
```

```
C   CALCULATE MEDIAN VALUES FOR ELASTIC S/N CURVE PARAMETERS
C   BASED ON ESTIMATED DECOMPOSITION DATA

          IF (RANGME(2,1) .EQ. 0.0) THEN
C            RANGE ON Me IS A POINT
             MEDME(1) = RANGME(1,1)
          ELSE
C            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,
     &                 KHATE, BZEROE)

          BZERO = 1.0 / SQRT (0.5 * ((1.0 / BZEROP) ** 2
     &                             + (1.0 / BZEROE) ** 2))

          WRITE(7,985) BZERO

C   CHECK TYPE OF VARIATION DESIRED AND FIX M RANGES AT MEDIAN
C   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

C   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, SXY, SY2, DD,
     &                 SWHAT2, SUHAT2, NPPR)

C   CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION

          CALL MUSIG (1, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTAP, MOP,
     &                SIG2P, MCHATP, MUP, SIGP)

C   USE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGE ON Mp

          CALL NORRNG (1, MPNTP, MZEROP, MCPNTP, MCP, MCHATP, RANGMP)

C   WRITE RESULTS TO FILE

          WRITE(7,975) MCHATP(1,1)
          WRITE(7,935)
          WRITE(7,940) 'mp', RANGMP(1,1), RANGMP(2,1)
          WRITE(7,965) MUP(1), SIGP(1)

C   BEGIN INFORMATION AGGREGATION PROCESSING FOR THE ESTIMATED ELASTIC
C   STRAIN DATA

C   CALCULATE RESIDUAL VARIANCES

          CALL SW2SU2 (1, NSETS, NP, LNSTRE, LNNF, SX2, SXY, SY2, DD,
     &                 SWHAT2, SUHAT2, NPPR)

C   CALCULATE THE POSTERIOR MEAN AND STANDARD DEVIATION

          CALL MUSIG (1, SX2, DD, SWHAT2, SUHAT2, NPPR, DELTAE, MOE,
     &                SIG2E, MCHATE, MUE, SIGE)

C   USE PRIOR INFORMATION TO OBTAIN POSTERIOR RANGE ON Me

          CALL NORRNG (1, MPNTE, MZEROE, MCPNTE, MCE, MCHATE, RANGME)

C   WRITE RESULTS TO FILE
```

```
         WRITE(7,980) 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
            NP(J,2) = NP(J,1)
  100    CONTINUE
         MCHAT(2,1) = MCHATP(2,1)
         MCHAT(2,2) = MCHATE(2,1)

         CALL GTPVAR (NSETS, NP, 2, LAMN, MCHAT, PVAR)

         WRITE(7,990) 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 STRAIN DECOMPOSITION AND INFORMATION ',
     &       'AGGREGATION CALCULATIONS',////,2X,
     &       '95% CONFIDENCE INTERVAL AND POINT ESTIMATE OF mp',
     &       /,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)
```

```
 980 FORMAT(///,2X,'POINT ESTIMATE OF me FOR ESTIMATED ELASTIC ',
    &        'COMPONENTS',//,7X,'me = ',F9.6)

 985 FORMAT(/,2X,'TOTAL STRAIN E(BETAo) = ',F7.4)

 990 FORMAT(/,2X,'THE EXTENT OF DEPARTURE FROM THE MULTIPLE HEAT ',
    &        'MEDIAN S/N CURVE',/,2X,'WARRANTED BY THE AVAILABLE ',
    &        'INFORMATION',//,7X,E11.5)

      RETURN
      END


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


C   SUBROUTINE INITD PERFORMS THE INITIALIZATION ON THE PRIMARY ARRAYS
C   USED IN THE STRAIN FORMULATION OF THE INFORMATION AGGREGATION
C   MODEL CALCULATIONS
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  21JUN88      COMMENTS:  13JUL89
C      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE INITD (NPTS, RAWNF, RAWSTR, NDC, SP, SE, NTENS,
     &                  TNSILE, NP, LNNF, LNSTRP, LNSTRE, STRP, STRE,
     &                  MZEROP, MZEROE, MPNTP, MPNTE, MCP, MCE, MCPNTP,
     &                  MCPNTE, RFNP, NF, RFSTRE, RFSTRP, NBND, DELTAP,
     &                  DELTAE, MOP, MOE, SIG2P, SIG2E, MCHAT)

C   INPUTS:    ---
C   OUTPUTS:   NPTS, RAWNF, RAWSTR, NDC, SP, SE, NTENS, TNSILE,
C              NP, LNNF, LNSTRP, LNSTRE, STRP, STRE, MZEROP, MZEROE,
C              MPNTP, MPNTE, MCP, MCE, MCPNTP, MCPNTE, RFNP, NF,
C              RFSTRE, RFSTRP, NBND, DELTAP, DELTAE, MOP, MOE, SIG2P,
C              SIG2E, MCHAT

C     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),
     &         NP(0:MAXSET, MAXREG), NPTS(0:MAXSET), NTENS(0:MAXSET),
     &         RFNP(MAXREG)

      REAL     DELTAE(MAXREG), DELTAP(MAXREG),
     &         LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &         LNSTRE(MAXDAT, 0:MAXSET, MAXREG),
     &         LNSTRP(MAXDAT, 0:MAXSET, MAXREG), MCE(2, MAXREG),
     &         MCHAT(2, MAXREG), MCP(2, MAXREG), MOE(MAXREG),
     &         MOP(MAXREG), MZEROE(2, MAXREG), MZEROP(2, MAXREG),
     &         NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         RAWNF(MAXDAT, 0:MAXSET), RAWSTR(MAXDAT, 0:MAXSET),
     &         RFSTRE(MAXDAT, MAXREG), RFSTRP(MAXDAT, MAXREG),
     &         SE(MAXDAT, 0:MAXSET), SIG2E(MAXREG), SIG2P(MAXREG),
     &         SP(MAXDAT, 0:MAXSET), STRE(MAXDAT, 0:MAXSET),
     &         STRP(MAXDAT, 0:MAXSET), TNSILE(0:MAXSET, MAXTNS)

C                        LIST OF VARIABLES
C
C   DELTAE()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MUE() AND
C                 SIGE() CALCULATION
C   DELTAP()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED IN MUP() AND
C                 SIGP() CALCULATION
C   I           CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
```

```
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())
C  LNSTRE()     3-D ARRAY CONTAINING LN(SE()) OR LN(STRE())
C  LNSTRP()     3-D ARRAY CONTAINING LN(SP()) OR LN(STRP())
C  M            CONTROLS DO LOOP FOR EACH TENSILE TEST POINT IN A DATA SET
C  MAXDAT       MAXIMUM NUMBER OF POINTS IN 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  MAXTNS       MAXIMUM NUMBER OF TENSILE TESTS PER DATA SET ALLOWED
C  MCE()        2-D ARRAY CONTAINING VALUES OF THE RANGE ON Me CONSISTENT
C                 WITH GIVEN VALUE OF Co AND THE DATA -- NOTE:  THE Co
C                 CONSTRAINT IS NOT APPLICABLE TO THE STRAIN FORMULATION,
C                 BUT IS A NECESSARY PARAMETER TO SOME OF THE SUBROUTINES
C                 SHARED WITH THE STRESS CASE
C  MCHAT()      2-D ARRAY CONTAINING VALUES OF THE ESTIMATES OF Cp AND Ce,
C                 BASED ON MATERIALS DATA ONLY -- MCHAT(2,1) = MCHATP(2,1)
C                 AND MCHAT(2,2) = MCHATE(2,1)
C  MCP()        2-D ARRAY CONTAINING VALUES OF THE RANGE ON Mp CONSISTENT
C                 WITH GIVEN VALUE OF Co AND THE DATA -- NOTE:  THE Co
C                 CONSTRAINT IS NOT APPLICABLE TO THE STRAIN FORMULATION,
C                 BUT IS A NECESSARY PARAMETER TO SOME OF THE SUBROUTINES
C                 SHARED WITH THE STRESS CASE
C  MCPNTE()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MCE() (ALWAYS 0 FOR STRAIN CASE)
C  MCPNTP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MCP() (ALWAYS 0 FOR STRAIN CASE)
C  MOE()        1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
C                 ELASTIC COMPONENTS
C  MOP()        1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
C                 PLASTIC COMPONENTS
C  MPNTE()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZEROE()
C  MPNTP()      1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                 MZEROP()
C  MZEROE()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Me FOR
C                 ELASTIC COMPONENTS -- MZEROE(1,1) IS THE LOWER BOUND AND
C                 MZEROE(2,1) IS THE UPPER BOUND
C  MZEROP()     2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Mp FOR
C                 PLASTIC COMPONENTS -- MZEROP(1,1) IS THE LOWER BOUND AND
C                 MZEROP(2,1) IS THE UPPER BOUND
C  NBND()       1-D ARRAY CONTAINING UPPER BOUND (CYCLES) FOR THE NUMREG
C                 REGIONS OF INTEREST
C  NDC()        1-D ARRAY CONTAINING Number of given DeComposed STRAIN
C                 POINTS FOR EACH DATA SET
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
C                 SET IN EACH REGION
C  NPTS()       1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NTENS()      1-D ARRAY CONTAINING NUMBER OF TENSILE TEST DATA POINTS
C                 FOR EACH DATA SET
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) OR RAW TOTAL
C                 STRAIN DATA (%) FOR ALL S/N DATA SETS
C  RFNP()       1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                 (REFERENCE) MATERIAL S/N DATA SET
C  RFSTRE()     2-D ARRAY CONTAINING ELASTIC STRAIN POINTS (%), SE() OR
C                 STRE(), FOR THE SPECIFIC MATERIAL S/N DATA SET
C  RFSTRP()     2-D ARRAY CONTAINING PLASTIC STRAIN POINTS (%), SP() OR
C                 STRP(), FOR THE SPECIFIC MATERIAL S/N DATA SET
C  SE()         2-D ARRAY CONTAINING GIVEN ELASTIC STRAIN COMPONENTS (%)
C  SIG2E()      1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE
C                 FOR ELASTIC COMPONENTS
C  SIG2P()      1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE
C                 FOR PLASTIC COMPONENTS
C  SP()         2-D ARRAY CONTAINING GIVEN PLASTIC STRAIN COMPONENTS (%)
C  STRE()       2-D ARRAY CONTAINING CALCULATED ELASTIC STRAIN COMPONENTS (%)
C  STRP()       2-D ARRAY CONTAINING CALCULATED PLASTIC STRAIN COMPONENTS (%)
C  TNSILE()     2-D ARRAY CONTAINING TENSILE TEST DATA -- INDEXED FOR EACH
C                 DATA SET
```

```
      DO 100 J = 0, MAXSET
         NDC(J) = 0
         NPTS(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
            RAWNF(I,J) = 0.0
            RAWSTR(I,J) = 0.0
            STRE(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 = 1, MAXDAT
               LNNF(K,J,L) = 0.0
               LNSTRE(K,J,L) = 0.0
               LNSTRP(K,J,L) = 0.0
  350       CONTINUE
  325    CONTINUE
  300 CONTINUE

      DO 400 L = 1, MAXREG
         DO 450 J = 0, MAXSET
            NP(J,L) = 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
         MPNTP(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
            RFSTRE(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
```

```
  700 CONTINUE

      RETURN
      END

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


C  SUBROUTINE RDECHO READS THE DATA FROM SPECFD AND RELATD AND ECHOES
C  THE DATA TO SPECFO AND RELATO
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  21JUN88     FORMAT/COMMENTS:  15SEP89
C     VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      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,
C            NTENS, TNSILE, MZEROP, MPNTP, MZEROE, MPNTE, DELTAP,
C            DELTAE, MOP, MOE, SIG2P, SIG2E, KRATIO, LAMN
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER  MAXDAT, MAXREG, MAXSET, MAXTNS

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

      COMMON   IOUT

      INTEGER  I, 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), MZEROE(2, MAXREG),
     &         MZEROP(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)


C                     LIST OF VARIABLES
C
C  DELTAE()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED MUE() AND
C                SIGE() CALCULATION
C  DELTAP()    1-D ARRAY CONTAINING BAYESIAN MULTIPLIER USED MUP() AND
C                SIGP() CALCULATION
C  DESCRP()    1-D ARRAY CONTAINING DESCRIPTIONS OF EACH DATA SET
C  I           CONTROLS DO LOOP FOR EACH POINT IN A DATA SET
C  IOUT        OUTPUT DUMP CONTROLLER
C  J           CONTROLS DO LOOP FOR EACH DATA SET
C  KRATIO      RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  LAMN        LAMBDA-N -- RATIO OF Var (Ln N given S) / (m**2 C**2),
C                CONSTANT OVER REGIONS AND COMPONENTS
C  M           CONTROLS DO LOOP FOR EACH TENSILE POINT IN A DATA SET
C  MAXDAT      MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG      MAXIMUM NUMBER OF REGIONS ALLOWED
C  MAXSET      MAXIMUM NUMBER OF S/N DATA SET ALLOWED
C  MAXTNS      MAXIMUM NUMBER OF TENSILE TESTS PER DATA SET ALLOWED
C  MOE()       1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
C                ELASTIC COMPONENTS
C  MOP()       1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION MEAN FOR
C                PLASTIC COMPONENTS
```

```
C  MPNTE()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MCE() (ALWAYS 0 FOR STRAIN CASE)
C  MPNTP()     1-D ARRAY CONTAINING THE NUMBER OF POINTS, 0, 1, OR 2, IN
C                MCP() (ALWAYS 0 FOR STRAIN CASE)
C  MPROC       Materials PROCess variation -- CONTROLS MATERIALS PROCESS
C                VARIATION -- 0 - NO VARIATION;  1 - VARIATION
C  MZEROE()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Me FOR
C                ELASTIC COMPONENTS -- MZEROE(1,1) IS THE LOWER BOUND AND
C                MZEROE(2,1) IS THE UPPER BOUND
C  MZEROP()    2-D ARRAY CONTAINING VALUES OF THE PRIOR RANGE ON Mp FOR
C                PLASTIC COMPONENTS -- MZEROP(1,1) IS THE LOWER BOUND AND
C                MZEROP(2,1) IS THE UPPER BOUND
C  NDC()       1-D ARRAY CONTAINING Number of given DeComposed STRAIN
C                POINTS FOR EACH DATA SET
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NTENS()     1-D ARRAY CONTAINING NUMBER OF TENSILE TEST DATA POINTS
C                FOR EACH DATA SET
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) OR RAW TOTAL
C                STRAIN (%) DATA FOR ALL S/N DATA SETS
C  SE()        2-D ARRAY CONTAINING GIVEN ELASTIC STRAIN COMPONENTS (%)
C  SIG2E()     1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE
C                FOR ELASTIC COMPONENTS
C  SIG2P()     1-D ARRAY CONTAINING THE PRIOR NORMAL DISTRIBUTION VARIANCE
C                FOR PLASTIC COMPONENTS
C  SP()        2-D ARRAY CONTAINING GIVEN PLASTIC STRAIN COMPONENTS (%)
C  TNSILE()    2-D ARRAY CONTAINING TENSILE TEST DATA -- INDEXED FOR EACH
C                DATA SET
C  VARY        CONTROLS TYPE OF CURVE VARIATION DESIRED -- 0 - NO
C                VARIATION;  1 - S/N RANDOMNESS ONLY;  2 - UNIFORM
C                VARIATION;  3 - TRUNCATED NORMAL VARIATION

C     READ DECOMPOSED STRAIN DATA FOR THE SPECIFIC MATERIAL DATA
C     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. 10) 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. 10) WRITE(8,920) RAWSTR(I,0), RAWNF(I,0),
     &      SP(I,0), SE(I,0)
   50 CONTINUE

C     READ REMAINING TOTAL STRAIN DATA FOR THE SPECIFIC MATERIAL
C     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), MZEROP(2,1)
      WRITE(3,960) MPNTP(1), MZEROP(1,1), MZEROP(2,1)
      READ(1,*) MPNTE(1), MZEROE(1,1), MZEROE(2,1)
      WRITE(3,965) MPNTE(1), MZEROE(1,1), MZEROE(2,1)

      IF (IOUT .EQ. 10) THEN
         WRITE(8,960) MPNTP(1), MZEROP(1,1), MZEROP(2,1)
         WRITE(8,965) MPNTE(1), MZEROE(1,1), MZEROE(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

      IF (VARY .EQ. 3) THEN
C        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
C     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
            CALL TRMNAT
         ENDIF

         WRITE(6,950) DESCRP(J), NDC(J), NPTS(J), NTENS(J)
         WRITE(6,910)

         IF (IOUT .EQ. 10) THEN
            WRITE(8,950) DESCRP(J), NDC(J), NPTS(J), NTENS(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)
            WRITE(6,920) 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),
     &         SP(I,J), SE(I,J)
125      CONTINUE

         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)
150      CONTINUE

         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)
175      CONTINUE

100   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)
```

```
960 FORMAT(////,2X,'EXOGENOUS INFORMATION',
     &         ///,2X,'EXPLICIT CONSTRAINT ON mp',
     &         /,9X,'NUMBER OF POINTS IN RANGE:',2X,I1,
     &         //,15X,'LOWER BOUND',5X,'UPPER BOUND',
     &         //,16X,F7.4,10X,F7.4)

965 FORMAT(///,2X,'EXPLICIT CONSTRAINT ON me',
     &         /,9X,'NUMBER OF POINTS IN RANGE:',2X,I1,
     &         //,15X,'LOWER BOUND',5X,'UPPER BOUND',
     &         //,16X,F7.4,10X,F7.4)

970 FORMAT(///,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS -- mp',
     &         //,12X,'DELTA',8X,'mo',10X,'SIGMA2',
     &         //,10X,F7.2,5X,F7.4,5X,E11.5)

975 FORMAT(///,2X,'PRIOR NORMAL DISTRIBUTION PARAMETERS -- me',
     &         //,12X,'DELTA',8X,'mo',10X,'SIGMA2',
     &         //,10X,F7.2,5X,F7.4,5X,E11.5)

980 FORMAT(//,2X,'TENSILE DATA',/)

990 FORMAT(10X,F8.5)

995 FORMAT(//,2X,'K*/K:',8X,E11.5,/,2X,'LAMBDAN:',5X,E11.5)

      RETURN
      END


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


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,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE PREP (NSETS, NUM, PLAS, ELAS, RAWNF, NP, LNSTRP,
     &                 LNSTRE, LNNF, RFNP, RFSTRP, RFSTRE, NF)

C  INPUTS:   NSETS, NUM, PLAS, ELAS, RAWNF
C  OUTPUTS:  NP, LNSTRP, LNSTRE, LNNF, RFNP, RFSTRP, RFSTRE, NF

C     IMPLICIT NONE

      INTEGER  MAXDAT, MAXREG, MAXSET

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

      COMMON   IOUT

      INTEGER  IOUT, J, K, NP(0:MAXSET, MAXREG), NSETS,
     &         NUM(0:MAXSET), RFNP(MAXREG)

      REAL     ELAS(MAXDAT, 0:MAXSET), LNNF(MAXDAT, 0:MAXSET, MAXREG),
     &         LNSTRE(MAXDAT, 0:MAXSET, MAXREG),
     &         LNSTRP(MAXDAT, 0:MAXSET, MAXREG),
     &         NF(MAXDAT, MAXREG), PLAS(MAXDAT, 0:MAXSET),
     &         RAWNF(MAXDAT, 0:MAXSET), RFSTRE(MAXDAT, MAXREG),
     &         RFSTRP(MAXDAT, MAXREG)


C                 LIST OF VARIABLES
C
C  ELAS()       2-D ARRAY CONTAINING ELASTIC STRAIN COMPONENTS (%)
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   LNNF()       3-D ARRAY CONTAINING LN(RAWNF())
C   LNSTRE()     3-D ARRAY CONTAINING LN(SE()) OR LN(STRE())
C   LNSTRP()     3-D ARRAY CONTAINING LN(SP()) OR LN(STRP())
C   MAXDAT       MAXIMUM NUMBER OF POINTS IN 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   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
C                  SET IN EACH REGION
C   NSETS        NUMBER OF RELATED MATERIAL S/N DATA SETS
C   NUM()        1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C   PLAS()       2-D ARRAY CONTAINING PLASTIC STRAIN COMPONENTS (%)
C   RAWNF()      2-D ARRAY CONTAINING RAW CYCLES TO FAILURE DATA FOR ALL S/N
C                  DATA SETS
C   RFNP()       1-D ARRAY CONTAINING THE NUMBER OF POINTS FOR THE SPECIFIC
C                  (REFERENCE) MATERIAL S/N DATA SET
C   RFSTRE()     2-D ARRAY CONTAINING ELASTIC STRAIN POINTS (%), SE() OR
C                  STRE(), FOR THE SPECIFIC MATERIAL S/N DATA SET
C   RFSTRP()     2-D ARRAY CONTAINING PLASTIC STRAIN POINTS (%), SP() OR
C                  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(RAWNF(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,*) 'RAWNF = ',RAWNF(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)
```

```
      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,*) 'ELAS = ', ELAS(K,0), ' RFSTRE = ', RFSTRE(K,1)
            WRITE(8,*) 'RAWNF = ', RAWNF(K,0), ' NF = ', NF(K,1)
         ENDIF
  300 CONTINUE

      RETURN
      END


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


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               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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             TNSILE, KP, KE, MP, ME
C  OUTPUTS:   STRP, STRE
C  SUBPROGRAMS:  TRMNAT

C     IMPLICIT NONE

      INTEGER  MAXDAT, MAXSET, MAXTNS

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

      COMMON   IOUT

      INTEGER  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,
     &         RAWNF(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)


C                  LIST OF VARIABLES
C
C  ALPHA()     2-D ARRAY CONTAINING THE RATIOS OF Sp TO Se FOR EACH
C                 TOTAL STRAIN POINT
C  DESCRP()    1-D ARRAY CONTAINING DESCRIPTIONS OF EACH S/N DATA SET
C  I           CONTROLS DO LOOP FOR EACH POINT IN A DATA SET
C  IOUT        OUTPUT DUMP CONTROLLER
C  J           CONTROLS DO LOOP FOR EACH DATA SET
C  KE          S-N CURVE LOCATION PARAMETER, Ke, FOR THE ELASTIC STRAIN
C                 S-N CURVE
C  KP          S-N CURVE LOCATION PARAMETER, Kp, FOR THE PLASTIC STRAIN
C                 S-N CURVE
C  KTERM       EQUAL TO KP / KE
```

```
C  LAMBDA      2-D ARRAY CONTAINING VALUES FOR THE PLASTIC FRACTION OF
C                THE TOTAL STRAIN
C  M           CONTROLS DO LOOP FOR EACH TENSILE TEST DATA POINT
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  MAXTNS      MAXIMUM NUMBER OF TENSILE TESTS PER DATA SET ALLOWED
C  ME          S/N CURVE SHAPE PARAMETER, Me, FOR THE ELASTIC STRAIN
C                S/N CURVE
C  MP          S/N CURVE SHAPE PARAMETER, Mp, FOR THE PLASTIC STRAIN
C                S/N CURVE
C  MTERM       EQUAL TO  -[(1 / Mp) - (1 / Me)]
C  NDC         1-D ARRAY CONTAINING Number of given DeComposed STRAIN
C                POINTS FOR EACH DATA SET
C  NPTS()      1-D ARRAY CONTAINING NUMBER OF POINTS IN S/N DATA SETS
C  NSETS       NUMBER OF RELATED MATERIAL S/N DATA SETS
C  NTENS()     1-D ARRAY CONTAINING NUMBER OF TENSILE TEST DATA POINTS
C                FOR EACH DATA SET
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) OR RAW TOTAL
C                STRAIN DATA (%) FOR ALL S/N DATA SET
C  SE()        2-D ARRAY CONTAINING GIVEN ELASTIC STRAIN COMPONENTS (%)
C  SP()        2-D ARRAY CONTAINING GIVEN PLASTIC STRAIN COMPONENTS (%)
C  STRE()      2-D ARRAY CONTAINING CALCULATED ELASTIC STRAIN COMPONENTS (%)
C  STRP()      2-D ARRAY CONTAINING CALCULATED PLASTIC STRAIN COMPONENTS (%)
C  TNSILE()    2-D ARRAY CONTAINING TENSILE TEST DATA -- INDEXED FOR EACH
C                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)
     &      WRITE(8,*) 'J = ', J, ' NDC = ', NDC(J),
     &                 ' NPTS = ', NPTS(J), ' NTENS = ', NTENS(J)


C        IF GIVEN COMPONENT DATA EXITS, NO DECOMPOSTION IS REQUIRED

         DO 150 I = 1, NDC(J)
            STRP(I,J) = SP(I,J)
            STRE(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 = ', STRE(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)
```

```
          STRE(I,J) = RAWSTR(I,J) - STRP(I,J)
          IF (IOUT .EQ. 10) THEN
             WRITE(8,*) 'I = ', I, ' RAWSTR = ', RAWSTR(I,J),
     &                             ' RAWNF = ',  RAWNF(I,J)
             WRITE(8,*) 'ALPHA = ', ALPHA(I,J),
     &                  ' LAMBDA = ', LAMBDA(I,J)
             WRITE(8,*) 'STRP = ', STRP(I,J), ' STRE = ', STRE(I,J)
          ENDIF
  200    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)
               WRITE(8,*) 'STRP = ', STRP(I,J), ' STRE = ', STRE(I,J)
               WRITE(8,*) '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(0), NPTS(0)

      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,/)

      RETURN
      END
```

```
C***************************************************************************

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

C     IMPLICIT NONE

      INTEGER MAXDAT, MAXREG

      PARAMETER (MAXDAT = 50, MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NP, NPTS(MAXREG), NUMREG, VARY, ZROREG

      REAL     BIGK(0:MAXREG), BZERO, K, MEANZ, MM(0:MAXREG),
     &         MU(MAXREG), NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &         RANGEM(2, MAXREG), SBND(0:MAXREG), SIG(MAXREG),
     &         STR(MAXDAT, MAXREG), SZ2, ZZ(MAXDAT)

      DOUBLE PRECISION RAND


C                       LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C              EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING S/N DATA SET
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C              DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXDAT    MAXIMUM NUMBER OF POINTS IN S/N DATA SET (PER REGION) ALLOWED
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MEANZ     SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()      1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL
C              DISTRIBUTION MEAN FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NF()      2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C              SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP        TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()    1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C              SPECIFIC MATERIAL S/N DATA SET
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  RAND      RANDOM NUMBER SEED
C  RANGEM()  2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C              FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C              RANGEM(2,L) IS THE UPPER BOUND
C  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()
C  SIG()     1-D ARRAY CONTAINING VALUES OF THE POSTERIOR NORMAL DISTRIBUTION
```

```
C                 STANDARD DEVIATION FOR EACH REGION
C   STR()     2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL S/N
C                 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;
C                 1 - S-N RANDOMNESS ONLY;  2 - UNIFORM VARIATION;
C                 3 - TRUNCATED NORMAL VARIATION
C   ZROREG    ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C                 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

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, 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

      WRITE(7,900) NUMREG, BZERO

      DO 200 L = ZROREG, NUMREG
         WRITE(7,910) L, MM(L), BIGK(L), NBND(L), SBND(L)
  200 CONTINUE

      WRITE(7,920)

C   FORMAT STATEMENTS

  900 FORMAT(///,2X,'SELECTED VALUES OF S/N CURVE PARAMETERS',
     &        //,2X,'NUMBER OF REGIONS: ',I4,5X,'BETAo = ',F8.4,
     &        //,2X,'REGION',7X,'m',15X,'K',9X,'LIFE BOUND',5X,
     &        'STRESS BOUND',/)

  910 FORMAT(5X,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. 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

      SUBROUTINE FINDM (RAND, NUMREG, RANGEM, MM)

C  INPUTS:    RAND, NUMREG, RANGEM
C  OUTPUTS:   MM
C  SUBPROGRAMS:  RANDOM, TRMNAT

C     IMPLICIT NONE

      INTEGER MAXREG

      PARAMETER (MAXREG = 3)

      COMMON  IOUT

      INTEGER IOUT, L, NUMREG

      REAL    MM(0:MAXREG), PICK(2), RANGEM(2, MAXREG), X

      DOUBLE PRECISION RAND

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

```
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              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),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF

  100 CONTINUE

      RETURN
      END


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


C*************************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C*************************************************************************

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

C                LIST OF VARIABLES
C
C  FRAC       UNIFORM (0,1) RANDOM VARIATE
C  IOUT       OUTPUT DUMP CONTROLLER
C  RANA       CONSTANT FOR LCG
C  RANC       CONSTANT FOR LCG
C  RAND       RANDOM NUMBER SEED
C  RANDIV     INTERNAL CALCULATION
C  RANM       CONSTANT FOR LCG
C  RANSUB     INTERNAL CALCULATION
C  RANT       INTERNAL CALCULATION
C  RANX       INTERNAL CALCULATION
```

```
C          USING LCG RANDOM # GENERATOR

      RANA = 671093.0
      RANC = 7090885.0
      RANM = 33554432.0

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

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

      RETURN
      END

C     NOTES:  IOUT=2 DUMPS TO SCREEN


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


C  SUBROUTINE FINDMN CALCULATES THE VALUE OF M FOR EACH REGION BY
C  SAMPLING OFF THE APPROPRIATE TRUNCATED NORMAL M DISTRIBUTION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  7JUN88      COMMENTS:  13FEB89
C     VERSION:  MATCHR V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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


C                     LIST OF VARIABLES
C
C  IOUT         OUTPUT DUMP CONTROLLER
C  L            CONTROLS DO LOOP FOR EACH REGION
C  MAXREG       MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()         1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  MU()        1-D ARRAY CONTAINING THE MEAN OF M FOR EACH REGION
C  NUMREG       NUMBER OF REGIONS OF INTEREST
C  PICK()       1-D ARRAY CONTAINING ADJUSTED RANGE ON M TO BE SAMPLED FROM
C  RAND         RANDOM NUMBER SEED
C  RANGEM()     2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGES ON M
C                 FOR EACH REGION -- RANGEM(1,L) IS THE LOWER BOUND AND
C                 RANGEM(2,L) IS THE UPPER BOUND
C  SIG()       1-D ARRAY CONTAINING THE STANDARD DEVIATION OF M FOR EACH
```

```
C                REGION
C  X          NORMAL(MU,SIGMA) RANDOM VARIATE USED TO OBTAIN VALUE SAMPLED
C                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
   10       CALL NORMGN (RAND, MU(L), SIG(L), X)
            IF ((X .LT. RANGEM(1,L)) .OR. (X .GT. RANGEM(2,L))) GOTO 10
            MM(L) = X
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'RANGEM(1,L) =', RANGEM(1,L),
     &                    ' RANGEM(2,L) =', RANGEM(2,L)
               WRITE(8,*) 'L =', L, ' X =', X, ' MM(L) =', MM(L)
            ENDIF
         ELSE
C           ADJUST RANGE ACCORDING TO PREVIOUS M VALUE
C           AND THEN SAMPLE
            PICK(1) = AMAX1(MM(L-1), RANGEM(1,L))
            PICK(2) = RANGEM(2,L)
            IF (PICK(1) .GT. PICK(2)) THEN
C              NO RANGE EXISTS -- THIS SHOULD NOT BE POSSIBLE
C              STOP PROGRAM
               WRITE(8,*) 'IMPOSSIBLE M RANGE IN REGION', L
               CALL TRMNAT
            ELSE
C              SAMPLE ON ADJUSTED RANGE
   20          CALL NORMGN (RAND, MU(L), SIG(L), X)
               IF ((X .LT. PICK(1)) .OR. (X .GT. PICK(2))) GOTO 20
               MM(L) = X
            ENDIF
            IF (IOUT .EQ. 10) THEN
               WRITE(8,*) 'L =', L, ' MM(L-1) =', MM(L-1),
     &                    ' RANGEM(1,L) =', RANGEM(1,L)
               WRITE(8,*) 'PICK(1) =', PICK(1), ' PICK(2) =', PICK(2)
               WRITE(8,*) 'RANGEM(2,L) =', RANGEM(2,L), ' X =', X,
     &                    ' MM(L) =', MM(L)
            ENDIF
         ENDIF

  100 CONTINUE

      RETURN
      END


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


C**************************************************************************
C  SUBROUTINE NORMGN GENERATES A NORMALLY DISTRIBUTED RANDOM NUMBER
C  WITH MEAN, MU, AND STANDARD DEVIATION, SIGMA
C  PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C        DATE:  3FEB88
```

```
C     VERSION:  MATCHR V7, V7.1, 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
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C*********************************************************************

      SUBROUTINE NORMGN (RAND, MU, SIGMA, X)

C     SUBPROGRAM:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND

      REAL    FRAC, MU, PI, SIGMA, X, U1, U2, Z1, Z2

      PARAMETER (PI = 3.1415926536)

      INTEGER IOUT

C             LIST OF VARIABLES
C
C  FRAC     UNIFORM(0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  MU       MEAN OF NORMAL DISTRIBUTION
C  RAND     RANDOM NUMBER SEED
C  SIGMA    STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X        NORMAL RANDOM VARIATE
C  U1       UNIFORM RANDOM NUMBER U(0,1)
C  U2       UNIFORM RANDOM NUMBER U(0,1)
C  Z1       NORMAL RANDOM NUMBER ON N(0,1)
C  Z2       NORMAL RANDOM NUMBER ON N(0,1)

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

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

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

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

      X = SIGMA * Z1 + MU
      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 V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5
```

```
      SUBROUTINE TRNSFM (NPTS, STR, NF, NUMREG, MM, NBND, NP, ZZ)
C  INPUTS:   NPTS, STR, NF, NUMREG, MM, NBND
C  OUTPUTS:  NP, ZZ
C     IMPLICIT NONE
      INTEGER MAXDAT, MAXREG
      PARAMETER (MAXDAT = 50, MAXREG = 3)
      COMMON IOUT
      INTEGER I, IOUT, K, L, LL, NP, NPTS(MAXREG), NUMREG
      REAL    MM(0:MAXREG), MML, NBND(0:MAXREG), NF(MAXDAT, MAXREG),
     &        STR(MAXDAT, MAXREG), ZZ(MAXDAT)

C                     LIST OF VARIABLES
C
C  I          CONTROLS DO LOOP FOR EACH DATA POINT
C  IOUT       OUTPUT DUMP CONTROLLER
C  K          CONTROLS DO LOOP FOR EACH DATA POINT IN EACH REGION
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LL         CONTROLS INNER DO LOOP FOR EACH REGION
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SAMPLED VALUES OF M FOR EACH REGION
C  MML        EQUAL TO MM(L) FOR A SET OF CALCULATIONS
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NF()       2-D ARRAY CONTAINING RAWNF() (CYCLES TO FAILURE) FOR THE
C               SPECIFIC MATERIAL S/N DATA SET BROKEN INTO REGIONS
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N DATA SET
C  NPTS()     1-D ARRAY CONTAINING THE NUMBER OF POINTS PER REGION FOR THE
C               SPECIFIC MATERIAL S/N DATA SET
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  STR()      2-D ARRAY CONTAINING RATSTR() FOR THE SPECIFIC MATERIAL
C               S-N DATA SET BROKEN INTO REGIONS (PSI OR %)
C  ZZ()       1-D ARRAY CONTAINING TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)

C  INITIALIZE VARIABLES

      NP = 0

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

C  BEGIN CALCULATIONS

      DO 100 L = 1, NUMREG
         MML = MM(L)
         IF (IOUT .EQ. 10) WRITE(8,*)'L =', L, ' MM =', MM(L), ' MML =',
     &         MML, ' NPTS =', NPTS(L)

         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))
     &                  * ((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) =',
     &            MM(LL), ' ZZ =', ZZ(NP)
  300       CONTINUE
```

```
  200    CONTINUE

  100 CONTINUE

      RETURN
      END

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

C  SUBROUTINE SMNVAR CALCULATES THE Sample MeaN and VARiance OF
C  Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  24AUG87     COMMENTS:  13JUL89
C     VERSION:  MATCHR V5.3, V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2,
C                      V8.3, V8.4, V8.5
C               MATGRM V3.3, V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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
C  I          CONTROLS DO LOOP FOR EACH DATA POINT IN A DATA SET
C  IOUT       OUTPUT DUMP CONTROLLER
C  MAXDAT     MAXIMUM NUMBER OF S/N DATA POINTS (PER REGION) ALLOWED
C  MEANZ      SAMPLE MEAN OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  NP         TOTAL NUMBER OF POINTS IN THE SPECIFIC MATERIAL S/N
C               DATA SET
C  SZ2        SAMPLE VARIANCE OF TRANSFORMED DATA, Z = F(STR, NF, NBND, MM)
C  ZZ()       1-D ARRAY CONTAINING THE TRANSFORMED S/N DATA,
C               Z = F(STR,NF,NBND,MM)

C  INITIALIZE VARIABLES

      MEANZ = 0.0
      SZ2 = 0.0

C  CALCULATE THE MEAN OF ZZ(), MEANZ

      DO 100 I = 1, NP
         MEANZ = MEANZ + ZZ(I)
         IF (IOUT .EQ. 10) WRITE(8,*)'NP =', NP, ' I =', I,
     &      ' ZZ =', ZZ(I), ' MEANZ =', MEANZ
  100 CONTINUE
      MEANZ = MEANZ / FLOAT(NP)
      IF (IOUT .EQ. 10) WRITE(8,*)' MEANZ =', MEANZ

C  CALCULATE THE VARIANCE OF ZZ(), SZ2

      DO 200 I = 1, NP
         SZ2 = SZ2 + (ZZ(I) - MEANZ) ** 2
         IF (IOUT .EQ. 10) WRITE(8,*)'I =', I, ' SZ2 =', SZ2
```

```
  200 CONTINUE
      SZ2 = SZ2 / FLOAT(NP - 1)
      IF (IOUT .EQ. 10) WRITE(8,*)' SZ2 =', SZ2

      RETURN
      END


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


C  SUBROUTINE KBETA CALCULATES k AND BETAo FROM THE SAMPLE MEAN AND
C  VARIANCE OF Z = F(STR, NF, NBND, MM)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  6OCT87      COMMENTS:  13JUL89
C     VERSION:  MATCHR V6, V6.1, V6.2, V7, V7.1, V8, V8.1, V8.2, V8.3,
C                      V8.4, V8.5
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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


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


C  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
C               MATGRM V4, V4.1, V4.2, V4.3, V4.4, V4.5

      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)

C                        LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  BZERO     VALUE OF WEIBULL PARAMETER, BETAo, CHARACTERIZING SPECIFIC
C              MATERIAL DATA BASE
C  GAMMA     EULER'S CONSTANT
C  IOUT      OUTPUT DUMP CONTROLLER
C  K         VALUE OF k -- PARAMETER CHARACTERIZING THE SPECIFIC MATERIAL
C              DATA BASE
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST

C  INITIALIZE VARIABLES

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

C  CALCULATE K FOR REGION ONE

      BIGK(1) = (ALOG(2.0) ** (1.0 / BZERO)) * EXP(K + GAMMA / BZERO)
C       WRITE(7,*) 'REGION:  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)))
C          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
C        DATE:  22DEC88
C     VERSION:  MATCHR V8.2, V8.3, V8.4, V8.5
C               MATGRM V4.2, V4.3, V4.4, V4.5

      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)

C              LIST OF VARIABLES
C
C  BIGK()    1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C              FOR EACH REGION
C  IOUT      OUTPUT DUMP CONTROLLER
C  L         CONTROLS DO LOOP FOR EACH REGION
C  MAXREG    MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()      1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()    1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C              REGIONS OF INTEREST
C  NUMREG    NUMBER OF REGIONS OF INTEREST
C  SBND()    1-D ARRAY CONTAINING STRESS VALUES (PSI, R = -1.0)
C              CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH
C              REGION CONTAINED IN NBND()
C  ZROREG    ZeRO REGion -- VALUES CHOSEN TO FACILITATE REGION DO LOOP
C              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   SUBROUTINE ADJSTM CONTROLS THE CALCULATIONS NECESSARY TO CONSTRAIN
C   Me TO BE GREATER THAN THE SELECTED Mp BY ADJUSTING THE POSTERIOR
C   RANGE ON Me
C   PROGRAMMER:  L. NEWLIN
C         DATE:  7JUN88
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
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      SUBROUTINE ADJSTM (MP, RANGME, MEBND)

C   INPUTS:    MP, RANGME
C   OUTPUTS:   MEBND

C     IMPLICIT NONE

      INTEGER   MAXREG

      PARAMETER (MAXREG = 3)

      COMMON    IOUT

      INTEGER   IOUT, L

      REAL      MEBND(2, MAXREG), MP(0:MAXREG), RANGME(2, MAXREG)


C                     LIST OF VARIABLES
C
C   IOUT       OUTPUT DUMP CONTROLLER
C   L          CONTROLS DO LOOP FOR EACH POINT IN A REGION
C   MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C   MEBND()    2-D ARRAY CONTAINING ALLOWABLE RANGE ON Me, FOR ELASTIC
C                COMPONENTS, AFTER SELECTION OF Mp
C   MP()       1-D ARRAY CONTAINING SELECTED VALUE OF Mp FOR PLASTIC COMPONENTS
C   RANGME()   2-D ARRAY CONTAINING VALUES OF THE POSTERIOR RANGE ON Me
C                FOR ELASTIC COMPONENTS -- RANGME(1,1) IS THE LOWER
C                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*************************************************************************


C   THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
```

```
C  MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C  TRANSFORM METHOD"
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  18MAR87      COMMENTS:  15SEP89
C     VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                      V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C               MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                      V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE WEIBGN (BETA, RAND, WEIB)

C  INPUTS:    BETA, RAND
C  OUTPUTS:   WEIB
C  SUBPROGRAMS:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER IOUT

      REAL    ARG, BETA, ETA, FRAC, WEIB

      DOUBLE PRECISION RAND


C                 LIST OF VARIABLES
C
C  ARG     INTERMEDIATE CALCULATION VARIABLE
C  BETA    WEIBULL DISTRIBUTION SHAPE PARAMETER
C  ETA     WEIBULL DISTRIBUTION LOCATION PARAMETER
C  FRAC    UNIFORM (0,1) RANDOM VARIATE
C  IOUT    OUTPUT DUMP CONTROLLER
C  RAND    RANDOM NUMBER SEED
C  WEIB    WEIBULL(BETA,ETA) GENERATED RANDOM VARIATE


C     CALCULATE CONSTRAINED ETA

      ETA = 1.0 / (ALOG(2.0) ** (1.0 / BETA))

C     GENERATE WEIBULL RANDOM VARIATE

      CALL RANDOM(FRAC, RAND)
      ARG = -ALOG(1.0 - FRAC)
      WEIB = ETA * ARG**(1.0/BETA)
      IF (IOUT .EQ. 10) WRITE(8,*)'BETA = ', BETA, ' ETA =', ETA,
     &     ' FRAC =', FRAC, ' ARG =', ARG, ' WEIB =', WEIB

      RETURN
      END


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


C  SUBROUTINE KOMO CALCULATES Ko AND Mo FOR THE ZERO REGION (NO DATA
C  REGION TO THE LEFT). IT ACCOUNTS FOR TYING UP THE TENSILE POINT
C  AT SZERO, AND SCALING DOWN THE CURVE IF IT WENT ABOVE SZERO.
C  PROGRAMMER :  L. NEWLIN
C         DATE:  1AUG91
C      VERSION:  MATCHR V8.5   MATGRM V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
```

```
      SUBROUTINE KOMO (SZERO, BIGK, MM, NBND, TRSBND, TRBIGK,
     &                 FACTR, NUMREG)

C  INPUTS:   SZERO, BIGK, MM, NBND, TRSBND, FACTR
C  OUTPUTS:  TRBIGK, MM, TRSBND

C     IMPLICIT NONE

      INTEGER  MAXREG

      PARAMETER (MAXREG = 3)

      COMMON   IOUT

      INTEGER  IOUT, L, NUMREG

      REAL     BIGK(0:MAXREG), FACTR, MM(0:MAXREG), NBND(0:MAXREG),
     1         SCLK, SZERO, TRBIGK(0:MAXREG), TRSBND(0:MAXREG)

C                    LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M FOR
C               EACH REGION
C  FACTR      SCALE FACTOR = PHI * KRATIO * Z
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  MM()       1-D ARRAY CONTAINING SELECTED VALUES OF M FOR EACH REGION
C  NBND()     1-D ARRAY CONTAINING UPPER BOUNDS (CYCLES) FOR THE NUMREG
C               REGIONS OF INTEREST
C  NUMREG     NUMBER OF REGIONS
C  SCLK       ADJUSTMENT FACTOR FOR BIGK IF TRSBND(0) > SZERO
C  SZERO      STRESS TENSILE TEST POINT, So
C  TRBIGK()   1-D ARRAY CONTAINING VALUES OF K, ADJUSTED TO KEEP
C               SBND(0) < So FOR EACH TRIAL
C  TRSBND()   1-D ARRAY CONTAINING STRESS VALUES CORRESPONDING TO THE
C               LIFE BOUNDARY VALUES FOR EACH REGION CONTAINED IN NBND()
C               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, ' BIGK1 = ', TRBIGK(1)
         WRITE(8,*) 'MM1 = ', MM(1), ' MMo = ', MM(0)
      ENDIF

      RETURN
      END

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

C  FUNCTION GTLIFE CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR STRESS
C  BASED UPON THE MATERIALS CHARACTERIZATION S/N EQUATION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB89
```

```
C     VERSION:  MATCHR V8.3, V8.4, V8.5   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.

      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  OUTPUTS:  GTLIFE

C     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


C                       LIST OF VARIABLES
C
C  BIGK()     1-D ARRAY CONTAINING VALUES OF K, WHERE A = K ** M
C               FOR EACH REGION
C  GETLIF     VALUE TO BE ASSIGNED TO GTLIFE -- CYCLES TO FAILURE FOR
C               THE REQUIRED STRESS LEVEL
C  IOUT       OUTPUT DUMP CONTROLLER
C  KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  L          CONTROLS DO LOOP FOR EACH REGION
C  LNZ        NORMAL(0,PVAR) GENERATED RANDOM VARIATE
C  MAXREG     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               CORRESPONDING TO THE "LIFE BOUNDARY" VALUES FOR EACH REGION
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
               TEMP = MM(L) * (ALOG(BIGK(L)) - 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

      GTLIFE = GETLIF
```

```
      RETURN
      END

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

C  FUNCTION GTLIF2 CALCULATES THE CYCLES TO FAILURE FOR A PARTICULAR
C  VALUE OF TOTAL STRAIN BASED UPON THE MATERIALS CHARACTERIZATION
C  COMPONENT STRAIN S/N EQUATION.  THE SOLUTION IS FOUND USING A
C  NEWTON'S METHOD ITERATION SUBROUTINE AND THE LIFE CORRESPONDING
C  TO THE ELASTIC STRAIN IS USED AS THE INITIAL LIFE VALUE, No.
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  15FEB89      COMMENTS:  13JUL89
C     VERSION:  MATCHR V8.3, V8.4, V8.5   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.

      REAL FUNCTION GTLIF2 (STR, KP, KE, MP, ME, PHI, KRATIO, Z)

C  INPUTS:    STR, KP, KE, MP, ME, PHI, KRATIO, Z
C  OUTPUTS:   GTLIF2
C  SUBPROGRAMS:  NEWTON, TRMNAT

C     IMPLICIT NONE

      COMMON   IOUT

      INTEGER  IEND, IER, IOUT

      REAL     DERF, EPS, F, GETLIF, KE, KP, KRATIO, ME, MP, NZERO,
     &         PHI, STR, Z


C                       LIST OF VARIABLES
C
C  DERF       RESULTANT VALUE OF DERIVATIVE AT ROOT
C  EPS        INPUT VALUE WHICH SPECIFIES THE UPPER BOUND OF THE ERROR OF
C               THE RESULT
C  F          RESULTANT FUNCTION VALUE AT ROOT
C  GETLIF     VALUE TO BE ASSIGNED TO GTLIF2 -- CYCLES TO FAILURE FOR
C               THE REQUIRED STRAIN LEVEL
C  IEND       MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED
C  IER        RESULTANT ERROR PARAMETER, CODED -- 0 - NO ERROR;  1 - NO
C               CONVERGENCE AFTER IEND ITERATION STEPS;  2 - AT ANY
C               ITERATION STEP DERIVATIVE DERF WAS EQUAL TO ZERO
C  IOUT       OUTPUT DUMP CONTROLLER
C  KE         S/N CURVE LOCATION PARAMETER, Ke, FOR THE ELASTIC STRAIN
C               S/N CURVE
C  KP         S/N CURVE LOCATION PARAMETER, Kp, FOR THE PLASTIC STRAIN
C               S/N CURVE
C  KRATIO     RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C  ME         S/N CURVE SHAPE PARAMETER, Me, FOR THE ELASTIC STRAIN
C               S/N CURVE
C  MP         S/N CURVE SHAPE PARAMETER, Mp, FOR THE PLASTIC STRAIN
C               S/N CURVE
C  NZERO      EQUAL TO THE LIFE FROM THE ELASTIC PORTION OF THE CURVE
C               USED AS THE INITIAL VALUE IN THE NEWTON'S METHOD ITERATION
C  PHI        WEIBULL(BETAo, ETAo) GENERATED RANDOM VARIABLE
C  STR        VALUE OF TOTAL STRAIN (%) FOR WHICH CYCLES TO FAILURE
C               IS DESIRED
C  Z          LOG-NORMAL(0,PVAR) RANDOM VARIATE


      EPS = 1.0E-6
      IEND = 1000

C     CALCULATE INITAL 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
         IF (IOUT .EQ. 10) WRITE(8,*) 'GETLIF =', GETLIF
      ELSEIF (IER .EQ. 1) THEN
         WRITE(8,*)'NO CONVERGENCE AFTER SPECIFIED NO. ITERATION STEPS'
         CALL TRMNAT
      ELSEIF (IER .EQ. 2) THEN
         WRITE(8,*) 'DERIVATIVE EQUAL TO ZERO'
         CALL TRMNAT
      ELSE
         WRITE(8,*) 'ERROR CODE INCORRECTLY SPECIFIED'
         CALL TRMNAT
      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      VERSION:  MATCHR V8.1, V8.2, V8.3, V8.4, V8.5
C                MATGRM V4.1, V4.2, V4.3, V4.4, V4.5

      SUBROUTINE FCT (LIFE, F, DERF, STR, KP, KE, MP, ME, PHI, KRATIO,
     &                Z)

C   INPUTS:   LIFE, STR, KP, KE, MP, 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

C                         LIST OF VARIABLES
C
C   DERF      RESULTANT VALUE OF DERIVATIVE AT ROOT
C   F         RESULTANT FUNCTION VALUE AT ROOT
C   INVME     EQUAL TO 1/Me (INVerse of Me)
C   IMVMP     EQUAL TO 1/Mp (INVerse of Mp)
C   IOUT      OUTPUT DUMP CONTROLLER
C   KE        S/N CURVE LOCATION PARAMETER, Ke, FOR THE ELASTIC STRAIN
C               S/N CURVE
C   KEPROD    EQUAL TO KE * PHI * KRATIO * Z
C   KP        S/N CURVE LOCATION PARAMETER, Kp, FOR THE PLASTIC STRAIN
C               S/N CURVE
C   KPPROD    EQUAL TO KP * PHI * KRATIO * Z
C   KRATIO    RATIO OF K*/K, CONSTANT OVER REGIONS AND COMPONENTS
C   LIFE      VALUE OF N (CYCLES TO FAILURE)
C   ME        S/N CURVE LOCATION PARAMETER, Me, FOR THE ELASTIC STRAIN
C               S/N CURVE
C   MP        S/N CURVE LOCATION PARAMETER, Mp, FOR THE PLASTIC STRAIN
C               S/N CURVE
```

```
C  PHI       WEIBULL(BETAO, ETAO) GENERATED RANDOM VARIABLE
C  STR       VALUE OF TOTAL STRAIN (%) FOR WHICH CYCLES TO FAILURE IS
C              DESIRED
C  Z         LOG-NORMAL(0,PVAR) RANDOM VARIATE


      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


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


C  THE FOLLOWING SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE RTNI
C  TAKEN FROM "MATHEMATICS -- ROOTS OF NONLINEAR EQUATIONS" (SEE P. 220)
C
C  PURPOSE
C  TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM F(X) = 0 BY MEANS
C  OF NEWTON'S ITERATION METHOD.
C
C  USAGE
C  CALL NEWTON (X, F, DERF, XST, EPS, IEND, IER)
C
C  DESCRIPTION OF PARAMETERS
C  X       RESULTANT ROOT OF EQUATION F(X) = 0.
C  F       RESULTANT FUNCTION VALUE AT ROOT X.
C  DERF    RESULTANT VALUE OF DERIVATIVE AT ROOT X.
C  FCT     NAME OF THE EXTERNAL SUBROUTINE USED.  IT COMPUTES TO GIVEN
C            ARGUMENT X FUNCTION VALUE F AND DERIVATIVE DERF.  ITS
C            PARAMETER LIST MUST BE X, F, DERF.
C  XST     INPUT VALUE WHICH SPECIFIES THE INITIAL GUESS OF THE ROOT X.
C  EPS     INPUT VALUE WHICH SPECIFIES THE UPPER BOUND OF THE ERROR OF
C            RESULT X.
C  IEND    MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED
C  IER     RESULTANT ERROR PARAMETER CODED AS FOLLOWS
C              IER = 0 -- NO ERROR
C              IER = 1 -- NO CONVERGENCE AFTER IEND ITERATION STEPS
C              IER = 2 -- AT ANY ITERATION STEP DERIVATIVE DERF WAS
C                         EQUAL TO ZERO
C
C  REMARKS
C  THE PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER = 2 IF AT
C  ANY ITERATION STEP DERIVATIVE OF F(X) IS EQUAL TO 0.  POSSIBLY THE
C  PROCEDURE WOULD BE SUCCESSFUL IF IT IS STARTED ONCE MORE WITH
C  ANOTHER INITIAL GUESS XST.
C
C  SUBROUTINE AND FUNCTION SUBPROGRAMS REQUIRED
C  THE EXTERNAL SUBROUTINE FCT(X, F, DERF) MUST BE FURNISHED BY THE USER
C
C  METHOD
C  SOLUTION OF EQUATION F(X) = 0 IS DONE BY MEANS OF NEWTON'S ITERATION
C  METHOD, WHICH STARTS AT THE INITIAL GUESS XST OF A ROOT X.  CONVERG-
C  ENCE IS QUADRATIC IF THE DERIVATIVE OF F(X) AT ROOT X IS NOT EQUAL
```

```
C  TO ZERO.  ONE ITERATION STEP REQUIRES ONE EVALUATION OF F(X) AND ONE
C  EVALUATION OF THE DERIVATIVE OF F(X).  FOR TEST ON SATISFACTORY
C  ACCURACY SEE FORMULEA (2) OF MATHEMATICAL DESCRIPTION.
C  REFERENCE, SEE R. ZURMUEHL, PRAKTISCHE MATHEMATIK FUER INGENIEURE
C  UND PHYSIKER, SPRINGER, BERLIN/GOETTINGEN/HEIDELBERG, 1963, PP. 12-17
C
C  DATE:  22APR87;  REV. 7JUN88

      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, 1

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

C  SUBROUTINE 'SORTM' SORTS THE ARRAY, ALLM(), FROM LOWEST TO HIGHEST
C  M FOR EACH REGION
C  PROGRAMMER:  L. NEWLIN
C        DATE:  10FEB88
C     VERSION:  MATCHR V7, V7.1, 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
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, INC, IOUT, L, MAXMM, MAXREG, NUM, NUMREG

      PARAMETER (MAXMM = 20001, MAXREG = 3)

      LOGICAL INORDR

      REAL    ALLM(MAXMM, MAXREG), TEMP

C                    LIST OF VARIABLES
C
C  ALLM()     2-D ARRAY CONTAINING VALUES TO BE SORTED FOR EACH REGION
C  I          CONTROLS INSERTION POINTER
C  INC        SORT INCREMENT VARIABLE
C  INORDR     FLAG TO INDICATE WHETHER SORT IS FINISHED
C  IOUT       OUTPUT DUMP CONTROLLER
C  L          CONTROLS DO LOOP FOR EACH REGION
C  MAXMM      MAXIMUM NUMBER OF M'S TO BE SORTED
C  MAXREG     MAXIMUM NUMBER OF REGIONS ALLOWED
C  NUM        NUMBER OF ELEMENTS IN ALLM() TO BE SORTED
C  NUMREG     NUMBER OF REGIONS OF INTEREST
C  TEMP       TEMPORARY SORTING VARIABLE


      DO 400 L = 1, NUMREG

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

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

            IF (.NOT. INORDR) GOTO 20
            GOTO 10
         ENDIF

400 CONTINUE

      RETURN
      END
```

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 7-10 List of Subprograms for Program BFIT

NAME	SECTION	FUNCTION
BFIT	4.2.2	The main routine that controls the logical flow of the prior failure distribution parameter β estimation calculations.
LLS	4.2.2.1	Solves a linear least squares problem.

7.4.1.2 Description of Variables

A list of variables used in the prior failure distribution parameter β 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 7-11 List of Variables for Program BFIT

VARIABLE NAME	TYPE	DESCRIPTION
B	DRE	Slope parameter calculated by LLS, estimate of β.
DIFFX(MAXM)	DRE	1-D array containing values of the difference between **X(I)** and **MEANX** for each point used in the linear regression.

Table 7-11 List of Variables for Program BFIT (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
DIFFY(MAXM)	DRE	1-D array containing values of the difference between **Y(I)** and **MEANY** for each point used in the linear regression.
END	INT	Ending index position for regression performed by LLS.
FOFN	DRE	$F(N_i)$ in *Equation 2-8.*
I	INT	Controls DO loop.
IOUT	INT	Output dump controller.
LIFE(MAXM)	DRE	1-D array containing values of N_i, *Equation 2-8*, generated by the PFM.
LNC	DRE	Intercept parameter calculated by LLS.
M	INT	Length of F, the number of $(F(N_i), N_i)$ pairs provided by the PFM.
MAXM	INT	Maximum value for **M**.
MEANX	DRE	The sample mean of *X*.
MEANY	DRE	The sample mean of *Y*.
START	INT	Starting index position for regression performed by LLS.
SX2	DRE	The sample variance of *X*.
SXY	DRE	The sample covariance of *X* and *Y*.
X(MAXM)	DRE	1-D array containing values of ln N_i corresponding to the N_i's in the array **LIFE( )**.
Y(MAXM)	DRE	1-D array containing values of Y_i, *Equation 2-8*, corresponding to the N_i's in the array **LIFE( )**.

7.4.1.3 Program BFIT Listing

```
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
C   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)

C                       LIST OF VARIABLES
C
C  B          Shape parameter.
C  DUM        Dummy variable used during data entry.
C  END        Ending position for regression loops.
C  FOFN       F(N)
C  I          Controls DO loop.
C  IOUT       Output dump controller.
C  LIFE()     1-D array containing values of N generated by the PFM.
C  LNC        Intercept parameter.
C  M          The number of (F(N), N) pairs provided by the PFM.
C  MAXM       Maximum value for M.
C  START      Starting position for regression loops
C  Y()        1-D array containing values of Y corresponding to the N's
C               in the array LIFE().

C     INITIALIZE VARIABLES AND READ INPUT PARAMETERS

      OPEN (11, FILE = 'BFITD',  STATUS = 'OLD')
      OPEN (8,  FILE = 'IOUTPR', STATUS = 'NEW')
      OPEN (12, FILE = 'LOWLIF', STATUS = 'OLD')
      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.D0 - FOFN))
  100 CONTINUE

C     PERFORM LINEAR LEAST SQUARES TO FIND SHAPE PARAMETER B

      CALL LLS (LIFE, Y, START, END, B, LNC, IOUT)

C     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

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

C  SUBROUTINE LLS PERFORMS A Linear Least Squares FIT
C  PROGRAMMER:  L. NEWLIN
C        DATE:  17AUG90
C     VERSION:  BFIT V1, V1.1

      SUBROUTINE LLS (LIFE, Y, START, END, B, LNC, IOUT)

C       INPUTS:  LIFE, Y, START, END, IOUT
C      OUTPUTS:  B, LNC

C     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)
```

```
C                 LIST OF VARIABLES
C
C  B          Shape parameter.
C  DIFFX()    1-D array containing the difference between X and MEANX for
C               each LIFE.
C  DIFFY()    1-D array containing the difference between Y and MEANY for
C               each LIFE.
C  END        Ending position for regression loops.
C  I          Controls DO loops.
C  IOUT       Output dump controller.
C  LIFE()     1-D array containing values of the lives generated by the PFM
C               (already sorted).
C  LNC        Intercept parameter.
C  MAXM       Maximum number of fatigue lives allowed for regression.
C  MEANX      Sample mean of X.
C  MEANY      Sample mean of Y.
C  START      Starting position for regression loops.
C  SX2        Sample variance of X.
C  SXY        Sample covariance of X and Y.
C  X()        X = ln (Life)
C  Y()        Y = ln (- ln (1 - F(Life)))

C     INITIALIZE VARIABLES

      MEANX = 0.D0
      MEANY = 0.D0
      SX2 = 0.D0
      SXY = 0.D0
      DO 50 I = 1, MAXM
         X(I) = 0.D0
         DIFFX(I) = 0.D0
         DIFFY(I) = 0.D0
   50 CONTINUE

C     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

C     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 = SX2 / DFLOAT(END - START)
      SXY = SXY / DFLOAT(END - START)

      WRITE(8,*) 'SX2 = ', SX2, ' SXY = ', SXY

C     CALCULATE REGRESSION PARAMETERS

      B = SXY / 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.

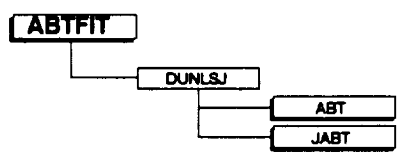

Figure 7-11 Tree Structure For Program ABTFIT

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 7-12 List of Subprograms for Program ABTFIT

NAME	SECTION	FUNCTION
ABT	4.2.3.1	Evaluates the function which defines the least squares problem.
ABTFIT	4.2.3	The main routine that controls the logical flow of the prior failure distribution parameter estimation calculations.
DUNLSJ	N/A	An IMSL routine that solves a nonlinear least squares problem by using a modified Levenberg-Marquardt algorithm.
JABT	4.2.3.2	Evaluates the Jacobian at a point *X*.

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 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 following parameters: **MAXJAC**, **MAXM**, **N**, and **N2**.

Table 7-13 List of Variables for Program ABTFIT

VARIABLE NAME	TYPE	DESCRIPTION
ABT	DRE	Evaluates the function given by *Equation 2-10* which defines the least squares problem.
ARGA, ARGB	DRE	Intermediate calculation variables.
B	DRE	Slope parameter, *b*.
END	INT	Ending index position for regression.
F(M)	DRE	Vector of length **M** containing the function values at **X**.
FJAC(MAXJAC, N)	DRE	The **M** by **N** matrix containing the calculated Jacobian at the approximate solution.
FOFN	DRE	$F(N_i)$ in *Equation 2-10*.
FSCALE(MAXM)	DRE	Vector of length **M** containing the diagonal scaling matrix for the functions.
FVEC(MAXM)	DRE	Vector of length **M** containing the residuals at the approximate solution.
I	INT	Controls DO loop.
IOUT	INT	Output dump controller.
IPARAM(6)	INT	Parameter vector of length 6, required by DUNLSJ.
J	INT	Array index variable.
JABT	DRE	Evaluates the Jacobian at a point **X**.
LDFJAC	INT	Leading dimension of **FJAC**.
LIFE(MAXM)	DRE	1-D array containing values of N_i *Equation 2-10*, generated by the PFM and scaled by **LSCALE**.
LIFEM(MAXM)	DRE	1-D array containing all values of N_i *Equation 2-10*, generated by the PFM.
LSCALE	DRE	Used to scale N_i values to control instability in θ estimation.

Table 7-13 List of Variables for Program ABTFIT (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
M	INT	Length of **F**, the number of $(F(N_i), N_i)$ pairs to be used.
MAXJAC	INT	Maximum value for **LDFJAC**.
MAXM	INT	Maximum value for **M**.
MTOT	INT	The total number of $(F(N_i), N_i)$ pairs provided by the PFM.
N	INT	Length of **X**, two for this application.
N2	INT	Dummy parameter replacing **N** in routines ABT and JABT.
RPARAM(7)	DRE	Parameter vector of length 7, required by DUNLSJ.
START	INT	Starting index position for regression.
X(N)	DRE	Vector of length **N** containing the approximate solution. **X(1)** is θ, and **X(2)** is α.
XGUESS(N)	DRE	Vector of length **N** containing the initial guess.
XSCALE(N)	DRE	Vector of length **N** containing the diagonal scaling matrix for the variables.
Y(MAXM)	DRE	1-D array containing values of Y_i, *Equation 2-10*, corresponding to the N_i's in the array **LIFE( )**.

7.4.2.4 Program ABTFIT Listing

```
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),
     &          FSCALE(MAXM), FVEC(MAXM), JABT, LIFE(MAXM),
     &          LIFEM(MAXM), LSCALE, RPARAM(7), X(N), XGUESS(N),
     &          XSCALE(N), Y(MAXM)
```

```
      EXTERNAL  ABT, JABT, DUNLSJ

C                            LIST OF VARIABLES
C
C  ABT          Evaluates the function which defines the nonlinear least
C                 squares problem.
C  B            Shape parameter.
C  DUM          Dummy variable used during data entry.
C  END          Ending position for regression loops
C  FJAC()       The M by N matrix containing the calculated Jacobian at the
C                 approximate solution.
C  FOFN()       1-D array containing values of F(Ni).
C  FSCALE()     Vector of length M containing the diagonal scaling matrix for
C                 the functions.
C  FVEC()       Vector of length M containing the residuals at the approximate
C                 solution.
C  I            Controls DO loop.
C  IOUT         Output dump controller.
C  IPARAM()     Parameter vector of length 6, required by DUNLSJ.
C  J            Array index variable.
C  JABT         Evaluates the Jacobian at a point X.
C  LDFJAC       Leading dimension of FJAC.
C  LIFE()       1-D array containing values of the life, Ni, generated by the
C                 PFM  to be used by the nonlinear least squares algorithm.
C  LIFEM()      1-D array containing all MTOT values of the life, Ni, generated
C                 by the PFM.
C  LSCALE       LIFE() scaling parameter to enable convergence.
C  M            Length of F, the number of (F(Ni), Ni) pairs provided by the
C                 PFM to be used by the nonlinear least squares algorithm.
C  MAXJAC       Maximum value for LDFJAC.
C  MAXM         Maximum value for M and MTOT.
C  MTOT         Length of FOFN() & LIFEM(), the number of (F(Ni), Ni) pairs
C                 provided by the PFM.
C  N            Length of X, two for this application.
C  RPARAM()     Parameter vector of length 7, required by DUNLSJ.
C  START        Starting position for regression loops
C  X()          Vector of length N containing the approximate solution.  X(1)
C                 is theta, X(2) is alpha.
C  XGUESS()     Vector of length N containing the initial guess.
C  XSCALE()     Vector of length N containing the diagonal scaling matrix for
C                 the variables.
C  Y()          1-D array containing values of Y corresponding to the Ni's in
C                 the array LIFE().

C     INITIALIZE VARIABLES AND READ INPUT PARAMETERS

      DATA FSCALE / MAXM*1.0D0 /

      IPARAM(1) = 0

      OPEN (11, FILE = 'PARAMS', STATUS = 'OLD')
      READ(11,*) IOUT, 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. 20) THEN
         OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')
         WRITE(8,*) '          THETA            ALPHA'
      ENDIF

      OPEN (12, FILE = 'LOWLIF', STATUS = 'OLD')
      DO 100 I = 1, MTOT
         READ(12,*) DUM, FOFN(I), LIFEM(I)
  100 CONTINUE
      CLOSE (12)

      DO 200 I = 1, M
         J = I + START - 1
```

```
         LIFE(I) = LIFEM(J) * LSCALE
         Y(I) = - DLOG (1.D0 - FOFN(J))
  200 CONTINUE

C     PERFORM NONLINEAR LEAST SQUARES TO FIND THETA AND ALPHA

      CALL DUNLSJ (ABT, JABT, M, N, XGUESS, XSCALE, FSCALE,
     &             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)

C                    LIST OF VARIABLES
C
C  B          Shape parameter.
C  F()        Vector of length M containing the function values at X.
C  I          Controls DO loop.
C  IOUT       Output dump controller.
C  LIFE()     1-D array containing values of the life, Ni, generated by
C               the PFM.
C  M          Length of F, the number of (F(Ni), Ni) pairs provided by
C               the PFM.
C  MAXM       Maximum value for M.
C  N          Length of X, two for this application.
C  N2         Dummy parameter replacing N in routines ABT and JABT.
C  X()        Vector of length N containing the approximate solution.
C               X(1) is theta, X(2) is alpha.
C  XGUESS()   Vector of length N containing the initial guess.
C  Y()        1-D array containing values of Y corresponding to the Ni's
C               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.D0 + (LIFE(I) ** B) / X(1))

         IF (IOUT .EQ. 10)
     &      WRITE(8,150) I, F(I), X(1), X(2), LIFE(I), Y(I)

  100 CONTINUE

  150 FORMAT (I5,1X,D10.3,1X,D10.3,1X,D10.3,1X,2D16.10)
  175 FORMAT (2X,'F  ',D13.6,1X,D13.6)

      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

      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)

C                       LIST OF VARIABLES
C
C  ARGA        Intermediate calculation variables.
C  ARGB        Intermediate calculation variables.
C  B           Shape parameter.
C  F()         Vector of length M containing the function values at X.
C  FJAC()      The M by N matrix containing the calculated Jacobian at the
C                approximate solution.
C  I           Controls DO loop.
C  IOUT        Output dump controller.
C  JABT        Evaluates the Jacobian at a point X.
C  LDFJAC      Leading dimension of FJAC exactly as specified in the dimension
C                statement of the calling program.
C  LIFE()      1-D array containing values of the life, Ni, generated by
C                the PFM.
C  M           Length of F, the number of (F(Ni), Ni) pairs provided by
C                the PFM.
C  MAXM        Maximum value for M.
C  N           Length of X, two for this application.
C  N2          Dummy parameter replacing N in routines ABT and JABT.
C  X(N)        Vector of length N containing the approximate solution.
C                X(1) is theta, X(2) is alpha.
C  XGUESS()    Vector of length N containing the initial guess.
C  Y()         1-D array containing values of Y corresponding to the Ni's
C                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.D0 + 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,*) 'ARGs ', ARGA, ARGB
            WRITE(8,*) 'FJAC1 = ', FJAC(I,1), ' FJAC2 = ', FJAC(I,2)
```

```
        ENDIF
  100 CONTINUE
  175 FORMAT (2X,'J   ',D13.6,1X,D13.6)
      RETURN
      END
```

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.

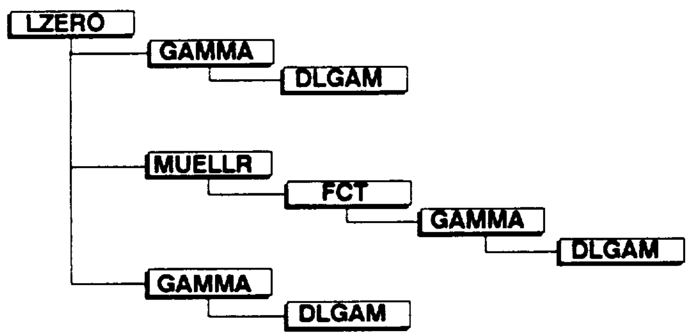

Figure 7-12 Tree Structure For Program LZERO

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 7-14 List of Subprograms for Program LZERO

NAME	SECTION	FUNCTION
DLGAM	4.2.4.2	Calculates the double precision value of $\ln[\Gamma(x)]$ for a given value of x.
FCT	4.2.4.4	Calculates the value of the function at the current iteration value of λ, in order to find the solution for the desired assurance level.
GAMMA	4.2.4.1	Calculates the cumulative distribution function for a Gamma variate.
LZERO	4.2.4	The main routine that controls the logical flow of the assurance calculation.

Table 7-14 List of Subprograms for Program LZERO (Cont'd)

NAME	SECTION	FUNCTION
MUELLR	4.2.4.3	Solves the general nonlinear equation of the $f(x) = 0$ form by means of Mueller's iteration method.
TRMNAT	4.2.4.5	Performs premature program termination when required.

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

VARIABLE NAME	TYPE	DESCRIPTION
A	DRE	Desired assurance level requested by the user.
ALPHA	DRE	Gamma distribution parameter α in *Equation 2-1.*
BETA	DRE	Weibull distribution parameter β in *Equation 2-1.*
DLGAM	DRE	Function that calculates the natural logarithm of a Gamma function.
EPS	DRE	Value of the maximum error allowed in the result of the Mueller's iteration method calculations.
F	DRE	Cumulative distribution function in *Equation 2-5*. When λ equals λ_o, **F** equals the desired assurance level.
FL	DRE	Cumulative distribution function in *Equation 2-5.* When λ equals λ_{lb}, **FL** equals the assurance level.
FU	DRE	Cumulative distribution function in *Equation 2-5.* When λ equals λ_{ub}, **FU** equals the assurance level.
IER	INT	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.
IOUT	INT	Output dump controller.
LAMBDA	DRE	The random variate λ in *Equation 2-1*. When λ equals λ_o, **F** equals the desired assurance level.

Table 7-15 List of Variables for Program LZERO (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
LAML	DRE	The lower bound of λ_0 in *Equation 2-1.* When λ equals λ_{lb}, FL equals the assurance level.
LAMU	DRE	The upper bound of λ_0 in *Equation 2-1.* When λ equals λ_{ub}, FU equals the assurance level.
NS	INT	Number of terms of the infinite series used to approximate the Gamma cumulative distribution function.
PSI	DRE	Computational precision for calculating the value of a Gamma cumulative distribution.
THETA	DRE	Gamma distribution parameter θ in *Equation 2-1.*

7.4.3.4 Program LZERO Listing

```
C   PROGRAM LZERO CALCULATES THE PARAMETER LAMBDA FOR A GIVEN ASSURANCE
C   PROGRAMMER:  L. NEWLIN
C         DATE:  24SEP91
C      VERSION:  V2
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      PROGRAM LZERO

C   SUBPROGRAMS: GAMMA, MUELLR, TRMNAT
C         FILES: 1:BAYESD-OLD, 2:LDAT-OLD,
C                3:LOUT-NEW,    8:IOUTPR-NEW

C     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

C             LIST OF VARIABLES
C
C   A          REQUESTED ASSURANCE LEVEL
C   ALPHA      GAMMA DISTRIBUTION PARAMETER
C   BETA       WEIBULL DISTRIBUTION PARAMETER
C   EPS        COMPUTATIONAL PRECISION FOR SUBPROGRAM MUELLR
C   F          CUMULATIVE DISTRIBUTION FUNCTION AT LAMBDA
C   FL         CUMULATIVE DISTRIBUTION FUNCTION AT LAML
C   FU         CUMULATIVE DISTRIBUTION FUNCTION AT LAMU
C   IER        RESULTANT ERROR PARAMETER CODED AS FOLLOWS:
C                IER = 0  -  NO ERROR.
C                IER = 1  -  NO CONVERGENCE AFTER IEND ITERATION STEPS
C                            FOLLOWED BY IEND SUCCESSIVE STEPS OF
C                            BISECTION.
C                IER = 2  -  BASIC ASSUMPTION FCT(XLI) * FCT(XRI) LESS
C                            THAN OR EQUAL TO ZERO IS NOT SATISFIED.
```

```
C  IOUT        OUTPUT DUMP CONTROLLER
C  LAMBDA      THRESHOLD LAMBDA AT WHICH F() IS CALCULATED
C  LAML        LAMBDA LOWER BOUND
C  LAMU        LAMBDA UPPER BOUND
C  NS          NUMBER OF TERMS OF THE INFINITE SERIES TO CALCULATE GAMMA
C  PSI         COMPUTATIONAL PRECISION FOR SUBPROGRAM GAMMA
C  THETA       WEIBULL DISTRIBUTION PARAMETER

      PSI = 1.0D-6
      EPS = 1.0D-10
      NS = 1000

      OPEN (1, FILE = 'BAYESD', STATUS = 'OLD')
      OPEN (2, FILE = 'LDAT',   STATUS = 'OLD')
      OPEN (3, FILE = 'LOUT',   STATUS = 'NEW')

      READ(1,*) BETA, THETA, ALPHA
      READ(2,*) IOUT, A, LAML, LAMU
      WRITE(3,900) ALPHA, THETA

      CLOSE(1)
      CLOSE(2)
      IF (IOUT .EQ. 10) OPEN (8, FILE = 'IOUTPR', STATUS = 'NEW')

      CALL 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 (LAMBDA, LAML, LAMU, EPS, NS, IER)

      IF (IER .EQ. 0) THEN
         CALL GAMMA (LAMBDA, ALPHA, THETA, NS, PSI, F)
         WRITE(3,920) F, LAMBDA
      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)

  910 FORMAT (//,2X,'Lambda lower bound  ',D12.7,
     &         5X,'Assurance  ',F9.7,
     &         //,2X,'Lambda upper bound  ',D12.7,
     &         5X,'Assurance  ',F9.7)

  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
```

```
C        DATE:  8DEC89
C     VERSION:  LZERO V1, V2

      SUBROUTINE GAMMA (X, ALPHA, THETA, NS, PSI, F)

C  SUBPROGRAMS:  DLGAM

C     IMPLICIT NONE

      COMMON   IOUT

      INTEGER  I, IOUT, NS

      DOUBLE PRECISION  ALPHA, DLGAM, F, FB, PSI, RI, THETA, X

C             LIST OF VARIABLES
C
C  ALPHA       WEIBULL DISTRIBUTION PARAMETER
C  DLGAM       FUNCTION THAT CALCULATES LOGARITHM OF GAMMA(XX)
C  F           CUMULATIVE DISTRIBUTION FUNCTION
C  FB          INCREMENT OF F
C  I           COUNTER FOR SERIES SUMMATION
C  IOUT        OUTPUT DUMP CONTROLLER
C  X           THRESHOLD AT WHICH F() IS CALCULATED
C  NS          NUMBER OF TERMS OF THE INFINITE SERIES TO CALCULATE GAMMA
C  PSI         COMPUTATIONAL PRECISION
C  RI          REAL VALUE OF I
C  THETA       WEIBULL DISTRIBUTION PARAMETER

      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'ALPHA = ', ALPHA, ' THETA = ', THETA
      I = -1
      F = 0.D0

C  SUM THE SERIES FROM I = 1, NS

    1 I = I + 1
      IF (I .GT. NS) GO TO 3
      RI = DFLOAT(I)
      FB = DEXP (- DLGAM (ALPHA + 1.D0 + 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.D0
      RETURN
      END

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

C  THE FUNCTION CALCULATES THE DOUBLE PRECISION NATURAL LOGARITHM OF
C  THE GAMMA FUNCTION OF A GIVEN DOUBLE PRECISION ARGUMENT.
C
C  NOTE:  THIS IS A MODIFIED VERSION OF THE SUBROUTINE OF THE SAME
C         NAME FOUND IN 'Mathematics -- Special Functions', IBM
C         APPLICATION PROGRAM, SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE,
C         VERSION III, PROGRAMMER'S MANUAL, Program Number 360-CM-03X,
C         Page 362.
C
C  METHOD:  THE EULER-MCLAURIN EXPANSION TO THE SEVENTH DERIVATIVE
C           TERM IS USED, AS GIVEN BY M. ABRAMOWITZ AND I.A. STEGUN,
C           'HANDBOOK OF MATHEMATICAL FUNCTIONS', U.S. DEPARTMENT OF
C           COMMERCE, NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICS
C           SERIES.  1966, EQUATION 6.1.41.

      DOUBLE PRECISION FUNCTION DLGAM(XX)
```

```
C  SUBPROGRAMS:  TRMNAT

      COMMON IOUT
      DOUBLE PRECISION XX, ZZ, TERM, RZZ, DLNG
      INTEGER IER, IOUT

C  XX      THE DOUBLE PRECISION ARGUMENT FOR THE LOG GAMMA FUNCTION
C  ZZ      VARIABLE ACTUALLY USED IN THE CALCULATION INSTEAD OF XX
C  TERM    USED TO CONVERT TO X <= 18
C  RZZ     1/ZZ**2
C  DLNG    HOLDS VALUE OF DLGAM DURING CALCULATIONS
C  IER     ERROR CONTROLLER:
C            -1 -- XX IS WITHIN 10**(-9) OF BEING ZERO OR XX
C                  IS NEGATIVE.  DLNG IS SET TO -1.0D35
C             0 -- NO ERROR.
C            +1 -- XX IS GREATER THAN 10**30. DLNG IS SET TO +1.0D35.

      IER = 0
      ZZ = XX
      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 = 1.D0/ZZ**2.
      DLNG = (ZZ-0.5D0)*DLOG(ZZ)-ZZ +0.9189385332046727 -DLOG(TERM)+
     1(1.D0/ZZ)*(.8333333333333333D-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

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

      SUBROUTINE TRMNAT

      STOP
      END
```

```
C**********************************************************************

C***********************************************************************
C  FUNCTION FCT CALCULATES THE CUMULATIVE DISTRIBUTION FUNCTION FOR
C  A GAMMA VARIATE MINUS THE DESIRED ASSURANCE
C  PROGRAMMER:  L. NEWLIN
C        DATE:  24SEP91
C     VERSION:  LZERO V2

      DOUBLE PRECISION FUNCTION FCT (X)

C     SUBPROGRAMS:  GAMMA

      COMMON IOUT

      COMMON / INFO / A, ALPHA, THETA, NS, PSI

      INTEGER  IOUT, NS

      DOUBLE PRECISION  A, ALPHA, F, PSI, THETA, X

C            LIST OF VARIABLES
C
C  A          REQUESTED ASSURANCE LEVEL
C  ALPHA      GAMMA DISTRIBUTION PARAMETER
C  F          CUMULATIVE DISTRIBUTION FUNCTION
C  IOUT       OUTPUT DUMP CONTROLLER
C  X          THRESHOLD LAMBDA AT WHICH F() IS CALCULATED
C  NS         NUMBER OF TERMS OF THE INFINITE SERIES TO CALCULATE GAMMA
C  PSI        COMPUTATIONAL PRECISION
C  THETA      WEIBULL DISTRIBUTION PARAMETER

      CALL GAMMA (X, ALPHA, THETA, NS, PSI, F)

      IF (IOUT .EQ. 10)
     &   WRITE(8,*) 'X = ', X, ' F = ', F

      FCT = F - A

      RETURN
      END

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

C  SUBROUTINE MUELLR
C  THE FOLLOWING SUBROUTINE IS A MODIFIED VERSION OF SUBROUTINE
C  DRTMI TAKEN FROM "Mathematics -- Roots of Nonlinear Equations,"
C  IBM APPLICATION PROGRAM, SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE,
C  VERSION III, PROGRAMMER'S MANUAL, Program Number 360-CM-03X, Page 219.
C
C  PURPOSE
C     TO SOLVE GENERAL NONLINEAR EQUATIONS OF THE FORM FCT(X) = 0
C     BY MEANS OF MUELLER'S ITERATION METHOD.
C
C  USAGE
C     CALL MUELLR (X, XLI, XRI, EPS, IEND, IER)
C
C  DESCRIPTION OF PARAMETERS
C     X       - DOUBLE PRECISION RESULTANT ROOT OF EQUATION FCT(X) = 0.
C     F       - DOUBLE PRECISION RESULTANT FUNCTION VALUE AT ROOT X.
C     FCT     - NAME OF THE EXTERNAL DOUBLE PRECISION FUNCTION
C               SUBPROGRAM USED.
C     XLI     - DOUBLE PRECISION INPUT VALUE WHICH SPECIFIES THE
C               INITIAL LEFT BOUND OF THE ROOT X.
C     XRI     - DOUBLE PRECISION INPUT VALUE WHICH SPECIFIES THE
C               UPPER BOUND OF THE ERROR OF RESULT X.
C     IEND    - MAXIMUM NUMBER OF ITERATION STEPS SPECIFIED.
```

```
C     IER     - RESULTANT ERROR PARAMETER CODED AS FOLLOWS
C                 IER = 0  -  NO ERROR.
C                 IER = 1  -  NO CONVERGENCE AFTER IEND ITERATION STEPS
C                             FOLLOWED BY IEND SUCCESSIVE STEPS OF
C                             BISECTION.
C                 IER = 2  -  BASIC ASSUMPTION FCT(XLI) * FCT(XRI) LESS
C                             THAN OR EQUAL TO ZERO IS NOT SATISFIED.
C
C  REMARKS
C     THE PROCEDURE ASSUMES THAT FUNCTIONS VALUES AT INITIAL
C     BOUNDS XLI AND XRI HAVE NOT THE SAME SIGN.  IF THIS BASIC
C     ASSUMPTION IS NOT SATISFIED BY INPUT VALUES XLI AND XRI, THE
C     PROCEDURE IS BYPASSED AND GIVES THE ERROR MESSAGE IER = 2.
C
C  SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C     THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)
C     MUST BE FURNISHED BY THE USER.
C
C  METHOD
C     SOLUTION OF EQUATION FCT(X) = 0 IS DONE BY MEANS OF MUELLER'S
C     ITERATION METHOD OF SUCCESSIVE BISECTIONS AND INVERSE
C     PARABOLIC INTERPOLATION, WHICH STARTS AT THE INITIAL BOUNDS
C     XLI AND XRI. CONVERGENCE IS QUADRATIC IF THE DERIVATIVE OF
C     FCT(X) AT ROOT X IS NOT EQUAL TO ZERO. ONE ITERATION STEP
C     REQUIRES TWO EVALUATIONS OF FCT(X). FOR TEST ON SATISFACTORY
C     ACCURACY SEE FORMULAE (3,4) OF MATHEMATICAL DESCRIPTION.
C     FOR REFERENCE, SEE G. K. KRISTIANSEN, ZERO OF ARBITRARY
C     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

      IF (IOUT .EQ. 20) THEN
         WRITE(8,*) 'X = ', X
         WRITE(8,*) '   XLI = ', XLI, ' XRI = ', XRI
      ENDIF

C     PREPARE ITERATION

      IER = 0
      XL = XLI
      XR = XRI
      X = XL
      TOL = X
      F = FCT(TOL)
      IF (F) 1, 16, 1
    1 FL = F
      X = XR
      TOL = X
      F = FCT(TOL)
      IF (F) 2, 16, 2
    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

    3 I = 0
      TOLF = 100. * EPS

C     START ITERATION LOOP

    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.
      IER = 1
   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) 16, 16, 22
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)

VARIABLE NAME	TYPE	DESCRIPTION
ALPHA	RE	Gamma distribution parameter α for prior distribution of Weibull parameter λ, *Equation 2-1.*
ALPHUP	RE	Gamma distribution parameter α' for posterior distribution of Weibull parameter λ, *Equation 2-2.*
B1	RE	B1-life[1] calculated from *Equation 2-6* by using prior distribution parameters α, θ, and β.
B1UP	RE	B1-life[1] calculated from *Equation 2-6* by using posterior distribution parameters α', θ', and β.
BETA	RE	Weibull distribution shape parameter β, *Equation 2-1.*
BP01	RE	B.01-life[1] calculated from *Equation 2-6* by using prior distribution parameters α, θ, and β.
BP01UP	RE	B.01-life[1] calculated from *Equation 2-6* by using posterior distribution parameters α', θ', and β.
BP1	RE	B.1-life[1] calculated from *Equation 2-6* by using prior distribution parameters α, θ, and β.
BP1UP	RE	B.1-life[1] calculated from *Equation 2-6* by using posterior distribution parameters α', θ', and β.

Table 7-16 List of Variables for Program BAYES (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
ETA	RE	Estimate of $E(\eta)$ where $\eta = \lambda^{-1/\beta}$ with $\lambda \sim \Gamma(\alpha, \theta)$
ETUP	RE	Estimate of $E(\eta)$ where $\eta = \lambda^{-1/\beta}$ with $\lambda \sim \Gamma(\alpha', \theta')$.
FAIL	INT	Number of test failure times in operating experience data.
I	INT	Controls DO loop.
INVALP	RE	Value of $1/\alpha$.
INVALU	RE	Value of $1/\alpha'$.
INVBET	RE	Value of $1/\beta$.
LAMBDA	RE	Expected value of $\lambda \sim \Gamma(\alpha, \theta)$ in *Equation 2-1*.
LAMBUP	RE	Expected value of $\lambda \sim \Gamma(\alpha', \theta')$ in *Equation 2-2*.
SUSP	INT	Number of test suspension times in operating experience data.
MAXTYM	INT	Maximum number of failure and/or suspension times allowed. The maximum number of times allowed is 50.
THETA	RE	Gamma distribution parameter θ for prior distribution of Weibull parameter λ, *Equation 2-1*.
THETUP	RE	Gamma distribution parameter θ' for posterior distribution of Weibull parameter λ, *Equation 2-2*.
TYME(MAXTYM)	RE	1-D array containing operating experience as failure and suspension times t_i, *Equation 2-2*.

[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%.

7.5.1.2 Program BAYES Listing

```
C  PROGRAM BAYES PERFORMS THE BAYESIAN UPDATING OF THE PRIOR DISTRIBUTION
C  (FROM THE PFM) WITH THE OPERATING HISTORY (BOTH FAILURES AND SUSPENSIONS)
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:   12MAY88      COMMENTS:  11SEP91
C     VERSION:  3.1
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      PROGRAM BAYES

C  FILES:  1:BAYESD-OLD;  2:BAYESO-NEW;  3:UBAYES-NEW;

C     IMPLICIT NONE

      INTEGER  MAXTYM

      PARAMETER (MAXTYM = 50)

      INTEGER  FAIL, I, SUSP

      REAL     ALPHA, ALPHUP, B1, B1UP, BETA, BP01, BP01UP, BP1, BP1UP,
     &         ETA, ETUP, INVALP, INVALU, INVBET, LAMBDA, LAMBUP,
     &         THETA, THETUP, TYME(MAXTYM)

C                          LIST OF VARIABLES
C
C  ALPHA     PRIOR GAMMA DISTRIBUTION PARAMETER ALPHA
C  ALPHUP    POSTERIOR OR UPDATED GAMMA DISTRIBUTION PARAMETER ALPHA
C  B1        PRIOR DISTRIBUTION B1 LIFE CALCULATED FROM BETA, THETA AND
C              ALPHA
C  B1UP      POSTERIOR OR UPDATED DISTRIBUTION B1 LIFE CALCULATED FROM BETA,
C                THETUP AND ALPHUP
C  BETA      WEIBULL DISTRIBUTION SHAPE PARAMETER
C  BP01      PRIOR DISTRIBUTION B.01 LIFE CALCULATED FROM BETA, THETA AND
C              ALPHA
C  BP01UP    POSTERIOR OR UPDATED DISTRIBUTION B.01 LIFE CALCULATED FROM BETA,
C              THETUP AND ALPHUP
C  BP1       PRIOR DISTRIBUTION B.1 LIFE CALCULATED FROM BETA, THETA AND
C              ALPHA
C  BP1UP     POSTERIOR OR UPDATED DISTRIBUTION B.1 LIFE CALCULATED FROM BETA,
C              THETUP AND ALPHUP
C  ETA       PRIOR WEIBULL DISTRIBUTION LOCATION PARAMETER
C  ETUP      POSTERIOR OR UPDATED WEIBULL DISTRIBUTION LOCATION PARAMETER
C  FAIL      NUMBER OF FAILURES IN OPERATING EXPERIENCE
C  I         CONTROLS DO LOOP
C  INVALP    INVerse of ALPha -- EQUAL TO 1/ALPHA
C  INVALU    INVerse of ALphUp -- EQUAL TO 1/ALPHUP
C  INVBET    INVerse of BETa -- EQUAL TO 1/BETA
C  LAMBDA    PRIOR WEIBULL DISTRIBUTION PARAMETER EQUAL TO (1/ETA)**BETA
C  LAMBUP    POSTERIOR OR UPDATED WEIBULL DISTRIBUTION PARAMETER EQUAL
C              TO (1/ETUP)**BETA
C  MAXTYM    MAXIMUM NUMBER OF FAILURE AND/OR SUSPENSION TIMES ALLOWED
C  SUSP      NUMBER OF SUSPENSIONS IN OPERATING EXPERIENCE
C  THETA     PRIOR GAMMA DISTRIBUTION PARAMETER THETA
C  THETUP    POSTERIOR OR UPDATED GAMMA DISTRIBUTION PARAMETER THETA
C  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
```

```
C  READ DATA FROM FILE BAYESD

      READ(1,*) BETA, THETA, ALPHA
      READ(1,*) FAIL, SUSP

      DO 50 I = 1, (FAIL + SUSP)
         READ(1,*) 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

      LAMBDA = ALPHA / THETA
      LAMBUP = ALPHUP / THETUP

      INVBET = 1.0 / BETA
      INVALP = 1.0 / ALPHA
      INVALU = 1.0 / ALPHUP

      ETA = (1.0 / LAMBDA) ** INVBET
      ETUP = (1.0 / LAMBUP) ** 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 ** (- INVALU) - 1)) ** INVBET
      BP1UP  = (THETUP * (0.999  ** (- INVALU) - 1)) ** INVBET
      B1UP   = (THETUP * (0.99   ** (- INVALU) - 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, ALPHA, ALPHUP, THETA, THETUP,
     &             LAMBDA, LAMBUP, ETA, ETUP
      WRITE(2,910) BP01, BP01UP, BP1, BP1UP, B1, B1UP
      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,'LAMBDA',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

VARIABLE NAME	TYPE	DESCRIPTION
FRAC	RE	The Uniform(0,1) random variate generated by RANDOM.
IOUT	INT	Output dump controller.
RANA	DRE	Constant used by the linear congruential algorithm.
RANC	DRE	Constant used by the linear congruential algorithm.
RAND	DRE	Random number seed.
RANDIV	DRE	Intermediate calculation variable.
RANM	DRE	Constant used by the linear congruential algorithm.
RANSUB	DRE	Intermediate calculation variable.
RANT	DRE	Intermediate calculation variable.
RANX	DRE	Intermediate calculation variable.

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
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C        VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                         V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
C                  MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                         V4.3, V4.4, V4.5
C***********************************************************************

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

C                  LIST OF VARIABLES
C
C  FRAC      UNIFORM (0,1) RANDOM VARIATE
C  IOUT      OUTPUT DUMP CONTROLLER
C  RANA      CONSTANT FOR LCG
C  RANC      CONSTANT FOR LCG
C  RAND      RANDOM NUMBER SEED
C  RANDIV    INTERNAL CALCULATION
C  RANM      CONSTANT FOR LCG
C  RANSUB    INTERNAL CALCULATION
C  RANT      INTERNAL CALCULATION
C  RANX      INTERNAL CALCULATION


C          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, σ^2) 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 7-18 List of Variables for Subprogram NORMGN

VARIABLE NAME	TYPE	DESCRIPTION
FRAC	RE	A Uniform(0,1) random variate generated by RANDOM.
IOUT	INT	Output dump controller.
MU	RE	Mean μ of the Normal distribution.
PI	RE	π, constant equal to 3.1415926536.
RAND	DRE	Random number seed.
SIGMA	RE	Standard deviation σ of the Normal distribution.
X	RE	The Normal(μ, σ^2) random variate generated by NORMGN.
U1	RE	U(0,1) random variate.
U2	RE	U(0,1) random variate.
Z1	RE	N(0,1) random variate.
Z2	RE	N(0,1) random variate.

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     VERSION:  MATCHR V7, V7.1, 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
C  The random variates are generated using the "Direct Method"
C  Abramowitz, M., and Stegun, I. A., editors, Handbook of
C    Mathematical Functions, National Bureau of Standards, Applied
C    Mathematics Series 55, Issued June 1964, Ninth Printing, November
C    1970 with corrections, pg. 953.
C*********************************************************************

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

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

C             LIST OF VARIABLES
C
C  FRAC     UNIFORM(0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  MU       MEAN OF NORMAL DISTRIBUTION
C  RAND     RANDOM NUMBER SEED
C  SIGMA    STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X        NORMAL RANDOM VARIATE
C  U1       UNIFORM RANDOM NUMBER U(0,1)
C  U2       UNIFORM RANDOM NUMBER U(0,1)
C  Z1       NORMAL RANDOM NUMBER ON N(0,1)
C  Z2       NORMAL RANDOM NUMBER ON N(0,1)

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

      CALL RANDOM (FRAC, RAND)
      U1 = FRAC

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

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

      X = SIGMA * Z1 + MU
      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, ρ, 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

VARIABLE NAME	TYPE	DESCRIPTION
A	RE	Lower bound *a* of the Beta distribution.
B	RE	Upper bound *b* of the Beta distribution.
GAM	RE	The Gamma(α, 1) random variate generator GAM.
IOUT	INT	Output dump controller.
RAND	DRE	Random number seed.
RHO	RE	Beta distribution parameter ρ.
THETA	RE	Beta distribution parameter θ.
W	RE	A Beta(0, 1, ρ, θ) random variate
X	RE	The Beta(a, b, ρ, θ) random variate generated by BETAGN.
Y1	RE	Gamma(α, 1) random variate.
Y2	RE	Gamma(α, 1) random variate.

7.6.3.2 Subprogram BETAGN Listing

```
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

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

C  TRANSFORMING STANDARD BETA DISTRIBUTION TO BETA DISTRIBUTION

      X = W * (B - A) + A
      IF (IOUT .EQ. 15) WRITE(8,*) 'W =', W, ' X =', X

      RETURN
      END
```

7.6.4 GAM Subprogram

7.6.4.1 Description of Variables

A list of variables used in the Gamma(α, 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 7-20 List of Variables for Subprogram GAM

VARIABLE NAME	TYPE	DESCRIPTION
A	RE	Intermediate calculation variable.
ALPHA	RE	Gamma distribution parameter α.
ARG	RE	Intermediate calculation variable.
GAM	RE	The Gamma(α, 1) random variate generated by GAM.
IOUT	INT	Output dump controller.
RAND	DRE	Random number seed.
U1	RE	U(0,1) random variate.
U2	RE	U(0,1) random variate.
V1	RE	Exponential(1) random variate.
V2	RE	Exponential(1) random variate.

7.6.4.2 Subprogram GAM Listing

```
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     pp. 407-409.

      REAL FUNCTION GAM (ALPHA, RAND)

C     SUBPROGRAM:  RANDOM

      COMMON  IOUT

      INTEGER IOUT

      REAL    A, ALPHA, ARG, U1, U2, V1, V2

      DOUBLE PRECISION RAND

      A = ALPHA - 1.
C     IF (IOUT .EQ. 15) WRITE(8,*) 'A =', A, ' ALPHA =', ALPHA
```

```
10    CALL RANDOM (U1, RAND)
      CALL RANDOM (U2, RAND)
      V1 = - ALOG(U1)
      V2 = - ALOG(U2)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'U1 =', U1, ' U2 =', U2, ' V1 =',
C    &                   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
```

7.6.5 WEIBGN Subprogram

7.6.5.1 Description of Variables

A list of variables used in the Weibull(β, $\eta(\beta)$) 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 7-21 List of Variables for Subprogram WEIBGN

VARIABLE NAME	TYPE	DESCRIPTION
ARG	RE	Intermediate calculation variable.
BETA	RE	Weibull distribution shape parameter β.
ETA	RE	Weibull distribution location parameter η.
FRAC	RE	A Uniform(0,1) random variate generated by RANDOM.
IOUT	INT	Output dump controller.
RAND	DRE	Random number seed.
WEIB	RE	The Weibull(β, $\eta(\beta)$) random variate generated by WEIBGN.

7.6.5.2 Subprogram WEIBGN Listing

```
C  THIS SUBROUTINE GENERATES WEIBULL(BETA,ETA) RANDOM VARIATES WITH
C  MEDIAN OF DISTRIBUTION CONSTRAINED TO BE ONE USING THE "INVERSE
C  TRANSFORM METHOD"
C  PROGRAMMER:  L. NEWLIN
C        DATE:  CODE:  18MAR87      COMMENTS:  15SEP89
C     VERSION:  MATCHR V4, V5, V5.1, V5.2, V5.3, V6, V6.1, V6.2,
C                      V7, V7.1, V8, V8.1, V8.2, V8.3, V8.4, V8.5
```

```
C                   MATGRM V2, V3, V3.1, V3.2, V3.3, V4, V4.1, V4.2,
C                          V4.3, V4.4, V4.5
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.

      SUBROUTINE WEIBGN (BETA, RAND, WEIB)

C  INPUTS:   BETA, RAND
C  OUTPUTS:  WEIB
C  SUBPROGRAMS:  RANDOM

C     IMPLICIT NONE

      COMMON  IOUT

      INTEGER IOUT

      REAL    ARG, BETA, ETA, FRAC, WEIB

      DOUBLE PRECISION RAND

C                   LIST OF VARIABLES
C
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
```

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

VARIABLE NAME	TYPE	DESCRIPTION
FRAC	RE	A Uniform(0,1) random variate generated by RANDOM.
IOUT	INT	Output dump controller.
RAND	DRE	Random number seed.
RHO1	RE	Lower bound *a* of the U(*a*,*b*) distribution.
RHO2	RE	Upper bound *b* of the U(*a*,*b*) distribution.
THE1	RE	Lower bound *c* of the U(*c*,*d*) distribution.
THE2	RE	Upper bound *d* of the U(*c*,*d*) distribution.
X	RE	The U(*a*,*b*) random variate generated by PRYRV.
Y	RE	The U(*c*,*d*) random variate generated by PRYRV.

7.6.6.2 Subprogram PRYRV Listing

```
C***********************************************************************
C   SUBROUTINE PRYRV GENERATES A PAIR OF U(RHO1,RHO2) AND U(THE1,THE2)
C   INDEPENDENT RANDOM VARIATES
C   PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C         DATE:  9MAR87
C   SUBPROGRAM:  RANDOM
C
C  Copyright (C) 1990, California Institute of Technology.
C  U.S. Government Sponsorship under NASA Contract NAS7-918
C  is acknowledged.
C***********************************************************************

      SUBROUTINE PRYRV (RAND, RHO1, RHO2, THE1, THE2, X, Y)

      COMMON   IOUT

      DOUBLE PRECISION RAND

      REAL     FRAC, RHO1, RHO2, THE1, THE2, X, Y

      INTEGER IOUT

      CALL RANDOM (FRAC, RAND)
C     IF (IOUT .EQ. 15) WRITE(8,*) 'FRAC =', FRAC
      X = FRAC * (RHO2 - RHO1) + RHO1

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

      IF (IOUT .EQ. 15) WRITE(8,*) 'RHO1 =', RHO1, ' RHO2 =', RHO2,
     &        ' THE1 =', THE1, ' THE2 =', THE2, ' X =', X, ' Y =', Y

      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.

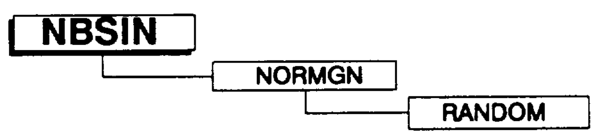

Figure 7-13 Tree Structure For Program NBSIN

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 7-23 List of Subprograms for Program NBSIN

NAME	SECTION	FUNCTION
NORMGN[1]	4.4.3	Generates two independent Normal(μ, σ^2) random variates.
RANDOM[2]	4.4.2	Uses a Linear Congruential random number Generator (LCG) to generate Uniform(0,1) random variates.
NBSIN	4.5.2	The main routine that controls the logical flow and performs the calculations of the time history generation.

1 The Normal distribution is discussed on *Page 2-23.*

2 The Uniform distribution is discussed on *Page 2-23.*

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 7-24 List of Variables for Program NBSIN
(Footnotes are at the end of the table)

VARIABLE NAME	TYPE	DESCRIPTION
A(MAXSIN)	DRE	1-D array containing the amplitudes of the sinusoidal processes.
CLIP	DRE	Peak-clipping level for the narrow-band processes.
DELTAT	DRE	The time increment Δt of *Equation 2-60*.
F	DRE	f (Hz) of *Equation 2-60*, the frequency used to determine the time increment, usually the value of $\max(f_o, f_c)$.
FC(MAXSIN)	DRE	1-D array containing values of the frequencies f_c (Hz) of the carrier (sinusoidal) processes.[1]
FILNUM(MAXHIS)	INT	1-D array containing the file unit numbers for the reference time history storage files.
FO(MAXRAN)	INT	f_o (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.
HISNAM(MAXHIS)	CH6	1-D array containing the HIStory NAMes for the time history files.
I	INT	Controls DO loop for each history and time increment.
IOUT	INT	Output dump controller.
J	INT	Controls DO loop for each sinusoidal process.
K	INT	Controls DO loop for each narrow-band process.
LASTT	DRE	Length T of the time histories to be generated.[2]
MAXHIS	INT	Maximum total number of time histories allowed. The maximum number of time histories allowed is 29.

Table 7-24 List of Variables for Program NBSIN (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
MAXRAN	INT	Maximum number of narrow-band time histories allowed. The maximum number of narrow-band histories allowed is 19.
MAXSIN	INT	Maximum number of sinusoidal time histories allowed. The maximum number of sinusoidal histories allowed is 10.
MAXTIM	INT	Maximum number of time increments allowed. The maximum number of increments is 25,000.
MU	DRE	Mean μ of the Normal distribution used to generate the narrow-band process. The value of μ must be zero.
N	DRE	N of *Equation 2-60*, the number of points per cycle of frequency f.
N1, N2	DRE	Normal(0, 1) random variates.
NCI(MAXRAN)	DRE	1-D array containing values of $N_c(t_i)$, *Equation 2-57*, the values of the cosine components of the narrow-band processes.
NCIOLD(MAXRAN)	DRE	1-D array containing values of $N_c(t_{i-1})$.[3]
NI(MAXRAN)	DRE	1-D array containing values of $N(t_i)$, *Equation 2-57*, the narrow-band processes.
NRAND(0:4)	INT	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(MAXRAN)	DRE	1-D array containing values of $N_s(t_i)$, *Equation 2-57*, the values of the sine components of the narrow-band processes.
NSIN	INT	The number of sinusoidal time histories.
NSIOLD(MAXRAN)	DRE	1-D array containing values of $N_s(t_{i-1})$.[4]
NTIM	INT	The number of time increments.
PHASE(MAXSIN)	DRE	1-D array containing the relative phase angle shifts (rad) of the sinusoidal processes.
RAND	DRE	Random number seed.
RHO(MAXRAN)	DRE	ρ of *Equation 2-60*, the 1-D array containing a value of the autocorrelation coefficient for each narrow-band process.
SHIFT	DRE	User-requested amount to shift all sinusoidal process phase angles (degrees).

Table 7-24 List of Variables for Program NBSIN (Cont'd)

VARIABLE NAME	TYPE	DESCRIPTION
SI(MAXSIN)	DRE	1-D array containing values of $S(t_i)$, the sinusoidal processes.
SIGMAC(MAXRAN)	DRE	1-D array containing values of σ_C the standard deviation of the cosine components of the narrow-band processes.
SIGMAN(MAXRAN)	DRE	1-D array containing values of σ_N, *Equation 2-60*, the standard deviation of the narrow-band processes.
TI(0:MAXTIM)	DRE	1-D array containing the time values, t_i.
TWOPI	DRE	Value of 2π.
UCI(MAXRAN)	DRE	1-D array containing values of $u_C(t_i)$ the Normal(0, σ_C^2) generated random variates for the cosine components of the narrow-band processes.[5]
USI(MAXRAN)	DRE	1-D array containing values of $u_S(t_i)$ the Normal(0, σ_S^2) generated random variates for the sine components of the narrow-band processes.[5] Note: $\sigma_S = \sigma_C$.
WC(MAXSIN)	DRE	1-D array containing values of the angular frequencies ω_C of the sinusoidal processes.
WO(MAXRAN)	DRE	ω_0 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.
XC	DRE	The damping coefficient ξ of *Equation 2-58*.
Z1(MAXRAN)	DRE	1-D array containing Normal(0, 1) random variates for the cosine component of the narrow-band process.
Z2(MAXRAN)	DRE	1-D array containing Normal(0, 1) random variates for the sine component of the narrow-band process.

---

1 f_c is discussed on *Page 2-30.*

2 T is discussed on *Page 2-30.*

3 See variable NCI( ).

4 See variable NSI( ).

5 See $u(t)$ in *Equation 2-60.*

7.7.1.4 Program NBSIN Listing

```
C   PROGRAM NBSIN CONTROLS THE GENERATION OF MULTIPLE SINUSOID AND
C   NARROW-BAND PROCESSES
C   PROGRAMMER:  L. NEWLIN
C         DATE:  CODE:  28JAN91      COMMENTS:  11SEP91
C      VERSION:  V5.4
C
C   Copyright (C) 1990, California Institute of Technology.
C   U.S. Government Sponsorship under NASA Contract NAS7-918
C   is acknowledged.

      PROGRAM NBSIN

C   SUBPROGRAMS:  NORMGN
C         FILES:  1:NBSIN-OLD;  8:IOUTPR-NEW;  11-39:user named-NEW

C     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),
     &          NCIOLD(MAXRAN), NI(MAXRAN), NSI(MAXRAN), NSIOLD(MAXRAN),
     &          PHASE(MAXSIN), RHO(MAXRAN), SHIFT, SI(MAXSIN), SIGMA,
     &          SIGMAC(MAXRAN), SIGMAN(MAXRAN), TI(0:MAXTIM),
     &          UCI(MAXRAN), USI(MAXRAN), WC(MAXSIN), WO(MAXRAN), 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 /


C                            LIST OF VARIABLES
C
C   A()         1-D ARRAY CONTAINING THE AMPLITUDES OF SINUSOIDAL PROCESSES
C   CLIP        PEAK-CLIPPING LEVEL
C   DELTAT      TIME INCREMENT
C   F           FREQUENCY USED TO DETERMINE TIME INCREMENT
C   FC()        1-D ARRAY CONTAINING FREQUENCIES OF SINUSOIDAL PROCESSES
C   FILNUM()    1-D ARRAY CONTAINING UNIT NUMBERS FOR TIME HISTORY FILES
C   FO()        1-D ARRAY CONTAINING FREQUENCIES OF NARROW-BAND PROCESSES
C   HISNAM()    1-D ARRAY CONTAINING HIStory NAMes FOR THE TIME HISTORIES
C   I           CONTROLS DO LOOP FOR EACH TIME INCREMENT
C   IOUT        OUTPUT DUMP CONTROLLER
C   J           CONTROLS DO LOOP FOR EACH SINUSOIDAL PROCESS
C   K           CONTROLS DO LOOP FOR EACH NARROW-BAND PROCESS
C   LASTT       LARGEST TIME VALUE TO BE CONSIDERED (LENGTH OF GENERATED
C                 TIME HISTORY)
C   MAXHIS      MAXIMUM TOTAL NUMBER OF TIME HISTORIES ALLOWED
C   MAXRAN      MAXIMUM NUMBER OF NARROW-BAND RANDOM HISTORIES ALLOWED
C   MAXSIN      MAXIMUM NUMBER OF SINUSOIDAL HISTORIES ALLOWED
C   MAXTIM      MAXIMUM NUMBER OF TIME INCREMENTS ALLOWED
C   MU          MEAN OF NARROW-BAND PROCESS -- EQUAL TO ZERO
C   N           NUMBER OF POINTS PER CYCLE OF FREQUENCY F
C   N1          NORMAL(0,1) RANDOM VARIATE
C   N2          NORMAL(0,1) RANDOM VARIATE
C   NCI()       1-D ARRAY CONTAINING VALUES OF Nc(i), THE MAGNITUDES OF
```

```
C                 THE COSINE COMPONENTS OF THE NARROW-BAND PROCESSES
C   NCIOLD()     1-D ARRAY CONTAINING VALUES OF Nc(i-1)
C   NI()         1-D ARRAY CONTAINING VALUES OF N(i), THE NARROW-BAND PROCESSES
C   NRAND()      1-D ARRAY CONTAINING THE NUMBER OF NARROW-BAND TIME HISTORIES
C                  FOR EACH CORRELATION GROUP WITH NRAND(0) BEING THE TOTAL
C                  NUMBER OF NARROW-BAND TIME HISTORIES
C   NSI()        1-D ARRAY CONTAINING VALUES OF Ns(i), THE MAGNITUDES OF
C                  THE SINE COMPONENTS OF THE NARROW-BAND PROCESSES
C   NSIN         NUMBER OF SINUSOIDAL HISTORIES
C   NSIOLD()     1-D ARRAY CONTAINING VALUES OF Ns(i-1)
C   NTIM         NUMBER OF TIME INCREMENTS
C   PHASE()      1-D ARRAY CONTAINING THE RELATIVE PHASE ANGLE SHIFTS OF THE
C                  SINUSOIDAL  PROCESSES (RADIANS)
C   RAND         RANDOM NUMBER SEED
C   RHO()        1-D ARRAY CONTAINING VALUES OF THE AUTOCORRELATION COEFFICIENT
C                  FOR EACH NARROW-BAND PROCESS
C   SHIFT        AMOUNT TO SHIFT ALL SINE PHASE ANGLES (DEGREES)
C   SI()         1-D ARRAY CONTAINING VALUES OF S(i), THE SINUSOIDAL
C                  PROCESSES
C   SIGMA        STANDARD DEVIATION OF NORMAL DISTRIBUTION -- EQUAL (CONSTRAINED)
C                  TO ZERO
C   SIGMAC()     1-D ARRAY CONTAINING VALUES OF STANDARD DEVIATION OF COSINE
C                  COMPONENTS
C   SIGMAN()     1-D ARRAY CONTAINING VALUES OF STANDARD DEVIATION OF
C                  NARROW-BAND PROCESSES
C   TI()         1-D ARRAY CONTAINING TIME VALUES, Ti
C   TWOPI        2 * PI
C   UCI()        1-D ARRAY CONTAINING NORMAL(0,SIGMAC) GENERATED RANDOM VARIATE
C                  FOR COSINE COMPONENTS
C   USI()        1-D ARRAY CONTAINING NORMAL(0,SIGMAS) GENERATED RANDOM VARIATE
C                  FOR SINE COMPONENTS -- SIGMAS = SIGMAC
C   WC()         1-D ARRAY CONTAINING ANGULAR FREQUENCIES OF THE SINUSOIDAL
C                  PROCESSES
C   WO()         1-D ARRAY CONTAINING ANGULAR FREQUENCIES OF NARROW-BAND
C                  PROCESSES
C   XC           DAMPING COEFFICIENT
C   Z1()         1-D ARRAY CONTAINING NORMAL(0,1) RANDOM VARIATES FOR THE COSINE
C                  COMPONENT OF THE NARROW-BAND PROCESS
C   Z2()         1-D ARRAY CONTAINING NORMAL(0,1) RANDOM VARIATES FOR THE SINE
C                  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, F, 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,*) HISNAM(K), SIGMAN(K), FO(K)
   10 CONTINUE

      DO 20 J = 1, NSIN
         READ(1,*) HISNAM(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

      IF (NRAND(4) .NE. 0) THEN
         CALL NORMGN (RAND, MU, SIGMA, N1, N2)
         DO 34 K = (1+NRAND(1)+NRAND(2)+NRAND(3)), NRAND(0)
            Z1(K) = N1
            Z2(K) = N2
   34    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
         DO 60 K = 1, NRAND(0)
            WRITE(8,*) 'RHO = ', RHO(K), ' SIGMAC = ', SIGMAC(K)
            WRITE(8,*) 'FO = ', FO(K), ' WO = ', WO(K)
            WRITE(8,*) 'NCIOLD = ', NCIOLD(K), ' NSIOLD = ', NSIOLD(K)
   60    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)
```

```
            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))
     &              + NSI(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

         IF (IOUT .EQ. 10) THEN
            WRITE(8,*) 'I = ', I, ' TI = ', TI(I)
            DO 121 K = 1, NRAND(0)
               WRITE(8,*) 'K = ', K, ' NI = ', NI(K)
               WRITE(8,*) 'UCI = ', UCI(K), ' USI = ', USI(K)
               WRITE(8,*) 'NCIOLD = ', NCIOLD(K), ' NSIOLD = ',
     &                    NSIOLD(K)
               WRITE(8,*) 'NCI = ', NCI(K), ' NSI = ', NSI(K)
 121        CONTINUE
         ENDIF

C **     UPDATE RECURSIVE PARAMETERS

         DO 140 K = 1, NRAND(0)
            NCIOLD(K) = NCI(K)
            NSIOLD(K) = NSI(K)
```

```
  140      CONTINUE

  100 CONTINUE
  150 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))
  160 CONTINUE

C ** OPEN FILES FOR SINUSOIDAL HISTORY STORAGE

      DO 170 J = 1, NSIN
         OPEN (FILNUM(J+NRAND(0)), FILE = HISNAM(J+NRAND(0)),
     &         STATUS = 'NEW')
  170 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))
  210    CONTINUE

C **     WRITE TO FILES

         DO 220 J = 1, NSIN
            WRITE(FILNUM(J+NRAND(0)),*) SI(J)
  220    CONTINUE

         IF (IOUT .EQ. 10) THEN
            DO 230 J = 1, NSIN
               WRITE(8,*) 'J = ', J, ' SI = ', SI(J)
  230       CONTINUE
         ENDIF

  200 CONTINUE

      STOP
      END


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


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:  14APR88
C    VERSION:  NBSIN V1, V1.1, V2, V3, V4, V5, V5.1, V5.3, V5.4
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, X1, X2)

C     IMPLICIT NONE

      COMMON  IOUT

      DOUBLE PRECISION RAND, MU, PI, SIGMA, X1, X2, U1, U2, Z1, Z2
```

```
      REAL    FRAC
      PARAMETER (PI = 3.1415926536)
      INTEGER IOUT

C             LIST OF VARIABLES
C
C  FRAC     UNIFORM(0,1) RANDOM VARIATE
C  IOUT     OUTPUT DUMP CONTROLLER
C  MU       MEAN OF NORMAL DISTRIBUTION
C  RAND     RANDOM NUMBER SEED
C  SIGMA    STANDARD DEVIATION OF NORMAL DISTRIBUTION
C  X1       NORMAL RANDOM VARIATE NUMBER ONE
C  X2       NORMAL RANDOM VARIATE NUMBER TWO
C  U1       UNIFORM RANDOM NUMBER U(0,1)
C  U2       UNIFORM RANDOM NUMBER U(0,1)
C  Z1       NORMAL RANDOM NUMBER ON N(0,1)
C  Z2       NORMAL RANDOM NUMBER ON N(0,1)

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

      CALL RANDOM (FRAC, RAND)
      U1 = 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)) * DCOS(2.D0 * PI * U2)
      Z2 = DSQRT (- 2.D0 * DLOG(U1)) * DSIN(2.D0 * PI * U2)

      X1 = SIGMA * Z1 + MU
      X2 = SIGMA * Z2 + MU
      IF ((IOUT .EQ. 10) .OR. (IOUT .EQ. 15))
     &   WRITE(8,*) 'Z1 =', Z1, ' Z2 =', Z2, ' X1 =', X1, ' X2 = ', X2

      RETURN
      END


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


C**********************************************************************
C     SUBROUTINE RANDOM USES AN LCG RANDOM NUMBER GENERATOR TO GENERATE
C     UNIFORMLY DISTRIBUTED RANDOM NUMBERS -- R.F. MILES, JPL
C
C     Miles, R. F., The RANDOM Computer Program: A Linear Congruential
C       Random Number Generator, JPL Publication 85-98, JPL Document
C       5101-277, Feb. 15, 1986.
C
C     PROGRAMMER:  L. GRONDALSKI, L. NEWLIN
C           DATE:  1DEC87
C**********************************************************************

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

C              LIST OF VARIABLES
C
C  FRAC       UNIFORM (0,1) RANDOM VARIATE
```

```
C  IOUT       OUTPUT DUMP CONTROLLER
C  RANA       CONSTANT FOR LCG
C  RANC       CONSTANT FOR LCG
C  RAND       RANDOM NUMBER SEED
C  RANDIV     INTERNAL CALCULATION
C  RANM       CONSTANT FOR LCG
C  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 = 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
```