ORNL-TM-1735
NASA CR- 72172



Contract No. W-7405-eng-26



CHEMICAL TECHNOLOGY DIVISION

# A PROGRAM FOR CALCULATING OPTIMUM DIMENSIONS OF ALPHA RADIOISOTOPE CAPSULES EXPOSED TO VARYING STRESS AND TEMPERATURE

J. P. Nichols
D. R. Winkler





These evaluation studies at ORNL were supported by the National Aeronautics and Space Administration, Lewis Research Center, under an Interagency Agreement with the Atomic Energy Commission – C-64217-A.

APRIL 1967

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
UNION CARBIDE CORPORATION
for the
U. S. ATOMIC ENERGY COMMISSION

CONTENTS

# A PROGRAM FOR CALCULATING OPTIMUM DIMENSIONS OF ALPHA RADIOISOTOPE CAPSULES EXPOSED TO VARYING STRESS AND TEMPERATURE

J. P. Nichols
D. R. Winkler

## ABSTRACT

A method and computer program were developed for calculating the creep and optimizing the dimensions of capsules filled with alpha-emitting radioisotopes. The method solves an integral equation that was developed assuming linear accumulation of partial creep lives and relating life to time-dependent stress and temperature using the Larson-Miller parameter. The computer program, CAPSUL, is written in Fortran language for the IBM 360/75 computer. The program makes a least squares fit of the creep life function using conventional constant stress, constant temperature creep data. Dimensions of capsules having maximum thermal power per unit of weight, volume, or area are calculated for a given creep life and pressure-temperature history using a numerical Lagrange Multiplier formulation. The program also calculates the life to a prescribed strain for capsules of given dimensions and pressure-temperature history. The method has been used to analyze creep data for the alloys 304 stainless steel, Hastelloy N, Cb-1% Zr, FS-85, and T-222.

---

## 1.0 INTRODUCTION

In capsules containing alpha-emitting radioisotopes for use in space power packages, it is desirable to provide maximum power per unit of weight, volume, or projected area within the constraints imposed by the need to maintain capsule integrity during normal operation and in the event of one or more accidental conditions. Because of the continuous generation of helium gas, together with decay of the thermal power, such capsules are characterized by time-dependent stress and temperature. Very high initial temperatures cause creep to be an important consideration in the design.

A model and computer program were formulated for calculation of the strain and optimum dimensions of capsules within the desired constraints.

BLANK PAGE

The following sections will describe the model and the computer program and present an analysis of experimental creep data that tend to confirm the model. A glossary of symbols, example problems, and a program list are included as appendices.

## 2.0 MATHEMATICAL MODEL

We wish to develop a phenomenological model of creep resulting from time-varying stress and temperature in certain metals for which the only available experimental data are ultimate strength properties at low temperatures and constant load, constant temperature creep properties at high temperatures. A precise analysis of the problem requires an equation of state that relates strain rate to stress, temperature, strain, and time. No single equation is available, but approximate equations may be developed for restricted classes of materials. One such equation, which has been substantiated for a number of metals[1,2] and plastics,[3] assumes that the fractional creep life for a given stress and temperature is independent of other fractions sustained under different conditions and that these fractions may be accumulated linearly. Stated mathematically:

$$1 = \int_0^{\Theta} \frac{dt}{\theta[\sigma(t),\ T(t)]} \tag{1}$$

where

$\theta[\sigma,T]$ = a function, hereafter called the "creep life function," that determines the life to a prescribed strain or rupture for a given stress, $\sigma$, and temperature, T,

t = time since application of the load,

$\Theta$ = resultant life to prescribed strain or rupture for time-varying stress and temperature.

A suitable creep life function can be determined by empirically fitting an equation to experimental data for stress as a function of a time-temperature parameter. Several time-temperature parameters, including those of Orr, Sherby, and Dorn; Manson and Haferd; and Larson and Miller, have been evaluated for this purpose; and it was found that the Larson-Miller

parameter provided the most accurate correlation for a wide selection of metals.[4] Larson and Miller[5] have related the creep life at a given stress to the absolute temperature by an equation that may be derived from the Arrhenius rate law,

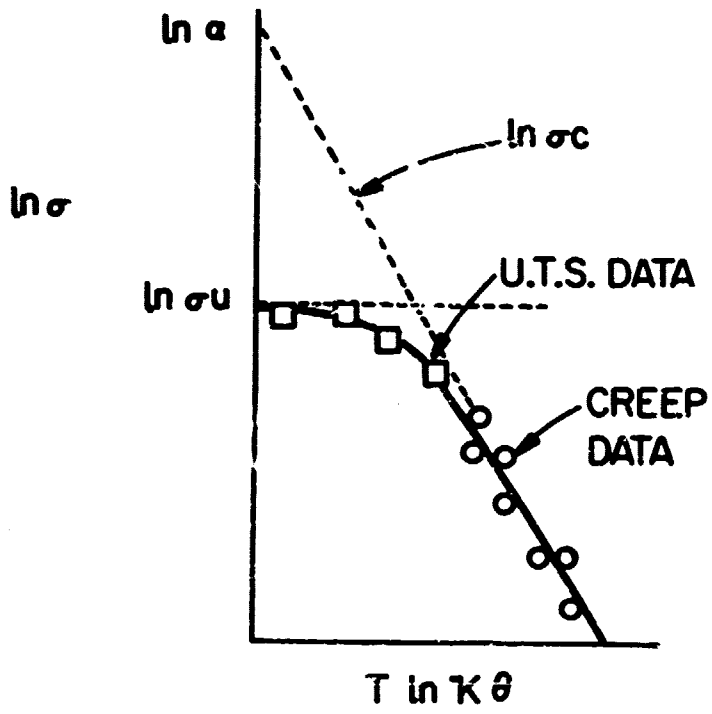
$$T \ln \kappa\theta = \text{constant, Larson-Miller parameter} .$$

The constant $\kappa$ has the physical connotation of "maximum rupture rate." The common logarithm of $\kappa$ is called the Larson-Miller constant and is in the range of 10 to 30 for most metals when time is measured in hours.

The development of the creep life function for the present model is illustrated in Fig. 1. Shown is a typical plot relating the logarithm of measured nominal, uniaxial stress to the Larson-Miller parameter. The creep data are obtained at temperatures generally above one-third the absolute melting temperature by measuring the time to a specified strain or rupture under conditions of constant load (constant nominal stress) and temperature. The data for ultimate strength (or stress for a specified "instantaneous" elastic and plastic strain) as a function of temperature are determined under conditions of constant imposed strain rate which is not necessarily the "natural" strain rate measured in creep experiments. We have assumed that the actual life measured in ultimate strength tests is a good approximation of the equivalent creep life, particularly for those materials which exhibit little strain hardening (nearly constant nominal stress for nominal strain greater than the yield), because (1) the life approximates unity and has little effect on the $\kappa\theta$ product, and (2) the stress is insensitive to the Larson-Miller parameter in the low-temperature range at which the ultimate strength data are used to supplement creep data.

As temperature increases the creep (or time dependent) strain for a given finite life becomes a progressively larger fraction of the total strain and the fractional strain from elastic and "instantaneous" plastic strain becomes progressively more negligible. At high temperatures the logarithm of the applied stress for many materials[4,5] is a linear function of the Larson-Miller parameter having intercept $\ln \alpha$ and slope m. We have assumed that the creep component of stress, $\sigma_c$, has this linear form at all temperatures.

ORNL DWG 66-11005 R1




1. For T large, $\sigma \cong \sigma c$

$$\ln \frac{\sigma c}{\alpha} = -m\,(T \ln K\theta)$$

2. For T small, $\sigma \cong \sigma u$

$$\frac{1}{\sigma^{\gamma}} = \frac{1}{\sigma u^{\gamma}} + \frac{1}{\sigma c^{\gamma}}$$

$$\sigma = \left[\frac{\alpha^{\gamma}\sigma u^{\gamma}}{\alpha^{\gamma} + \sigma u^{\gamma}(K\theta)^{\gamma mT}}\right]^{\frac{1}{\gamma}}$$

$$\theta = \frac{1}{K}\left(\frac{\alpha}{\sigma\sigma u}\right)^{\frac{1}{mT}} (\sigma u^{\gamma} - \sigma^{\gamma})^{\frac{1}{\gamma mT}}$$

Fig. 1. Derivation of $\theta$ ($\sigma$, T).

At temperatures below approximately one-third the absolute melting temperature the creep strain becomes negligible with respect to the elastic and plastic strain. In this region the stress is generally constant, $\sigma_u$, over a large domain of the parameter. This behavior, as well as the behavior at high temperatures, is accommodated by a resultant stress function that is generated by adding reciprocals of the time dependent (creep) and time independent components of stress. An empirically fitted constant, $\gamma$, provides for appropriate curvature in the transition region.

The computer program, to be described in the next section, has provision for determination of the constants, $\alpha$, m, and $\kappa$ by least squares analysis of a set $[\sigma, T, \theta]$ of creep data. The constants $\sigma_u$ and $\gamma$ are selected by the investigator by analysis of a curve of ultimate strength as a function of temperature and/or by iteration to determine the best fit of combined ultimate strength-creep data in a time-temperature domain of interest.

The general formulation of the approximate equation of state is obtained by substituting the creep life function (Fig. 1) in Eq. (1) and making provisions for individual safety factors on ultimate strength, $S_u$, and creep strength, $S_c$.

$$1 = \kappa \int_0^{\Theta} \left\{ \frac{S_c^{\gamma}\, \sigma_u^{\gamma}\, \sigma^{\gamma}(t)}{\alpha^{\gamma}[\sigma_u^{\gamma} - S_u^{\gamma}\, \sigma^{\gamma}(t)]} \right\}^{\frac{1}{\gamma m \Gamma(t)}} dt = D \qquad (2)$$

In principle, the integral in Eq. (2) can be evaluated numerically for any time behavior of stress and temperature. For radioisotope fuel capsules we have chosen to neglect the effect of strain on the volume of the capsule and reinforcing effects by layers other than the primary structural material. End effects are also neglected since the capsules have length-to-diameter ratio greater than 2. The stress is considered to be the maximum nominal tensile stress, the circumferential stress at the inner wall of the primary container

$$\sigma(t) = \frac{P(t)}{E}\left[\frac{R(4)^2 + R(3)^2}{R(4)^2 - R(3)^2}\right] \qquad (3)$$

where

P = pressure in the capsule,

E = weld efficiency,

R(4) = outer radius of primary container,

R(3) = inner radius of primary container.

The time-dependent pressure is calculated from the ideal gas law since the helium gas is well above the critical point. The volume of gas is the void volume at the center of the capsule plus any additional void in the fuel region. The moles of gas are those that are present initially plus those that are formed by decay of the radioisotope.

The heat flux is in transient equilibrium with the power in capsules containing long-lived radioisotopes. Assuming that the overall heat transfer coefficients do not vary appreciably with temperature, the temperature of the helium gas and the container wall vary with time in the following way:

$$T(t) = Ta + (T^{\bullet} - Ta)e^{-\lambda t} , \qquad (4)$$

where

$T^{\bullet}$ = initial temperature,

Ta = ambient temperature,

$\lambda$ = decay constant.

Explicit formulae for the individual temperatures, volumes, etc., are given in the following section which describes the program, CAPSUL.

Equation (2) reduces to one previously derived by Kennedy[6] under conditions of high constant temperature and constant stress rate, $\dot{\sigma}$. High temperature implies $\sigma = \dot{\sigma}t \ll \sigma_u$; therefore, if the safety factors are unity, Eq. (2) reduces to:

$$1 = \kappa \int_0^{\Theta} \left( \frac{\dot{\sigma}t}{\alpha} \right)^{\frac{1}{mT}} dt = \kappa \, (\dot{\sigma}/\alpha)^{\frac{1}{mT}} \, \frac{\Theta^{\frac{1}{mT} + 1}}{\frac{1}{mT} + 1} , \qquad (5)$$

where $\frac{1}{mT}$ and $\alpha/\kappa^{mT}$ are the constants "n" and "A" used by Kennedy.

Kennedy,[6] and later McCoy,[7,8,9] verified that this equation is valid for several materials (including 304 and 309 stainless steels, T-111,

T-222, and Cb-1% Zr) by comparing creep rupture lives obtained at high constant temperatures and constant stress with those at the same temperature but constant stress rate. These data serve, indirectly, to validate Eq. (2). The use of the CAPSUL program to re-evaluate these constant stress rate experiments by direct numerical integration is described in Sect. 4.3. The adequacy of the model can be tested for other materials and at lower temperatures by performing experiments in which stress and temperature are known functions of time and evaluating the integral in Eq. (2) either analytically or numerically.

## 3.0 CAPSUL PROGRAM

CAPSUL is a Fortran program for the IBM 360/75 computer. The program calculates the life to a prescribed creep strain and optimum dimensions of alpha radioisotope fuel capsules exposed to varying stress and temperature. The capsules (Fig. 2) are right cylinders with multilayered walls and elliptical end caps. Independent variables are RO, the inside radius of the capsule; X(2), the thickness of the fuel layer; and X(4), the thickness of the primary container wall.

The program has eight principal and eleven subsidiary subroutines. LSTSQ determines constants ($\alpha$, $\kappa$, m) in an equation for rupture life (or life to a prescribed strain) as a function of stress and temperature by a least squares fit of creep-ultimate strength data. Once the constants are determined, LSTSQ is normally bypassed for calculations with the same material and design life criterion. The subroutine MAX[10] uses a numerical Lagrange multiplier formulation to find a maximum of one of three thermal power functions (thermal power per unit projected area of a flat array of capsules, per unit volume of a rectangular parallelepiped that encloses a capsule and its auxiliary structural material, or per unit weight of capsule, each calculated in subroutine WR) subject to a time-integrated stress-temperature constraint (subroutine DR) that is dictated by a prescribed rupture or strain life. The subroutine RZERO calculates the allowable inside radius as a function of thermal power, if it is required that the capsule surface temperature not exceed a given value if the capsule is buried in an infinite medium of earth. The subroutine LIMIT examines the

T-222, and Cb-1% Zr) by comparing creep rupture lives obtained at high constant temperatures and constant stress with those at the same temperature but constant stress rate. These data serve, indirectly, to validate Eq. (2). The use of the CAPSUL program to re-evaluate these constant stress rate experiments by direct numerical integration is described in Sect. 4.3. The adequacy of the model can be tested for other materials and at lower temperatures by performing experiments in which stress and temperature are known functions of time and evaluating the integral in Eq. (2) either analytically or numerically.

## 3.0 CAPSUL PROGRAM

CAPSUL is a Fortran program for the IBM 360/75 computer. The program calculates the life to a prescribed creep strain and optimum dimensions of alpha radioisotope fuel capsules exposed to varying stress and temperature. The capsules (Fig. 2) are right cylinders with multilayered walls and elliptical end caps. Independent variables are RO, the inside radius of the capsule; X(2), the thickness of the fuel layer; and X(4), the thickness of the primary container wall.

The program has eight principal and eleven subsidiary subroutines. LSTSQ determines constants ($\alpha$, $\kappa$, m) in an equation for rupture life (or life to a prescribed strain) as a function of stress and temperature by a least squares fit of creep-ultimate strength data. Once the constants are determined, LSTSQ is normally bypassed for calculations with the same material and design life criterion. The subroutine MAX[10] uses a numerical Lagrange multiplier formulation to find a maximum of one of three thermal power functions (thermal power per unit projected area of a flat array of capsules, per unit volume of a rectangular parallelepiped that encloses a capsule and its auxiliary structural material, or per unit weight of capsule, each calculated in subroutine WR) subject to a time-integrated stress-temperature constraint (subroutine DR) that is dictated by a prescribed rupture or strain life. The subroutine RZERO calculates the allowable inside radius as a function of thermal power, if it is required that the capsule surface temperature not exceed a given value if the capsule is buried in an infinite medium of earth. The subroutine LIMIT examines the

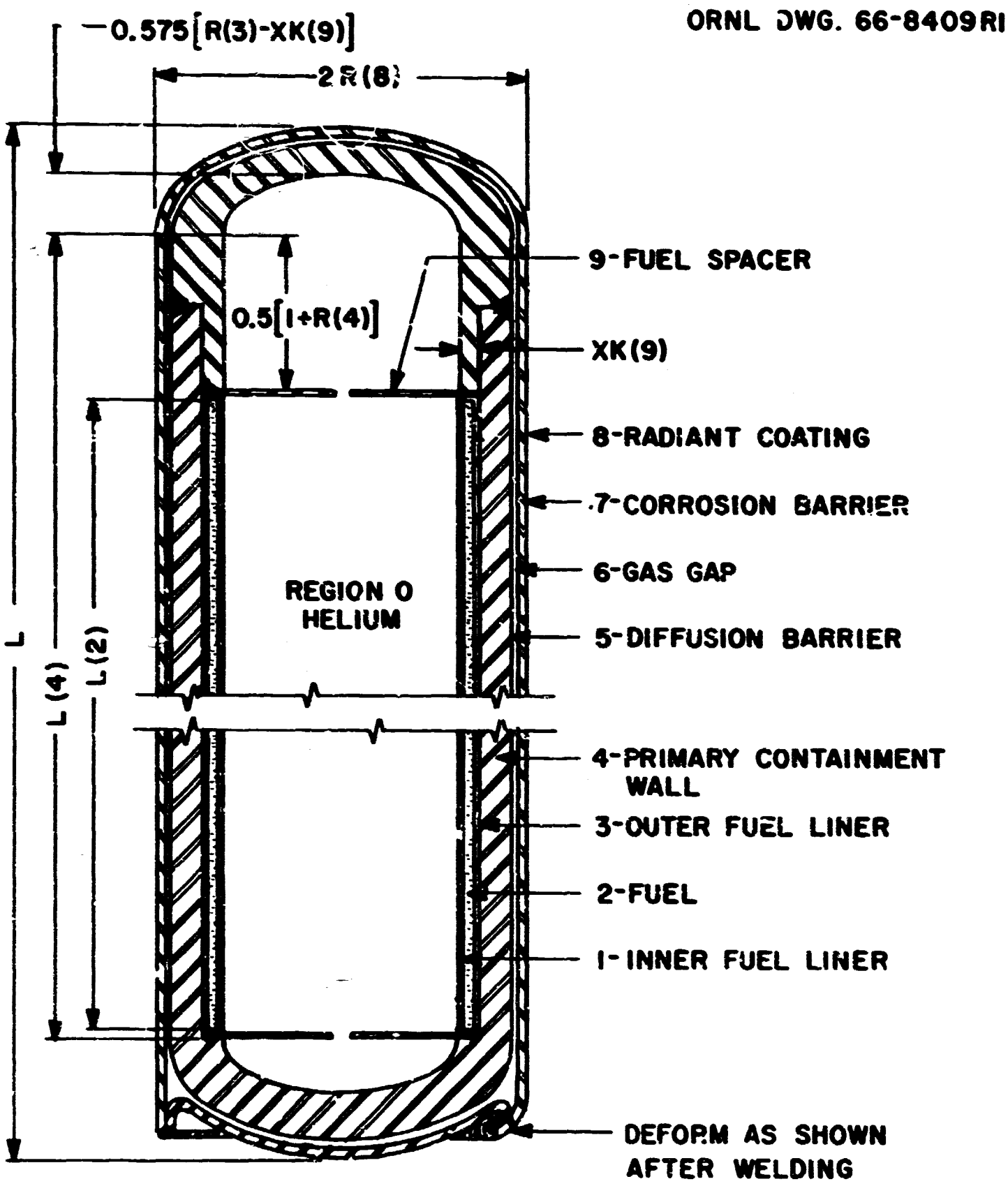



Fig. 2. Reference Design of Fuel Capsule

dimensions of the optimized capsules for adherence to limits dictated by engineering considerations. If the wall thickness or radius of the capsule is too small or too large, the appropriate dimension is fixed at its nearest limit and MAX or the subroutine DF is used to determine the maximum power function in the remaining variable(s). The subroutine DF calculates a single remaining variable to satisfy the stress-temperature constraint. The program subroutine THETAC calculates the rupture life or life to a prescribed strain of a capsule with specified dimensions. It is used to determine the life of capsules that have been optimized on the basis of another criterion. The subsidiary programs are WHT, which calculates the weight of the fuel capsule; CONVERT and VSU, which convert the capsule dimension variables to and from the MAX nomenclature; and SETUP, GTAIAM, VECT, CONVG, OUTPUT, ARITH, MATQ, and STEP, which are used by MAX. Library subroutines are SQRT, ABS, ALOG, ALOG10, SETFAULT, and EXP.

The main program, CAPSUL, reads and writes all input data, guides the selection of subroutines for prescribed options, and prints pertinent results. The program operates in any of four sequences. Each sequence selection requires a complete complement of data cards and is termed a "case." There is no upper limit on the allowable number of cases per run.

The sequence for a given case is determined by an input integer, MOPT. The first sequence (MOPT = 1) is used to provide a fit of the creep life function (Fig. 1) by a least squares analysis of creep data. The second sequence (MOPT = 2) fits the creep life function and calculates optimum capsule dimensions for a prescribed life. The third sequence (MOPT = 3) calculates optimum capsule dimensions when the constants of the creep life function are given as input. The fourth sequence (MOPT = 4) calculates the life (or safety factor for a prescribed life) of a capsule with given dimensions and material properties. The following sections will describe the formulation of the subroutines for these sequences and provide input information. A list of the program is given in Appendix C.

### 3.1 Least Squares Analysis of Creep Data (MOPT = 1)

This sequence begins with the reading of the constants MOPT, K, SIGU ($\sigma_u$), and GAMMA ($\gamma$). Next, the programs reads and stores information

from K data cards containing K triples [SIGMA(I), T(I), THETA(I)] of creep and/or ultimate strength data. Values of SIGMA(I) greater than SIGU are not permitted. The main program calls subroutine LSTSQ (which, in turn, calls MATQ) and values of the fitted constants (ALPHA, XM, XKO) are computed. The main program then computes the following quantities:

1. The value of stress, SIGB(I), predicted by the fitted creep life function for each pair [T(I), THETA(I)].

2. The value of life, THETB(I), predicted by the fitted creep life function for each pair [SIGMA(I), T(I)].

3. The Larson-Miller parameter, LMP(I) = T(I) $\log_{10}$ [(XKO)(THETA(I))], for each triple of data.

4. DLTH(I), the common logarithm of the ratio THETA(I) to THETB(I).

5. DLSG(I), the common logarithm of the ratio SIGMA(I) to SIGB(I).

6. SELTH, SELSG, and RESIG, the standard errors in the common logarithm of creep life, common logarithm of stress, and relative stress, respectively.

The quantities SELTH, SELSG, and RESIG are calculated as follows:

$$SELTH = \left[ \frac{1}{K - 3} \sum_{I=1}^{K} (DLTH(I))^2 \right]^{1/2}$$

$$SELSG = \left[ \frac{1}{K - 3} \sum_{I=1}^{K} (DLSG(I))^2 \right]^{1/2}$$

$$RESIG = \left[ \frac{1}{K - 3} \sum_{I=1}^{K} \left( \frac{SIGMA(I) - SIGB(I)}{SIGB(I)} \right)^2 \right]^{1/2}$$

The program prints the values K, SIGU, GAMMA, ALPHA, XM, XKO, SELTH, SELSG, and RESIG and the array I, SIGMA(I), T(I), THETA(I), LMP(I), THETB(I), SIGB(I), DLTH(I), DLSG(I).

The program then reads another data card containing the integer K. If the new value of K is greater than the previous value, the program reads an additional number of triples of creep data equal to the difference in the two values of K. These new data are stored together with the previous data and the entire calculational procedure is repeated. This procedure, which permits a sequential analysis of data which are ordered with respect to one of the variables [usually THETA(I)], continues until the program reads a card with K = 0. After reading a card with K = 0, the program proceeds to the next case.

Execution time of this sequence is less than one minute for analysis of 500 (the maximum allowable) number of triples of creep data that are read in at a single time.

### 3.1.1 LSTSQ

This subroutine prepares the elements of a matrix equation for determination of the constants ALPHA($\alpha$), XM(M), and XKO($\kappa$) by a least squares analysis of the creep life function (Fig. 1). The function is linearized by writing it in logarithmic form (see Appendix C). This procedure is an approximation in the sense that the sum of the squares of the residuals of the logarithms are minimized rather than those of the original variables.

### 3.1.2 MATQ

This subroutine solves the matrix equation AX = Y for X using modified Gaussian elimination (pivotal reduction using column pivots). A CO-OP description of this subroutine is given by Clark and Kam.[10]

### 3.1.3 Input Information

The sequence, MOPT = 1, uses only the data cards of type "a", "b", and "c", shown in Table 1. The cards of type "a" and "b" are followed by K cards of type "c". For sequential analysis the initial set is followed by stacks having a single card of type "b" followed by K'-K cards of type "c". Here, K' and K are the present and immediately preceeding values of K, respectively. The last card of type "c" in a case

Table 1. The Format of Input Data Cards for the CAPSUL Program

| Card Type | Format | MOPT | Data |
|---|---|---|---|
| a | 3I2 | 1,2,3,4 | MOPT, NMAX, NQ |
| b | I3,2F10.0 | 1,2 | K, SIGU, GAMMA |
| c | 3F20.0 | 1,2 | SIGMA(I), T(I), THETA(I) |
| d | 8F10.0 | 2,3,4 | GAMMA, RR, E, PS, TS, H, ZM, TA |
| e | 8F10.0 | 2,3,4 | T8, A, ETA, BETA, DELTA, C, G, TA1 |
| f | E10.3, 2I3 | 2,3,4 | LAMDA, N, NN |
| g | 9F8.0 | 2,3,4 | X(I), I = 1,9 |
| h | 9F8.0 | 2,3,4 | P(I), I = 1,9 |
| i | 9F8.0 | 2,3,4 | XK(I), I = 2,10 |
| j | 3E10.0 | 3.4 | ALPHA, YK, XM |
| k | 4F10.0 | 2,3 | X4U, X4L, R3U, R3L |
| l | 4E20.0 | 2,3,4 | SIGU, SC, SU, PHI |
| m | 4E20.0 | 2,3 | Q, DELRO, DELX2, DELX4 |
| n | 4E20.0 | 2,3 | TAU, T81, T82, XLD |
| o | 4E20.0 | 2,3 | RO, X(2), X(4), THET |
| p | 5I4 | 2,3 | IX, NLAM, NITER, ITER, LMOST |
| q | 4F10.0 | 2,3 | G, C, CRIT, ALAM |
| r | 4E20.0 | 4 | Q, DELTH, DELOME, THMAX |
| s | 4E20.0 | 4 | TAU, T81, XLD, RO |
| t | 3E20.0 | 4 | X(2), X(4), THET |

is followed by a card of type "b" with K = 0. The last case is followed by a card of type "a" with MOPT = 0.

### 3.2 Calculation of Capsule Dimensions for Maximum Specific Power (MOPT = 2 or 3)

The sequence begins by reading and writing MOPT, NMAX, and NQ plus 46 normally unchanging constants. If MOPT = 2, the program calls the entire sequence for MOPT = 1 to generate the three constants in the creep life function from experimental data; if MOPT = 3, these constants are read in and written. The program then reads and writes an additional 29 constants. Next, the program iterates (calling VSU and DR) to modify the initial estimates of X(2) and X(4) such that the constraining function is approximately satisfied ($0.3 \leq D \leq 4.0$). The iteration proceeds by multiplying the previous value of X(2) by 0.95 if D is too large or by multiplying the previous X(2) by 1.05 and the previous X(4) by 0.95 if D is too small. The calculation stops, prints pertinent data, and proceeds to the next case if an appropriate value of D is not determined in 100 iterations.

The subroutine VSU generates a set of independent variables [ALF(K) and their increments, DEL(K) and WEL(K)] for the MAX format from the capsule-dimension variables. The set of capsule-dimension variables to be used in a case is determined by the integer NMAX.

| | |
|---|---|
| NMAX = 1 | Variables are RO, X(2), and X(4). |
| NMAX = 2 | Variables are RO and X(2). |
| NMAX = 3 | The outer radius R(8) is specified. This option is used to generate capsules with given outside dimensions. |
| NMAX = 4 | The radius R(3) is specified. |
| NMAX = 5 | The radius RO is to be computed by the subroutine RZERO, assuming that the capsule is buried in an infinite medium. |

The subroutine DR performs a numerical integration of Eq. (2) and generates a value of the constraining function D for a current set of variables. The subroutine DR calls the subroutine CONVERT if NMAX $\neq$ 0 and NQ $\neq$ 0. The subroutine CONVERT reconverts the MAX variables to capsule dimensions, depending on the value of NMAX. If NMAX = 5, the subroutine CONVERT calls the subroutine RZERO to generate a burial-limited value of RO that is compatible with the current values of X(2) and X(4).

The main program calls the subroutine MAX if values of the variables are determined that approximately satisfy the constraining equation. MAX uses a numerical Lagrange multiplier formulation to find stationary values of the function W subject to the constraining equation D = C (C is the constrained value). The numerical technique seeks a minimum in a defined function YSQ by making successive linear approximations along the path of steepest descent. The function YSQ is the square of a vector that is zero when the constraining equation is satisfied and the gradients of the functions D and W are parallel, the condition for a local maximum or minimum in the function W. The function W is to be maximized in the present calculation; the minimum value is zero if X(2) = 0. The calculation proceeds by making outer and inner iterations. The outer iterations (counted by M) are steps in the domain of the function YSQ resulting from the linear approximation. Inner iterations (counted by LSTOP) prevent overstepping which might result in a divergent sequence.

The subroutine MAX calls the subroutines DR and WR to generate values of the constraining function D and the thermal power function W for use in tests and numerical computation of derivatives. The function W for a given case is chosen by the input integer NQ:

| | |
|---|---|
| NQ = 1 | Thermal power per unit of projected area (W = W1). |
| NQ = 2 | Thermal power per unit volume of a circumscribed rectangular parallepiped (W = W2). |
| NQ = 3 | Thermal power per unit of weight (W = W3). |

After each iteration, MAX writes current values of the following quantities:

| | |
|---|---|
| M | Number of the outer iteration. |
| D | Value of the constraining function. |
| W | Value of the thermal power function (W1, W2, or W3, depending on NQ) that is to be maximized. |
| YSQ | The function which has a value of zero at the desired solution point. The square of a vector Y. |
| Z | YSQ·$C^2$ (equal to YSQ for C = 1). |
| LSTOP | Number of the inner iteration within the outer iteration M. |
| Y(K) | Components of the solution vector Y. |
| ALF(K), K = 1,NR | Values of the independent variables and the Lagrange multiplier. |
| R(I), I = 0,9 | Outer radius of the nine regions of the capsule. |
| L | Length of the capsule. |
| AV2 | Thermal power of the capsule. |
| W1, W2, W3 | The specific thermal power functions. |
| WT | Weight of the capsule. |

The MAX calculation stops when either the function YSQ becomes smaller than a prescribed convergence criterion, $(CRIT)^2$, or a prescribed number of inner ($M \leq ITER$) and outer ($LMOST \leq LSTOP$) iterations is exceeded. It is recommended that the present type of calculation be stopped by the number of iterations since it is very difficult to predict an acceptable maximum value of the solution vector. The selection of a maximum allowable number of iterations has one disadvantage; it is often the case that variables determined in other than the last iteration provide a better solution of the problem. Since the properties of the capsule are completely described after each iteration, the "best" set of dimensions can be

determined by reviewing the printed matter and selecting the iteration for which (1) the constraint D = 1.00 is approximately satisfied, (2) the value W is maximum, and (3) YSQ is minimum.

When the MAX calculation stops, control is returned to the main program. The last set of calculated dimensions are then examined by the subroutine LIMIT to determine if the variables R(3) and X(4) are within the preselected domains R3L ≤ R(3) ≤ R3U and X4L ≤ X(4) ≤ X4U. If the variables [first X(4), then R(3)] are too small or too large, the appropriate dimension is fixed at its nearest limit; NMAX is changed to reflect a decrease in the number of independent variables; and control is returned to the main program. If the previous value of NMAX was 1 (new value 2 or 4), the entire calculation is repeated (starting with the iteration to determine suitable values of the variables and proceeding into the MAX subroutine) using the last computed values of RO and X(4) or X(2) and X(4) as initial estimates. If the previous value of NMAX was 2, 3, 4, or 5, the routine DF is called to calculate a value of the single remaining independent variable, X(2), that satisfies the constraining equation, DH = 1.00. When the calculation is completed, the main program writes a final list of the variables; these numbers are different from those computed in the last MAX iteration only if the program DF has been called.

The MAX calculational procedure does not insure convergence to the constrained maximum value of the function W. If the investigator is unsure of the neighborhood of the solution point, he should submit several cases with different initial estimates of the variables. If results are erratic, the tolerance limits on the functions (PHI and Q) may not be sufficiently small. Increments in the variables (DELRO, DELX2, and DELX4) must be chosen such as to cause only small changes in the functions.

The execution time of this sequence for 100 iterations (inner plus outer) with 100 initial increments (N) in the domain of integration is approximately 3.5 minutes. The execution time is approximately proportional to the product of ITER and N.

3.2.1 DR

Subroutine DR calculates the value, DH, of the constraining function, D, using Simpson's rule with N increments. An abbreviated formulation of the function, the more important internal dependent variables (the integrand and stress and temperature functions) follows. More specific formulations are relegated to the list of the subroutine (Appendix C). Symbols are defined in Appendix A.

$$DH = \int_{TNIT}^{TERM} F \, dt$$

$$F = \kappa \left[ \frac{S_c \sigma_u \sigma}{\alpha} \right]^{\frac{1}{mT4}} \left[ \sigma_u{}^{\gamma} - S_u{}^{\gamma} \sigma^{\gamma} \right]^{-\frac{1}{\gamma m T4}}$$

$$\sigma = \frac{RR}{E(V0 + \eta \cdot V2)} \left[ \frac{R(4)^2 + R(3)^2}{R(4)^2 - R(3)^2} \right] \left[ \frac{PS(V0 + \eta V2)}{RR \cdot TS} + \frac{H \cdot V2 \cdot P(2)}{XM} \left( 1 - e^{-\lambda \cdot T} \right) \right]$$

$$x \left[ TA + (T0 - TA) e^{-\lambda \cdot T} \right]$$

$$T4 = TA + (T40 - TA) e^{-\lambda T}$$

$$T40 = T8 + \frac{AV2}{2\pi \cdot L2} \sum_{J=4}^{8} \frac{1}{XK(J)} \left[ \frac{R(J)}{R(J+1)} \right]$$

$$T0 = T40 + \frac{AV2}{2\pi \cdot L2 \cdot XK(3)} \ln \frac{R(3)}{R(2)} + \frac{A}{4 \cdot XK(2)} \left[ R(2)^2 - R(1)^2 - 2 \cdot R(1)^2 \ln \frac{R(2)}{R(1)} \right]$$

DR doubles the number of increments in the domain of integration a maximum of 16 times to satisfy a convergence criterion, PHI. The input number PHI is compared to the ratio of the difference in the last two values of D to the last value. The statement "failed to converge" is

written if 16N increments are not sufficient. The library subroutine SETFAULT is used to set exponential underflows to zero.

The subroutine DR will accommodate functions other than those for which it is primarily intended (constant stress and/or temperature and constant derivatives of stress) by such devices as redefining the constants, setting T8 = TA, and dropping terms by setting leading constants equal to zero. In making such formulations, one must avoid negative arguments of logarithms (including values that are to be raised to a power) and division by zero.

### 3.2.2 WR

This subroutine calculates values of one of three specific thermal power functions for use as the function W in MAX. Choice of the function is determined by the current value of NQ. If the subroutine is called with NQ = 0, the values of all three thermal power functions are calculated. If NQ = 3, the subroutine calls subroutine WHT to calculate the capsule weight. The three functions are:

$$W1 = \frac{AV2}{4 \cdot XLD \cdot R(8) \cdot [R(8) + BETA]}$$

$$W2 = \frac{AV2}{8 \cdot XLD \cdot R(8) \cdot [R(8) + BETA] \cdot [R(8) + DELTA]}$$

$$W3 = \frac{AV2}{\sum_{I=1}^{9} V(I) \cdot P(I)}$$

### 3.2.3 DF

This subroutine uses Newton's method to calculate a value of X(2) to satisfy the constraint D = 1. The calculation proceeds until $D - 1 \leq Q$.

3.2.4 RZERO

This subroutine calculates RO from the following equation that relates the outside radius of a cylindrical capsule buried in an infinite conducting medium to the thermal power and temperature difference (difference between the maximum capsule surface temperature and ambient temperature of the conducting medium).

$$RO + \sum_{J=1}^{8} x(J) = \frac{AV2 \cdot \sinh^{-1}(XLD)}{4\pi \cdot XK(10) \cdot XLD \cdot (T82 - TA)}$$

In this equation AV2 (see list for subroutine WR) is a function of RO. The variable RO is initially estimated by an approximate formula and then calculated by iteration using Newton's method until the relative change in AV2 is less than $10^{-6}$ or 10 iterations have occurred. Convergence is normally accomplished in less than five iterations because the initial approximation is good and the derivatives are computed analytically. If the current values of the variables X(2) and X(4) are such that RO is negative, RO is set equal to 0.2.

3.2.5 LIMIT

This subroutine examines the last set of variables computed by MAX to determine if X(4) and R(3) are within the preselected domains $X4L \leq X(4) \leq X4U$ and $R3L \leq R(3) \leq R3U$. If the current value of NMAX is 1 and X(4) is too large or too small, the variable X(4) is set at its nearest limit and the entire MOPT sequence is repeated for NMAX = 2. If the current value of NMAX is 1 and $X4L \leq X(4) \leq X4U$, but R(3) is too large or too small, the variable R(3) is set at its nearest limit and the calculational sequence is repeated for NMAX = 4. If the current value of NMAX is 2 and R(3) is too large or too small, R(3) is set at its nearest limit and the independent variable X(2) is calculated to satisfy the constraint D = 1.00. If the current value of NMAX is 3, 4, or 5 and X(4) is too large or too small, X(4) is set at its nearest limit and X(2) is, again, calculated by the subroutine DF.

### 3.2.6 WHT

This subroutine calculates the radii, length, and weight of the fuel capsule using the current values of the variables RO, X(2), and X(4). The weight, WT, is calculated as the sum of the product of volume and density of each region.

### 3.2.7 VSU, CONVERT

These two subroutines convert the capsule dimension variables to and from the variables used in MAX. Depending on the current value of NMAX, VSU generates the variables ALF(K) from RO, X(2), and/or X(4) and the increments DEL(K) = WEL(K) from DELRO, DELX2, and/or DELX4. CONVERT reconverts to capsule dimensions and calls RZERO if NMAX = 5.

### 3.2.8 MAX

The MAX program, used as a subroutine in the CAPSUL program, was written by F. H. S. Clark and F. B. K. Kam of ORNL. The reader is referred to their report[10] or to the program list (Appendix C) for detailed information. The following will describe the sequence of calculations in MAX and a few changes that were made in the program for use in CAPSUL.

The main routine MAX begins by computing numbers that are to be used as convergence criteria and setting the outer iteration index M equal to one. Subroutine SETUP is then called. SETUP, which calls the subroutines DR and WR, produces values of the functions D and W and all their first and second derivatives at the current trial set of independent variables, ALF(K). When control returns from SETUP to MAX, a test is made on the input number NLAM. If NLAM = 0, the subroutine GTALAM is called to generate an initial estimate of ALAM, the Lagrange multiplier. A value of NLAM other than zero signals that the initial estimate of ALAM has been provided as input. Subroutine VECT is called next to generate components of the solution vector Y.

Subroutine CONVG is next called. This subroutine tests to determine if YSQ is less than the product of G ($< 1$) and its value in the previous iteration. If either (1) M is equal to one, (2) YSQ is less than the computed product, or (3) the input integer LMOST is equal to zero (a

change), an index, JWAY, is set equal to one. If these criteria are not satisfied, JWAY is set equal to zero and the components of the step that was last made in the domain of the function Y are multiplied by 0.5. The index JWAY is set equal to -1 if the number of these (JWAY = 0) inner iterations has exceeded the input integer, LMOST.

After CONVG, subroutine OUTPUT writes current values of the pertinent indices, variables, and functions and returns control to MAX. The MAX program then tests the index, JWAY, to determine whether to stop the case and proceed to the next one (JWAY = 1), to try an inner iteration with a reduced step in the domain of the function (JWAY = 0), or to proceed with convergence tests (JWAY = 1). If JWAY = 0, the function YSQ is reevaluated with the reduced increments until either the conditions for JWAY = 1 or -1 are met.

If JWAY = 1, tests are made to determine if the trial solution is converged or if the prescribed number (ITER) of outer iterations in M have been made. If either of these questions is answered affirmatively, the calculation is halted and input for the next case is called. Otherwise, subroutine ARITH is called. In this subroutine elements of a matrix (A) are evaluated at the new trial point. Next, subroutine MATQ is called. This solves for X the matrix equation (A)X = Y. The index M is increased by one and subroutine STEP is called, with subsequent operations following as previously described.

### 3.2.9 Input Information

The sequence for MOPT = 2 or 3 requires, in order, one each of the data cards "a" and "d" through "i" (Table 1). For MOPT = 2, these initial cards are followed by cards of type "b" and "c", stacked in the same order as for MOPT = 1; again, the last card of type "c" is followed by a card "b" with K = 0. The next card, of type "j", is included only if MOPT = 3. One each of the remaining data cards of type "k" through "q" then follows for either MOPT = 2 or 3. The last case is followed by a card of type "a" with MOPT = 0. The input constants T81, TA1, and NN are not used for MOPT = 2 or 3; consequently, these fields may be left blank.

### 3.3 Capsule Lifetime Analysis (MOPT = 4)

This sequence begins by reading and writing 67 input constants, the first of which is MOPT (NMAX and NQ must be zero). The program then calls the subroutine THETAC. This subroutine uses the subroutine DR and Newton's method with an initial estimate of life, THET, to calculate a value of the resultant life, THETA, that satisfies the constraint D = 1.00. In this case, D is the sum of two integrals. The first integral, over the time domain from zero to the input number TAU, uses the value T8 as the initial steady state temperature of the outer surface of the capsule and TA as the temperature of the environment. The second integral, over the time period TAU to THET, uses T81 as the initial (time zero) steady state temperature of the surface of the capsule and TA1 as the temperature of the environment. A value of TAU = 0 sets the first integral equal to zero.

If, on any iteration, THET is larger than an input constant, THMAX, and D is less than one, THET is set equal to THMAX and a new variable, OMEGA, is computed by Newton's method to satisfy the constraint D = 1. OMEGA is a number that multiplies the safety factors SC and SU.

After each iteration, the program writes the current values of either THETA, D, and the contribution of the second integral to D; or THMAX, D, and OMEGA. The calculation stops and returns control to the main program when D - 1 is less than the input number, Q.

Execution time of the program with N = NN = 100 is generally less than 20 seconds.

The sequence for MOPT = 4 requires, in order, one each of the data cards "a", "d" through "j", "l", and "r" through "t" (Table 1). The last case is followed by a card of type "a" with MOPT = 0.

## 4.0 RESULTS OF ANALYSIS OF CREEP DATA

Creep data for several alloys have been analyzed to confirm the applicability of the model used in the CAPSUL program. The following sections will present a statistical analysis of the predicted creep life functions for three commercial materials, an analysis of the errors in time extrapolations, and results obtained in predicting constant stress rate data from conventional, constant stress, creep data.

## 4.1 Predicted Creep Life Functions

Creep life functions for three commercial alloys (304 stainless steel, Hastelloy N, and Cb-1% Zr) were generated using the MOPT = 1 option of the CAPSUL program. The fitted creep life function for rupture of 304 stainless steel (Fig. 3) was made using 189 reduced creep data points[11] for 18 heats of bar and plate, covering the temperature domain from 900 to 1700°F and rupture life to 100,000 hr. The reported data, at decade intervals, were generated by interpolation or extrapolation of the larger body of experimental creep data for a given heat of material and temperature. Only three of the 100,000-hr data points were obtained from experiments terminated at approximately 100,000 hr. The other thirteen data points were obtained by extrapolation of data from experiments terminated at earlier times in the 10,000 to 100,000-hr decade. The constants $\gamma$ and $\sigma_u$ were chosen prior to the least squares analysis to force a fit of the ultimate strength vs temperature (0.1-hr rupture) data in the temperature domain above 400°F. A parametric plot of the derived creep rupture function (Fig. 4) shows that the fit is good over the entire range of the variables. The frequency distribution of the error (Fig. 5) in the logarithm of the measured life and stress with respect to values predicted by the fitted function is approximately gaussian. The distribution of the error in relative stress is also approximately normal; the relative standard error in stress is 0.17 (68% confidence level).

The fitted creep life function (Fig. 6) for rupture of Hastelloy N (INOR-8) was made using 93 data points[12,13] for five heats of rods, sheet, and plate, covering the domains in temperature from 1100 to 1800°F and rupture life to 14,400 hr. A parametric plot of the predicted function and data at four temperatures is shown in Fig. 7. In this case, the distribution of error (Fig. 8) is not gaussian, but the frequency peaks on both sides of the fitted function. This phenomenon is explained by the fact that approximately half of the data were obtained with the commercial Hastelloy N alloy which is somewhat stronger than the initial version of the alloy, which was called INOR-8. If it is assumed that the frequency distribution is gaussian, the relative standard error in stress is 0.16.

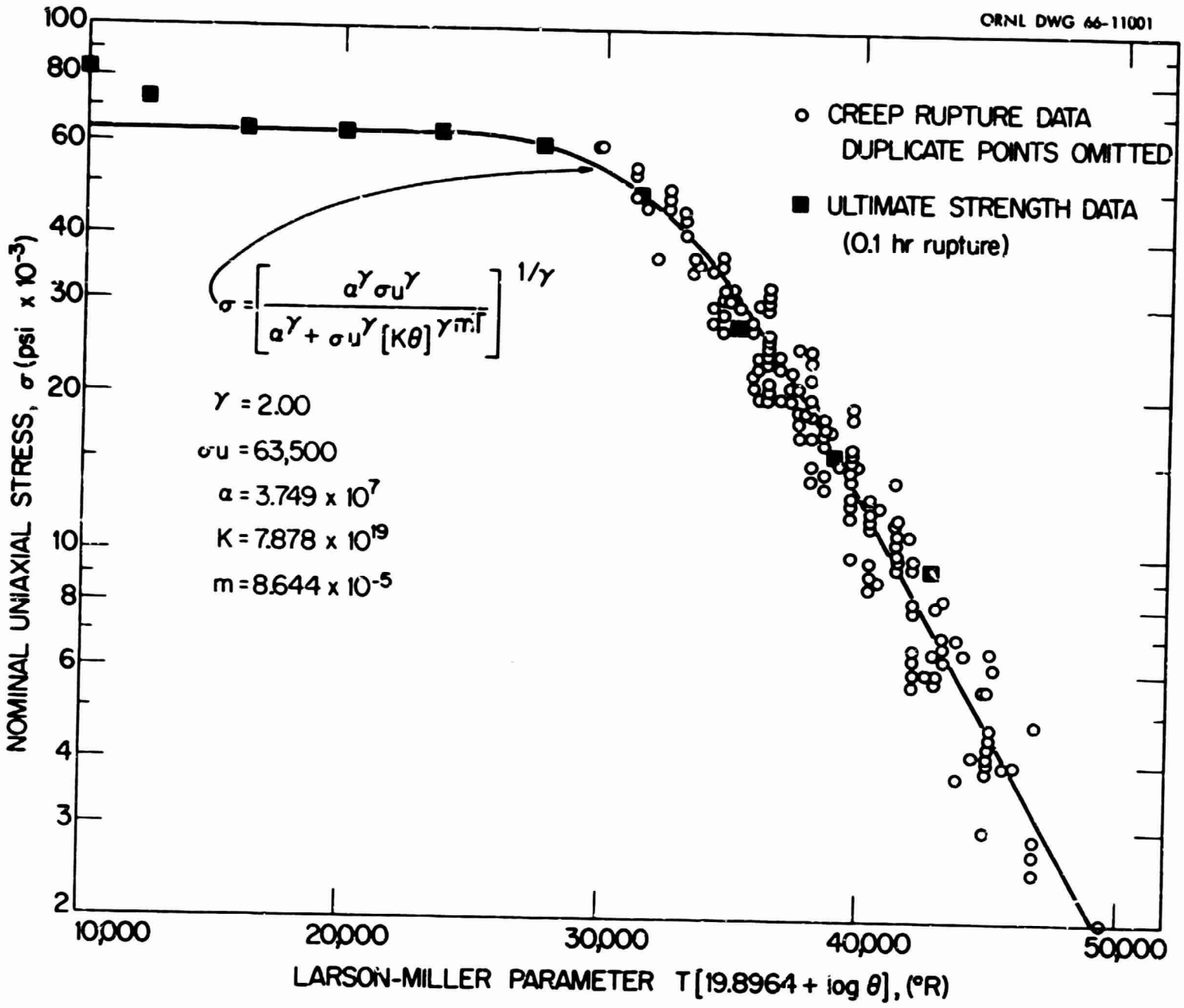



Fig. 3. Master Larson-Miller Plot for Creep Rupture in 304 Stainless Steel. 18 heats. 189 data points. $1360 \leq T \leq 2160$. $10 \leq \theta \leq 100{,}000$.

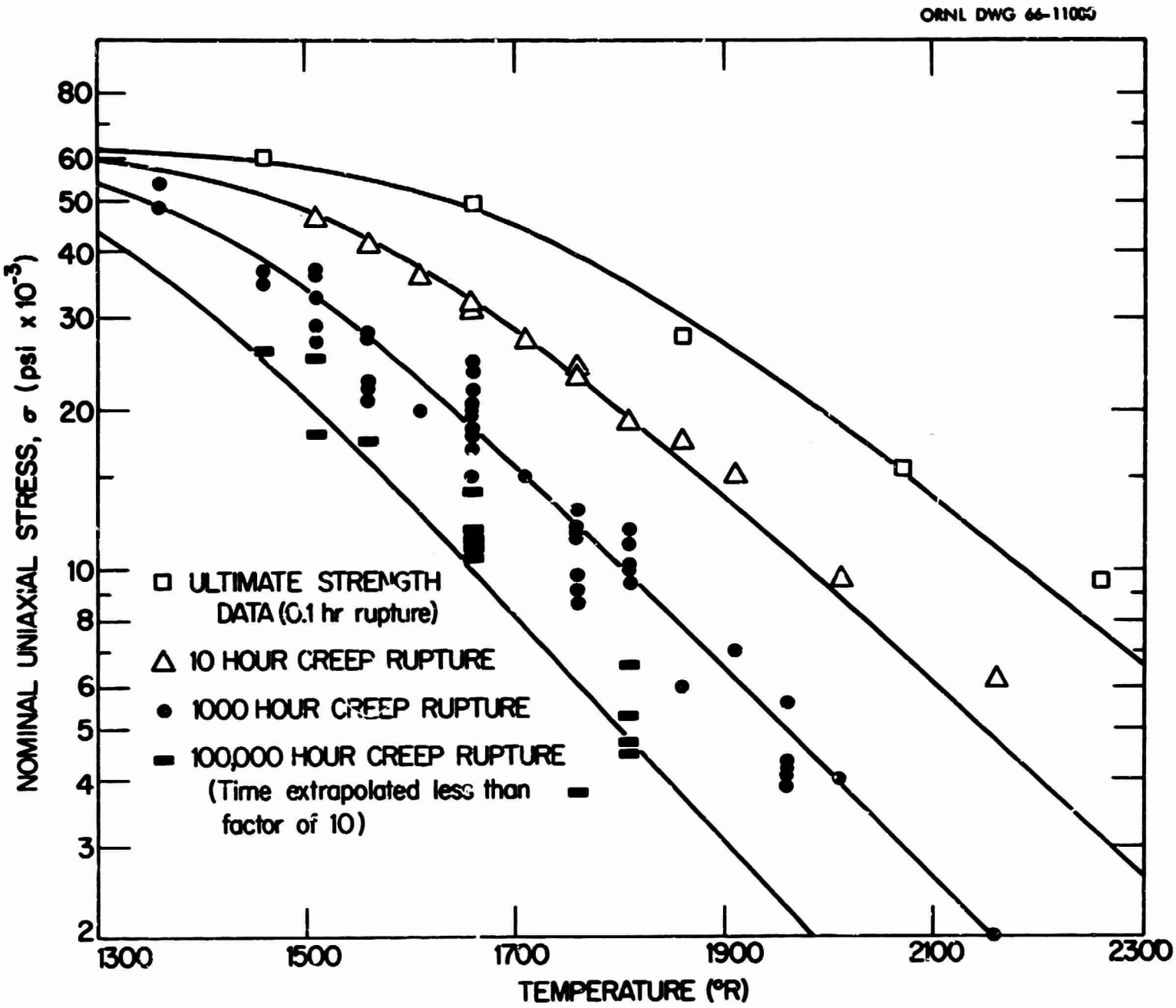



Fig. 4. Parametric Plot of the Derived Creep Rupture Function for 304 Stainless Steel Compared with Experimental Data. 18 heats of bar and plate. $1360 \leq T \leq 2160$. $\theta \leq 100{,}000$ hr.

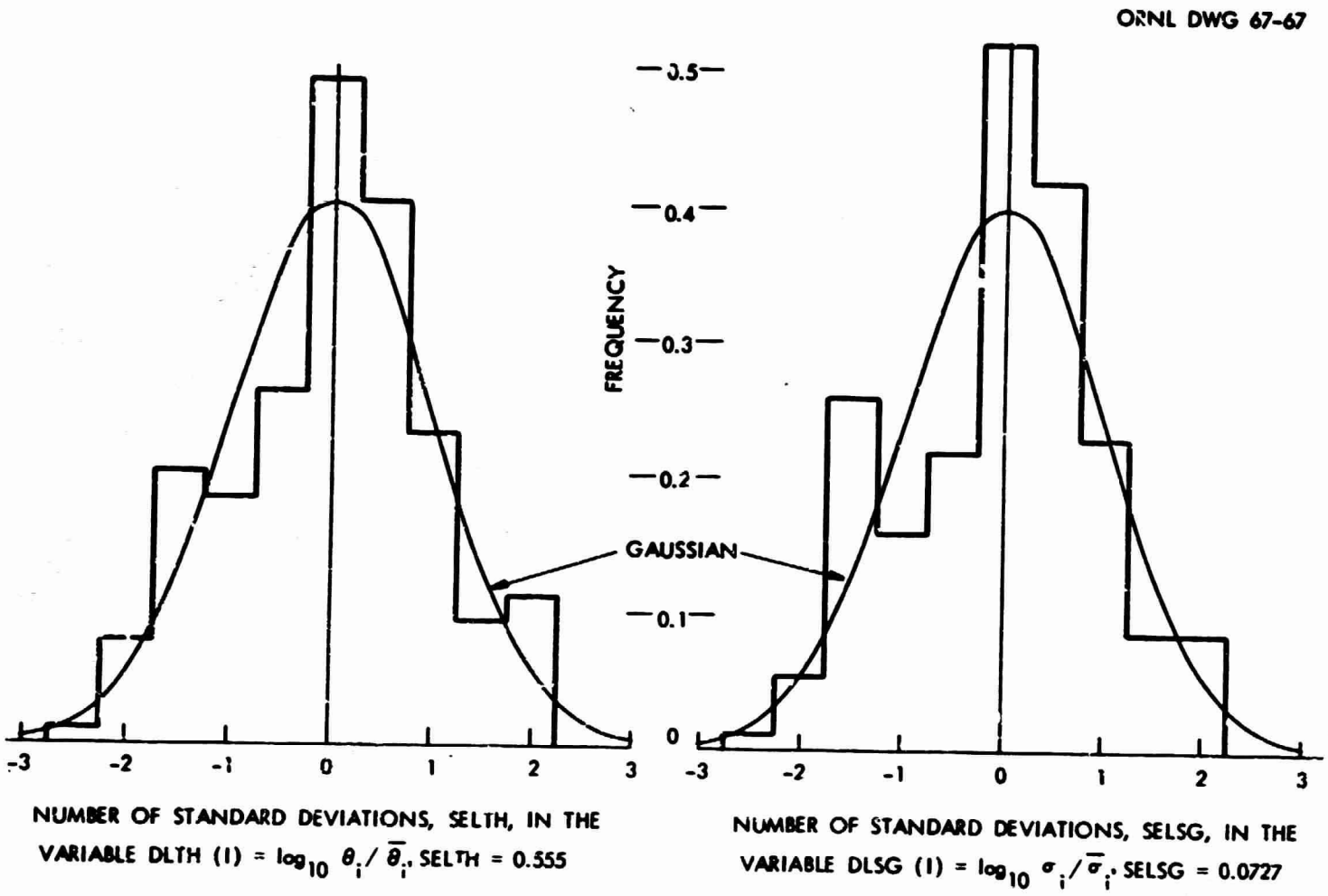



Fig. 5: Distribution of the Error in the Logarithm of Measured Rupture Life and Stress for 304 Stainless Steel with Respect to Values Predicted by a Least-Squares Fit of the Creep Rupture Function. Area normalized to 1.00.

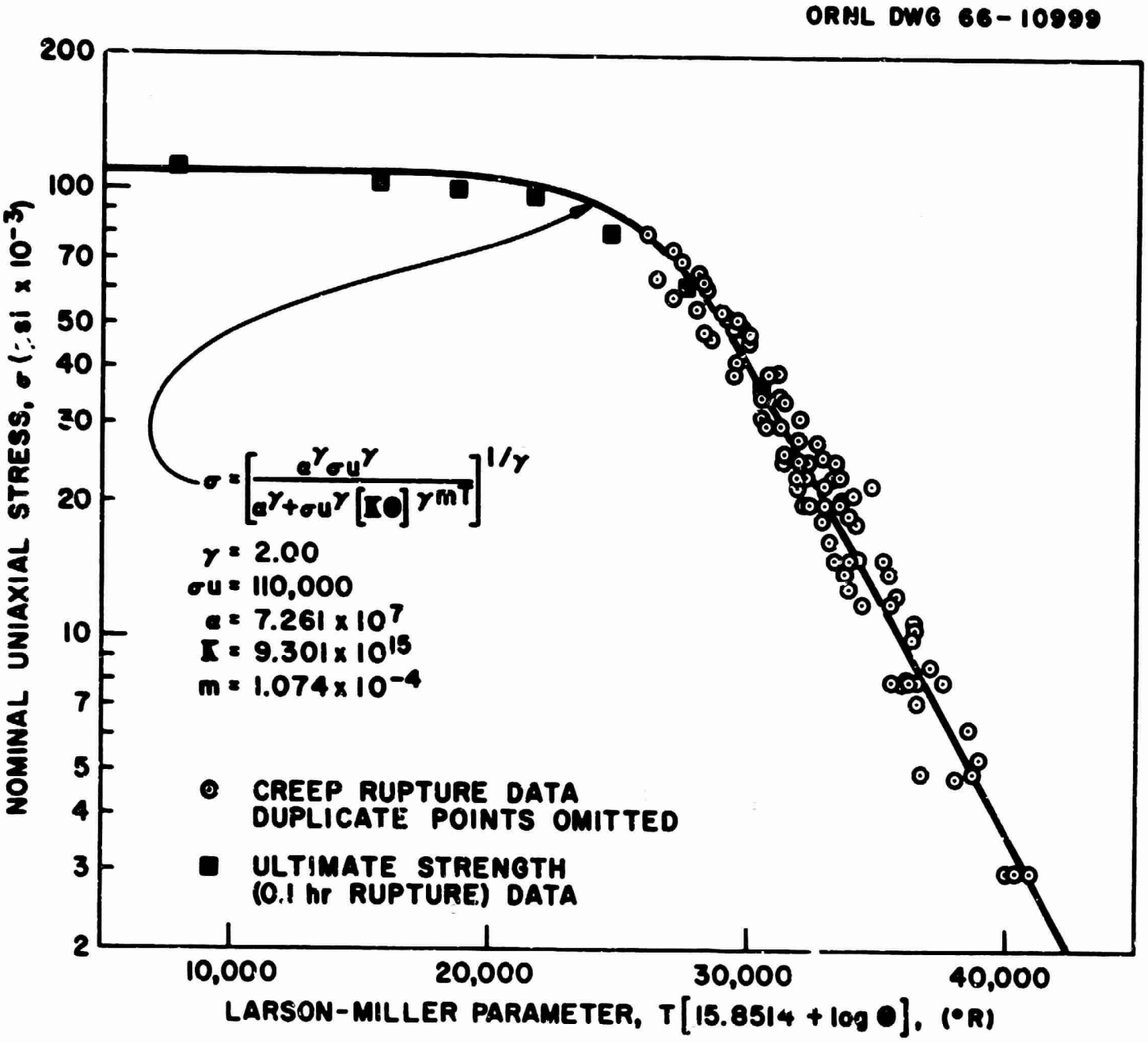



Fig. 6. Master Larson-Miller Plot for Creep Rupture in Hastelloy N. 5 heats. 93 data points. $1560 \leq T \leq 2260$. $\Theta \leq 14{,}400$ hr.

ORNL DWG. 66-11004

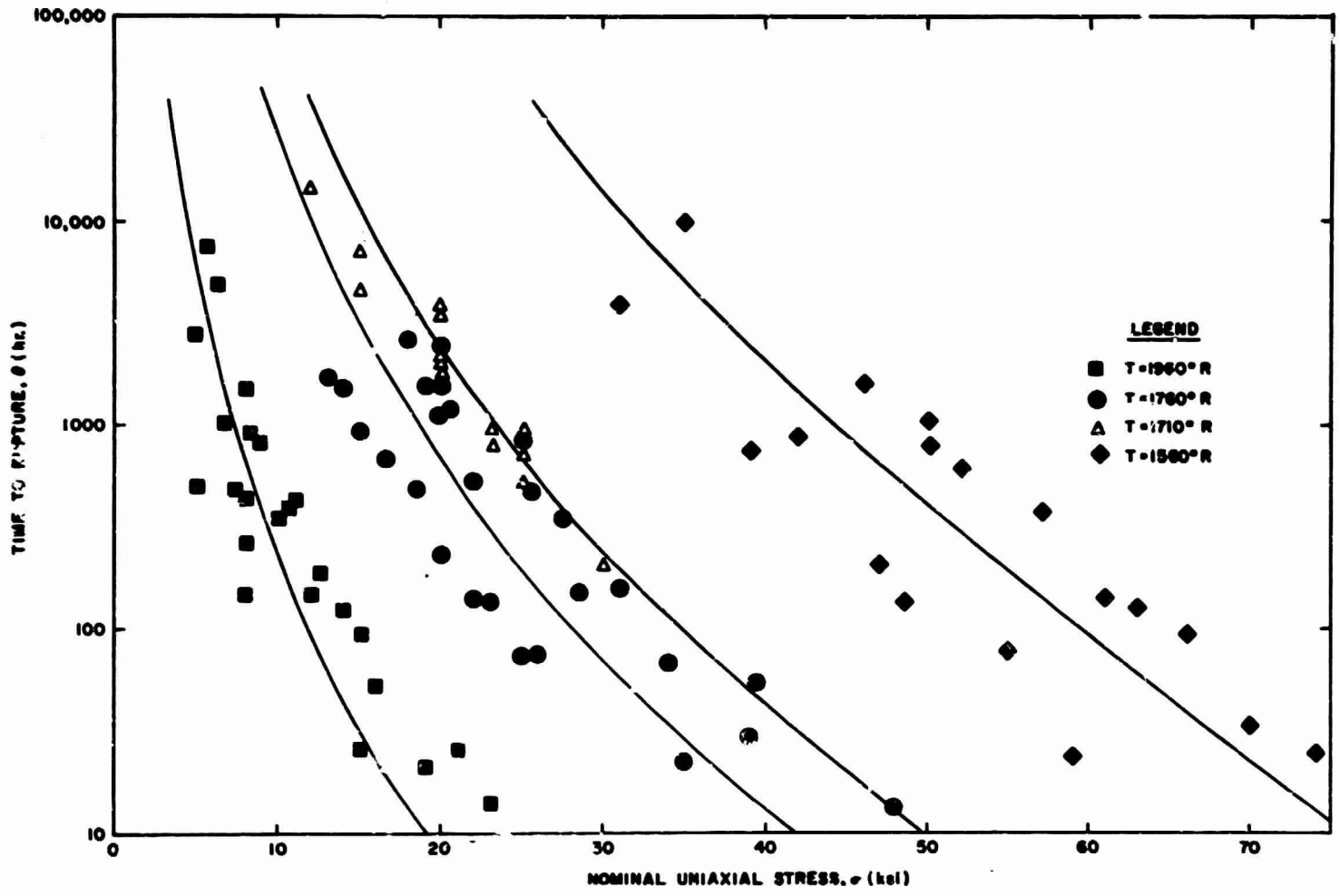



Fig. 7. Parametric Plot of the Derived Creep Rupture Function, $\theta(\sigma, T)$, for INOR-8 Compared with Experimental Data. The function was determined by a least-squares fit of an assumed functional type using 93 data points measured with 5 heats at 7 temperatures over a rupture life range of 0.5 to 14,000 hr.

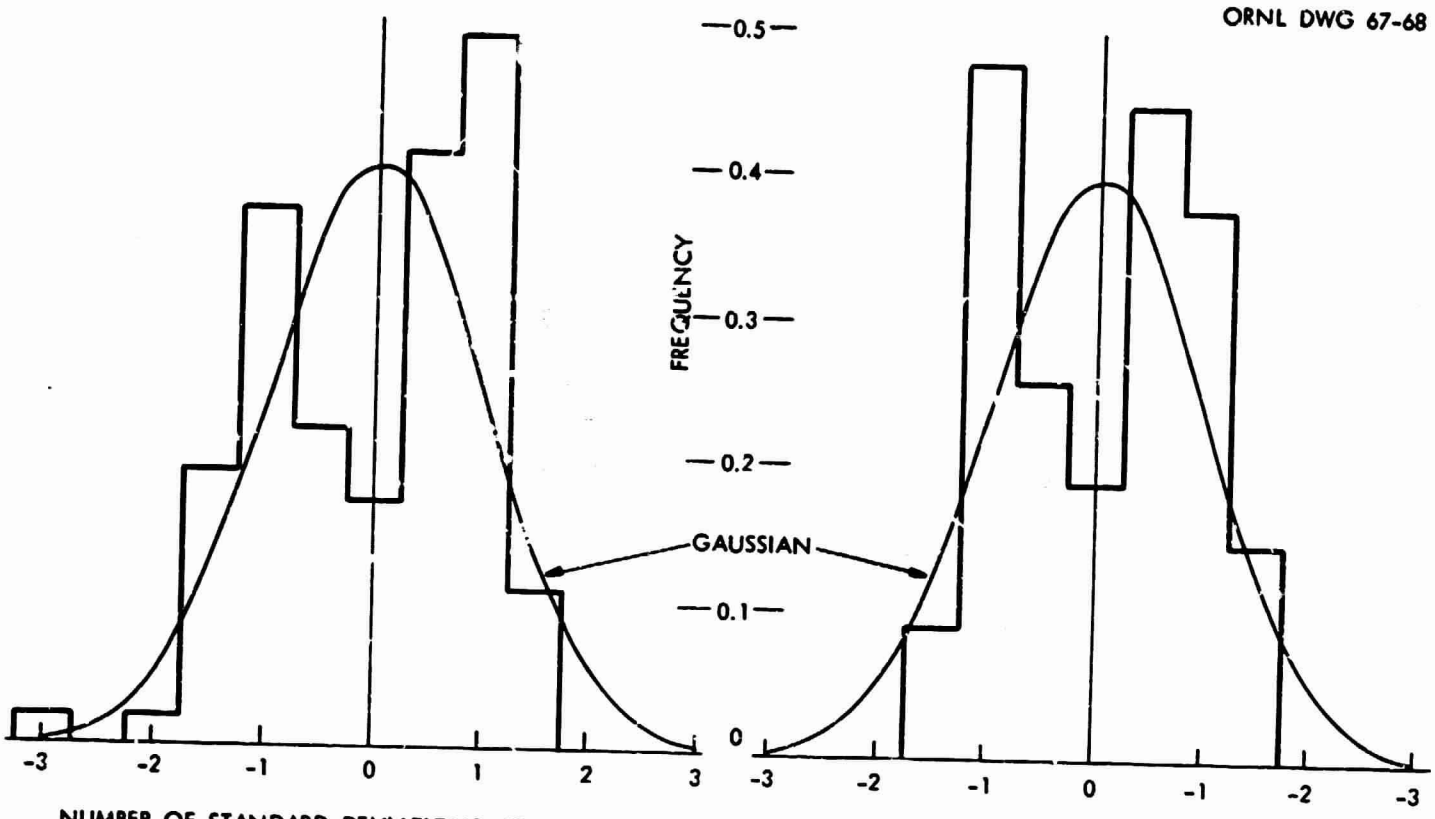



Fig. 8. Distribution of the Error in Measured Rupture Life and Stress for Hastelloy N with Respect to Values Predicted by a Least-Squares Fit of the Creep Rupture Function. Area normalized to 1.00.

The creep life function for rupture of Cb-1% Zr (Fig. 9) was generated using data[14,15] from 25 heats of sheet, bar, and plate, covering the temperature domain from 1600 to 2200°F and rupture life to 1733 hours. The abnormally large amount of scatter in the data, shown in the parametric plot (Fig. 10), is caused, primarily, by changes that were made in the composition and heat treatment of the alloy that resulted, accidentally or deliberately, between the periods of testing. The frequency distribution of the error (Fig. 11) is approximately gaussian. The relative standard error in stress is 0.14.

The analysis of these creep data indicates that the general creep life function chosen for the model can provide a good description of constant stress, constant temperature creep behavior in 304 stainless steel, Hastelloy N, and Cb-1% Zr. Based on the results of others[4,5] who have correlated creep data with the Larson-Miller parameter (but, generally not extending the fit into the ultimate strength range), we assume that the same conclusion will apply to many other metals and alloys (probably including most alloys of copper, nickel, iron, aluminum, and the refractory metals). It is apparent from the analysis that the ultimate strength data are useful for complementing the creep data in the high-stress, low-temperature region.

The parametric plots of the fitted functions and data show that the distribution of data with respect to the fitted functions tends to be random in that the relative error does not vary significantly over the domain of the variables. In general, we have found that the frequency distribution function is most nearly gaussian when either one heat or many heats of an alloy are analyzed. There may be significant deviation from gaussian behavior if only a few heats of material, of varying properties, are analyzed.

## 4.2 Analysis of Errors in Time Extrapolations

The accuracy of the model for extrapolation of creep life in the three commercial alloys was investigated by making fits of constant stress and temperature creep data with rupture life less than a selected time; using these fitted functions to predict the stress as a function of

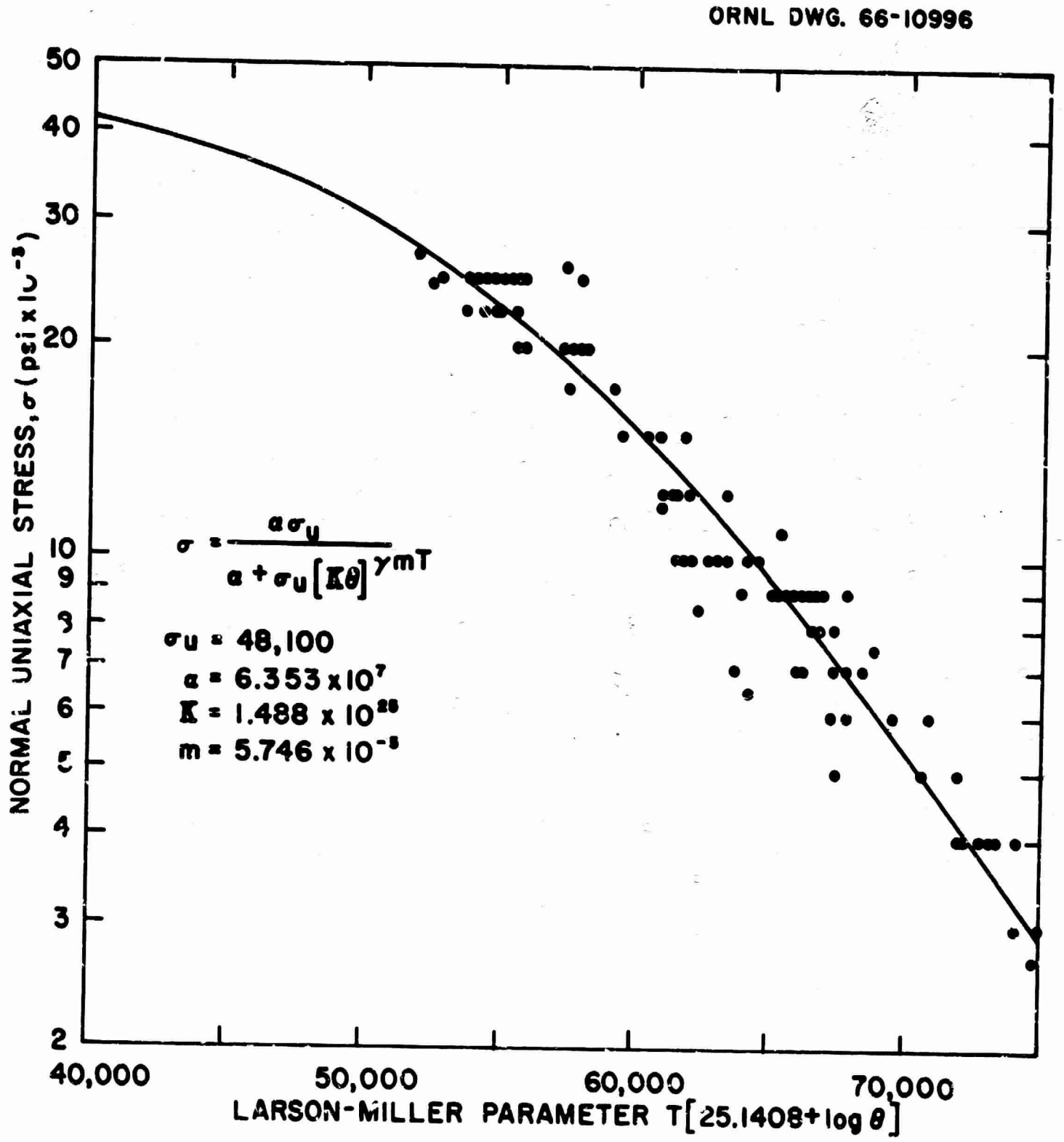



Fig. 9. Master Larson-Miller Plot for Creep Rupture in Cb--1% Zr. 25 heats. $2060 \leq T \leq 2660$. $\theta \leq 1733$ hr.

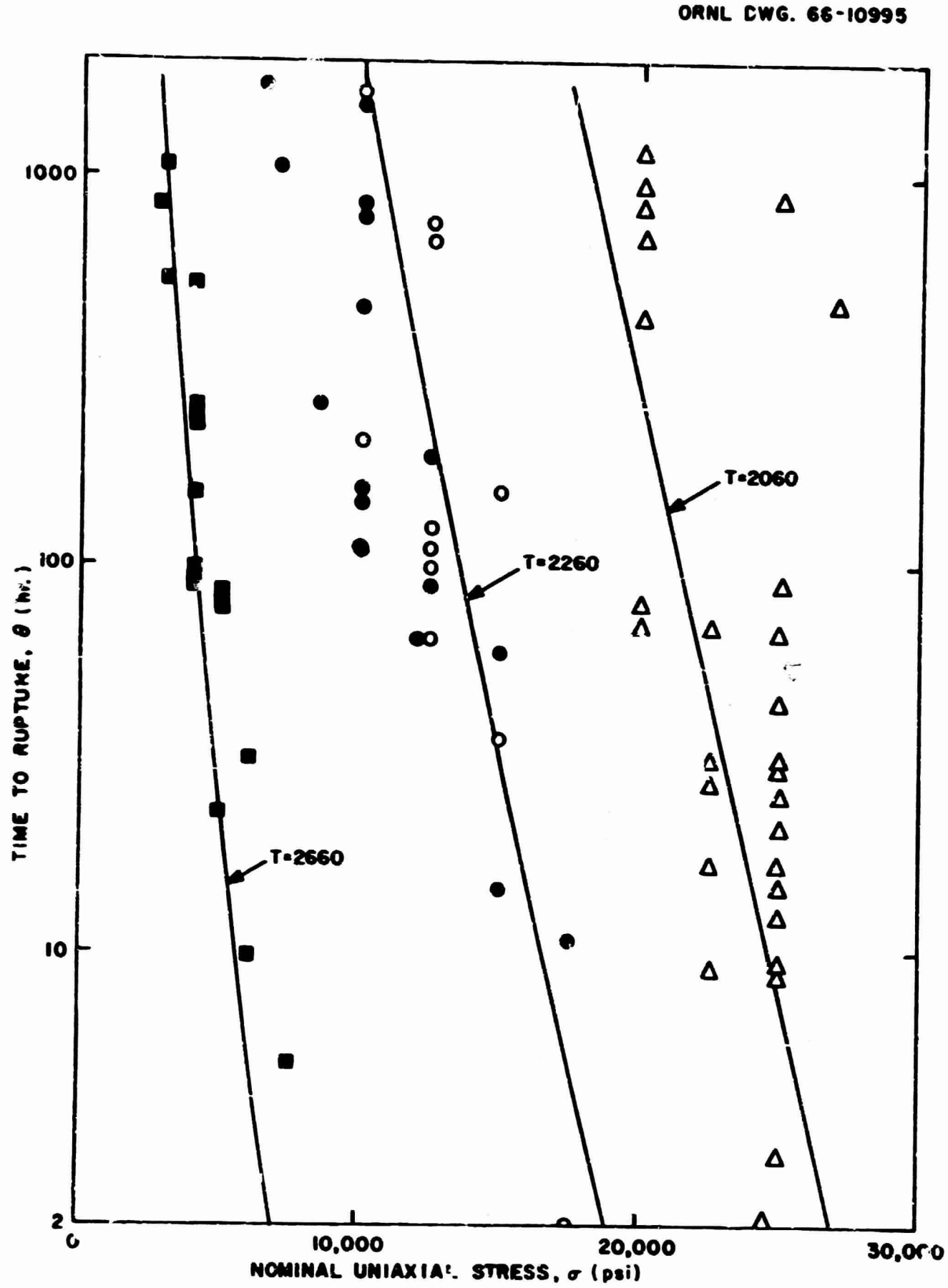



Fig. 10. Parametric Plot of the Derived Creep Rupture Function of Cb--1% Zr Alloy Compared with Experimental Data. 25 heats. 104 data points. $2060 \leq T \leq 2660$. $\theta \leq 1733$. ORNL data points closed; Pratt & Whitney data points open.

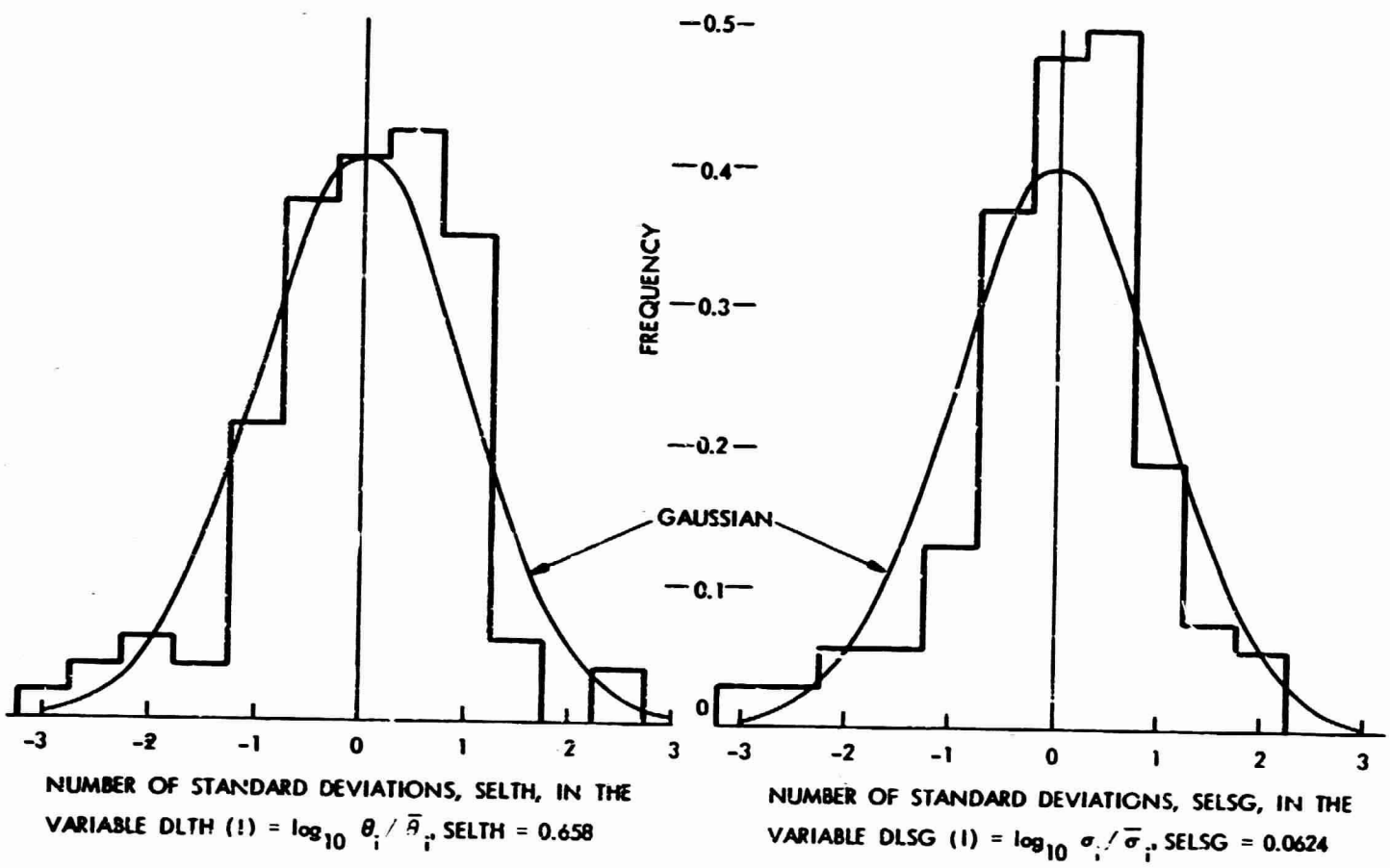



Fig. 11. Distribution of the Error in the Logarithm of Measured Rupture Life and Stress for Cb--1% Zr with Respect to Values Predicted by a Least-Squares Fit of the Creep Rupture Function. Area normalized to 1.00.

temperature to cause rupture at a greater time; and comparing the predicted stress with the actual average stress which caused failure at the greater time. The accuracy of the extrapolation, expressed in terms of the maximum relative error (or bias) in the predicted stress as a function of temperature, is then compared to the standard error (error at 67 percent confidence level since the distribution of stress data is approximately gaussian) in relative stress caused by the normal scatter of data about the predicted best-fit function. The model is then known to provide for adequate extrapolation of creep within the time range such that the bias in the predicted stress is small as compared to the standard error.

The first column of Table 2 shows the material, number of heats, and temperature range of the creep data that were analyzed. The next two columns show the values of the selected times and the corresponding number of data points (with rupture life less than the selected time) for which best-fit functions were evaluated. The fourth column shows the values of rupture life for which stress as a function of temperature was predicted by the model. The fifth column shows the standard error in relative stress for each of best-fit functions. The last column shows the maximum bias in the predicted stress, defined as the ratio of the maximum difference between the predicted and mean-measured stress to the mean-measured stress within the temperature range of the experiments.

Two very striking, but tenuous, conclusions may be drawn based on the results of Table 2.

1. It appears that, for a large number of metals, the present creep model can provide a fit of creep data such that the standard error in relative stress will not exceed 15 to 17 percent. Knowledge of the statistics of the fit and the frequency distribution function will permit the choice of a design stress for a predetermined confidence level.

2. In each of the commercial alloys analyzed, the bias in predicted stress for a long-time extrapolation becomes small, using only 100 to 200-hr creep data. For these alloys, the creep data for times greater than 100 to 200 hours is superfluous. The data seem to suggest that if

Table 2. Statistics and Predictability of Creep Rupture Data for Several Alloys

| Material<br>No. of Heats<br>Temp. Domain, °F | No. of Data Points | Maximum Lifetime in Data (hr) | Life to be Predicted (hr) | Relative Standard Error in Stress | Maximum Absolute Relative Bias in Predicted Stress |
|---|---|---|---|---|---|
| 304 Stainless Steel[a]<br>18 heats<br>900 ≤ T ≤ 1700 | 15 | 10 | 100,000 | 0.03 | -1.0 |
| | 63 | 100 | 100,000 | 0.13 | -0.10 |
| | 123 | 1,000 | 100,000 | 0.14 | -0.09 |
| | 173 | 10,000 | 100,000 | 0.16 | -0.05 |
| | 189 | 100,000 | 100,000 | 0.17 | 0.0 |
| Hastelloy N<br>5 heats<br>1100 ≤ T ≤ 1800 | 26 | 100 | 10,000 | 0.16 | 0.2 |
| | 40 | 200 | 10,000 | 0.16 | 0.08 |
| | 55 | 500 | 10,000 | 0.17 | -0.04 |
| | 69 | 1,000 | 10,000 | 0.16 | -0.02 |
| | 81 | 2,000 | 10,000 | 0.16 | 0.008 |
| | 93 | 15,000 | 10,000 | 0.16 | 0.0 |
| Cb - 1% Zr<br>25 heats<br>1600 ≤ T ≤ 2200 | 49 | 80 | 1,500 | 0.09 | 0.25 |
| | 66 | 150 | 1,500 | 0.10 | 0.07 |
| | 83 | 300 | 1,500 | 0.12 | -0.04 |
| | 92 | 750 | 1,500 | 0.13 | 0.05 |
| | 104 | 1,800 | 1,500 | 0.14 | 0.0 |

[a]Note use of previously reduced data described on p. 23.

creep data are available that span two to three decades in time (1 to 200 hours, for example), then a fitted function can be determined that will permit extremely long extrapolations in time with the error of the extrapolation being less than the normal scatter in moderate-time creep rupture data. This conclusion assumes, of course, that the long-term environment is the same as the test environment.

The data also suggest a possible method for avoiding experiments of unnecessarily long term in creep determinations. The creep program would begin by generating first short time; then, progressively, longer time data. The composite data would then be fitted, sequentially, as each new, longer-time data point is generated; and the fitted function would be used to predict the conditions to cause failure (or a given strain) at a very long time (perhaps 100,000 hours). The creep program would be terminated when sufficient data are generated such that sequential data cause small and random (as opposed to monotonic) changes in the predicted conditions to cause long-term failure.

### 4.3 Analysis of Constant Stress Rate Tests

The accuracy of the model in predicting creep, under conditions of varying stress, was investigated by analyzing constant stress rate tests with the CAPSUL program. The time to rupture of the alloy T-222 and time to 1 percent strain of the alloy FS-85 exposed to a constantly increasing stress rate and constant temperature were calculated (Fig. 12) using a creep life function that was generated using conventional creep and ultimate strength data[9,16] over a wide range of stress and temperature. The predicted stress rates agree with the measured data for the alloy T-222 within experimental error. Agreement with the FS-85 data is satisfactory, but it appears that the predicted results are biased at the particular temperature at which the experiments were conducted.

McCoy[8] has analyzed these same creep data using a simpler, but less general, model. McCoy fitted conventional creep data, at the specific temperature of interest, by assuming that the logarithm of stress is linearly related to the logarithm of the creep life, an assumption that

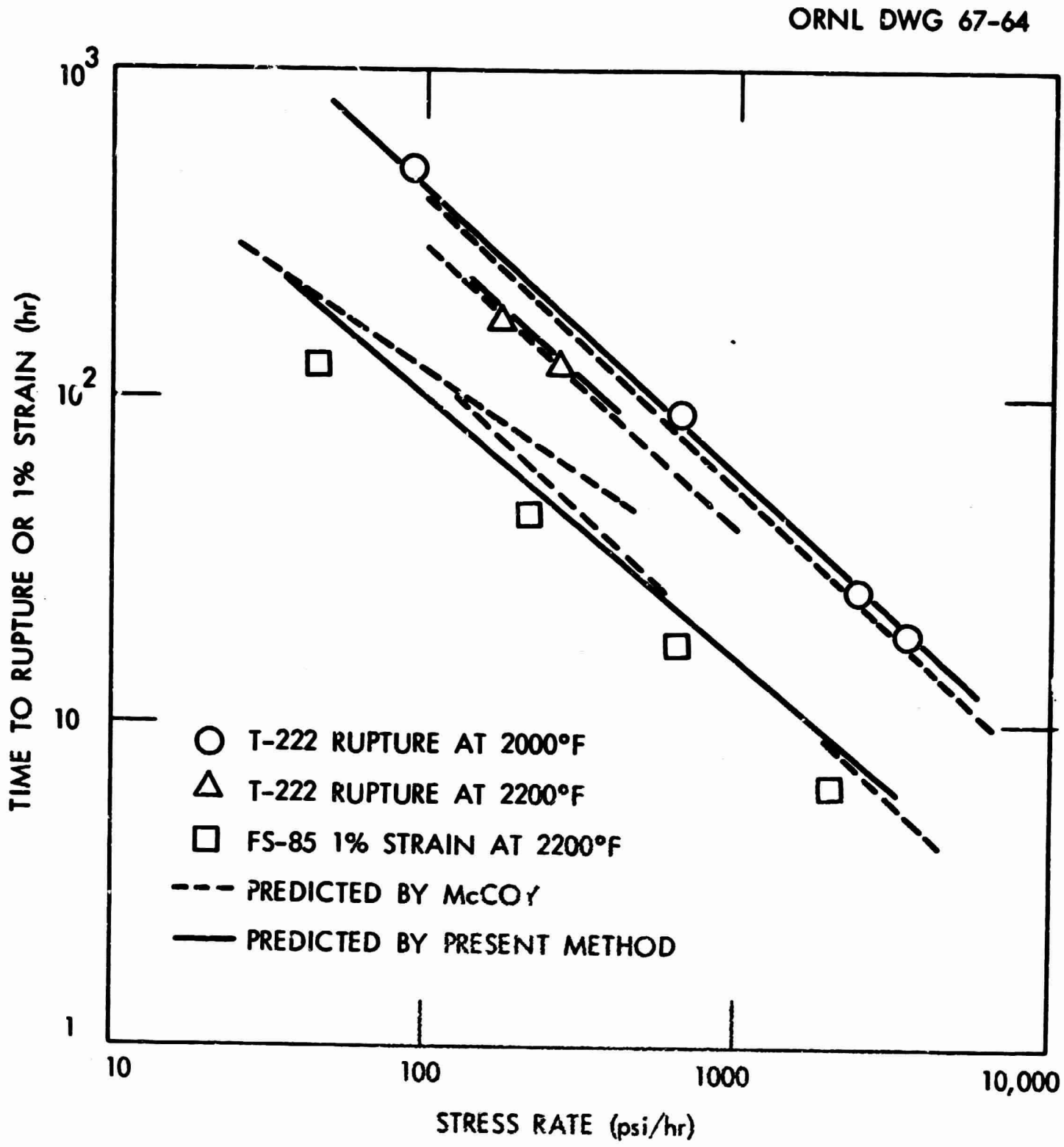



Fig. 12. Predicted and Observed Creep Behavior of T-222 and FS-85 at a Constant Stress Rate and Constant Temperature.

is equivalent to the assumption used in the present method if the temperature is high. McCoy's predictions are also good, but no better than those predicted by the present method in spite of the fact that his model was based on data only for the specific temperature of interest. McCoy required two equations to provide a good fit of the FS-85 data over the stress rate domain of the experiments.

The procedure for determining the 1-percent-creep function for the FS-85 is shown as an example of the MOPT = 1 sequence of the CAPSUL program in Appendix B. Also shown is one of the constant stress rate calculations for rupture of the T-222 alloy as an example of the MOPT = 4 sequence.

## REFERENCES


1. E. L. Robinson, Trans. Am. Soc. Mech. Engrs. 74(5), 778-81 (1952).

2. S. V. Serenson and V. S. Strelayev, "Creep Resistance and Low-Cycle Fatigue of Fibreglass Plastics," pp. 3-93 to 3-98 in Proceedings of Joint International Conference on Creep, Institute of Mechanical Engineers, 1963.

3. A. J. Kennedy, Processes of Creep and Fatigue in Metals, Wiley, New York, 1963, pp. 402-28.

4. E. C. Larke and N. P. Inglis, "A Critical Examination of Some Methods of Analyzing and Extrapolating Stress-Rupture Data," pp. 6-33 to 6-47 in Proceedings of Joint International Conference on Creep, Institute of Mechanical Engineers, 1963.

5. F. R. Larson and J. Miller, Trans. Am. Soc. Mech. Engrs. 74(5), 765-75 (1952).

6. C. R. Kennedy, Metals and Ceramics Div. Ann. Progr. Rept. May 31, 1962, ORNL-3313, pp. 44-45.

7. H. E. McCoy, Metals and Ceramics Div. Ann. Progr. Rept. June 30, 1965, ORNL-3870, pp. 122-23.

8. H. E. McCoy, "Creep-Rupture Properties Under Conditions of Constant and Varying Stresses," Nucl. Appl. 2(6), 481-84 (1966).

9. H. E. McCoy, Creep Rupture Properties of the Tantalum-Base Alloy T-222, ORNL-TM-1576 (Sept. 1966).

10. F. H. S. Clark and F. B. K. Kam, A Generalized One-Constraint Lagrange Multiplier Numerical Formulation, ORNL-3742 Rev. (June, 1965).

11. W. F. Simmons and J. A. Van Echo, The Elevated Temperature Properties of Stainless Steels, American Society for Testing and Materials, Philadelphia, Pa., 1965.

12. R. W. Swindeman, The Mechanical Properties of INOR-8, ORNL-2780 (Jan., 1961).

REFERENCES (contd)

13. J. T. Venard, Tensile and Creep Properties of INOR-8 for the Molten Salt Reactor Experiment, ORNL-TM-1017 (February, 1965).

14. H. E. McCoy, Creep Properties of the Nb-1% Zr Alloy, ORNL-TM-905 (August, 1964).

15. Creep Rupture Data for Cb-1 Zr Alloy, CNIM-4267, Pratt and Whitney (Sept. 1962).

16. R. L. Stephenson, Creep-Rupture Properties of FS-85 Alloy and Their Response to Heat Treatment, ORNL-TM-1456 (July 1966).

# APPENDIX A

## Units and Glossary of Symbols

### Units

| | | |
|---|---|---|
| Length | - | inch |
| Mass | - | pound |
| Time | - | hour |
| Temperature | - | degrees Rankine |
| Heat | - | Btu |

### Symbols

| | |
|---|---|
| A, A0, A2 | Thermal power per unit of fuel region. |
| AIAM, AIF(IX + 1) | The Lagrange multiplier. |
| AIF(K), K = 1,19 | Independent variables in MAX. |
| ALPHA, $\alpha$ | The fitted value of creep stress for a Larson-Miller parameter of zero. |
| AV2 | Thermal power of the fuel region. |
| BETA | One-half the surface-to-surface distance between parallel columns of capsules. |
| C | Constrained value of DH. |
| CRIT | An input value of a convergence criterion for MAX. |
| D, DH | Value of the constraining function, Eq. (2). |
| DEL(K), K = 1,19 | The increment in the variables AIF(K) in calculating DH. |
| DELOME | The increment in OMEGA. |
| DELTA | Thickness of the structural liner surrounding the capsule. |
| DELRO | The increment in RO. |

| | |
|---|---|
| DELTH | The increment of THET. |
| DELX2 | The increment in X(2). |
| DELX4 | The increment in X(4). |
| DLSG(I), I = 1,500 | The common logarithm of SIGMA(I)/SIGB(I). |
| DLTH(I), I = 1,500 | The common logarithm of THETA(I)/THETB(I). |
| E | Capsule weld efficiency. |
| ETA | Volume fraction of gas space in the fuel region. |
| F | Integrand of the constraining function. |
| G | A number slightly less than 1.0 in the convergence criterion for MAX. |
| GAMMA, $\gamma$ | A constant in F, usually 1.0 or 2.0. |
| H | Weight fraction of alpha-emitting isotopes in the fuel. |
| I | An index. |
| ITER | Maximum number of outer iterations in MAX. |
| IX | Number of independent variables in MAX. |
| J | An index. |
| JAM | An integer that counts and limits to 100 the number of iterations to adjust input estimates of X(2) and X(4) such that $0.3 \leq DH \leq 4.0$. |
| K | Maximum number of data points in LSTSQ. Elsewhere an index. |
| L | An index in THETAC. Overall length of the fuel capsule in the main program. |
| L2 | Length of fueled section of capsule. |
| L4 | Length of the straight section of the capsule. |
| LAMDA, $\lambda$ | Decay constant of the radioisotope. |

| | |
|---|---|
| LMOST | Maximum number of inner iterations in MAX. |
| LMP(I), I = 1,500 | Value of the Larson-Miller parameter. |
| LSTOP | An index which counts inner iterations in MAX. |
| M | An index which counts outer iterations in MAX. An index in RZERO. |
| MOPT | An input index which decides the sequence to use in the main program |
| N, NN | The initial number of (even) increments for Simpson's rule integration in DR. If MOPT = 2 or 3, N is the initial number of increments in the time domain 0.0 to THET. If MOPT = 4, N is the initial number of increments in the domain 0.0 to TAU and NN is the initial number of increments in the domain TAU to THET. |
| NITER | An input index to determine if a convergence criterion is to be used in MAX. |
| NLAM | An input index to determine if GTALAM is to be used to estimate the initial value of the Lagrange multiplier. |
| NMAX | An input index that specifies the set of independent capsule dimension variables that are subject to optimization. |
| NNN | An index that causes additional creep data cards to be read and analyzed by LSTSQ together with data previously in storage. |
| NQ | An input index that decides the function to be maximized. |
| NR | The index of the Lagrange multiplier, IX + 1. |
| OMEGA, XK(1) | A number that multiplies the safety factors SC and SU in THETAC to satisfy DK = 1.0 if THET > THMAX. |
| P(I), I = 1,9 | Density of the material in region I of the capsule. |
| PHI | An input convergence criterion for the Simpson's Rule integration in DR. |

| | |
|---|---|
| PS | Initial pressure in the fuel capsule. |
| Q | An input convergence criterion on DH for use in the subroutines DF and THETAC. |
| R(I), I = 1,8 | Outer radius of region I of the capsule. |
| RESIG | The standard error in relative stress for the fitted creep life function. |
| RK | YSQ·G, a test quantity in MAX. |
| RO | The outer radius of the void region of the capsule. |
| R3L | The imposed lower limit for the outer radius of region 3 of the capsule. |
| R3U | The imposed upper limit for the outer radius of region 3 of the capsule. |
| SC, $S_c$ | The safety factor on creep stress. |
| SELTH | The standard error in the common logarithm of creep life for the fitted creep life function. |
| SELSG | The standard error in the common logarithm of stress for the fitted creep life function. |
| SIGB(I), I = 1,500, $\sigma$ | The value of stress predicted by the fitted creep life function for the values T(I) and THETA(I). |
| SIGMA(I), I = 1,500 | The measured value of stress in the input creep data. |
| SIGT | The current value of stress as a function of time in DR. |
| SIGU, $\sigma_u$ | An imposed upper limit on stress as a function of the Larson-Miller parameter. |
| SU, $S_u$ | The safety factor on SIGU. |
| T, t | Time since application of the stress. |
| T(I), I = 1,500 | The measured value of absolute temperature in the creep data. |
| TA | Ambient temperature of the earth (MOPT = 2 or 3) and the first environment (MOPT = 4). |

| | |
|---|---|
| TA1 | Ambient temperature of the second environment (MODT = 4), see T81. |
| TAU | Lifetime in the initial service environment for use in THETAC. |
| TERM | The upper limit of the integral in DR either TAU or THET. |
| THET, THETA, $\theta$ | Resultant life of the fuel capsule exposed to varying stress and temperature. |
| THETA(I), I = 1,500 | The measured value of life to a prescribed creep criterion in the input creep data. |
| THETB(I), I = 1,500, $\theta$ | The value of life predicted by the fitted creep life function for the values SIGMA(I) and T(I) |
| THMAX | An imposed upper limit on the value of THET in THETAC. |
| TNIT | The lower limit of the integral in DR, either zero or TAU. |
| TS | Temperature of the gas wher the capsule is sealed. |
| T0 | The initial steady state temperature of the helium gas in the fuel capsule as calculated in DR. |
| T4 | The instantaneous value of the steady state temperature of the inner wall of the primary structural material. |
| T40 | The initial steady state temperature of the inner wall of the primary structural material. |
| T8 | The imposed initial steady state of the outer surface of the capsule. In THETAC this is the initial temperature in the time period zero to TAU |
| T81 | In subroutine THETAC the imposed initial steady state temperature of the surface of the fuel capsule in the time period TAU to THETA expressed as the temperature at zero time (i.e., at full thermal power). |

| | |
|---|---|
| T82 | The initial steady state temperature of the surface of the capsule assuming that it becomes buried in an infinite medium of conductivity XK(10) at zero time. |
| V(I), I = 1,9 | Volume of region I of the capsule. |
| VO | Volume of the void region. |
| V2 | Volume of the fuel region. |
| W, WH | The specific thermal power function that is to be maximized (W1, W2, or W3). |
| WEL(K), K = 1,19 | The increment in the variables ALF(K) in calculating WH. |
| WT | Weight of the fuel capsule. |
| W1 | Thermal power of the capsule per unit of projected area. |
| W2 | Thermal power of the capsule per unit of volume of a circumscribed rectangular parallepiped. |
| W3 | Thermal power of the capsule per unit of weight. |
| X(I), I = 1,9 | Thickness of region I of the capsule. |
| X(K,1), XX(K,1), K = 1,19 | An increment in the variable ALF(K). |
| XLD | The overall length-to-diameter ratio of the capsule. |
| XM, m | A fitted constant in the creep life function. |
| YKO, YK, H | A fitted constant in the creep life function. |
| XK(I), I = 2,8 | Thermal conductivity of region I of the capsule. |
| XK(9) | Thickness of the capsule bushing (see Fig. 2). |
| XK(10) | Temperature averaged thermal conductivity of the infinite medium in which the capsule is immersed. |
| X4L | The imposed lower limit on the thickness of the primary structural material of the capsule. |

| | |
|---|---|
| X4U | The imposed upper limit on the thickness of the primary structural material of the capsule. |
| YK | See XKO. |
| YSQ | The square of the vector $\bar{y}$, which is the function minimized in MAX. |
| ZM | Atomic weight of the alpha-emitting radioisotope. |

## Capsule Regions (Fig. 2)

| | |
|---|---|
| 0 | Helium |
| 1 | Inner fuel liner |
| 2 | Radioisotope fuel |
| 3 | Outer fuel liner |
| 4 | Primary containment wall |
| 5 | Diffusion barrier |
| 6 | Gas gap |
| 7 | Corrosion barrier |
| 8 | Radiant coating |
| 9 | Fuel spacer |
| 10 | An infinite heat conducting medium |

# APPENDIX B

## Example Problems

### B1. MOPT = 1

This sequence is used to generate constants in the creep life function for 1 percent strain in the alloy FS-85. A photograph of the output is shown in Table B1. The creep data are the 28 data points (K = 28) reported by Stephenson.[16] The ultimate strength at room temperature (SIGU) is taken to be 80,000 psi. A moderately large value of 2.0 is chosen for the constant GAMMA to provide a rather abrupt change in the behavior of the function in the transition range. Shown on the second and third rows of the output are the computed constants (ALPHA, XM, XKC) and the standard errors in life and stress (SELTH, SELSG, and RESIG) relative to the fitted function. The array of basic experimental data [SIGMA(I), T(I), THETA(I)] is shown in columns two through four of the output. Other columns list pertinent calculated results [LMP(I), THETB(I), SIGB(I). DLTH(I), and DLSG(I)] for each data point.

Table B1

K = 28 SIGU = 80000. GAMMA = 2.00

ALPHA = 0.336050E 08 XM = 0.916876E-04 XKO = 0.133654E 14

SELTH = 0.418337E 00 SELSG = 0.974960E-01 RESIG = 0.202662E 00

| I | SIGMA(I) | T(I) | THETA(I) | LMP(I) | THETB(I) | SIGB(I) | DLTH(I) | DLSG(I) |
|---|---|---|---|---|---|---|---|---|
| 1 | 35000. | 2260. | 4.10 | 0.31050E 05 | 0.11064E 02 | 0.41041E 05 | -0.43112E 00 | -0.69151E-01 |
| 2 | 30000. | 2260. | 24.00 | 0.32784E 05 | 0.26966E 02 | 0.30627E 05 | -0.50602E-01 | -0.89825E-02 |
| 3 | 25000. | 2260. | 128.00 | 0.34427E 05 | 0.73123E 02 | 0.22490E 05 | 0.24316E 00 | 0.45943E-01 |
| 4 | 22500. | 2260. | 205.00 | 0.34889E 05 | 0.12771E 03 | 0.20544E 05 | 0.20554E 00 | 0.39505E-01 |
| 5 | 21000. | 2260. | 470.00 | 0.35704E 05 | 0.18296E 03 | 0.17467E 05 | 0.40975E 00 | 0.80005E-01 |
| 6 | 25000. | 2460. | 0.74 | 0.31968E 05 | 0.44193E 01 | 0.35334E 05 | -0.77612E 00 | -0.15025E 00 |
| 7 | 20000. | 2460. | 6.30 | 0.34256E 05 | 0.12936E 02 | 0.23247E 05 | -0.31247E 00 | -0.65341E-01 |
| 8 | 17500. | 2460. | 32.00 | 0.35993E 05 | 0.24202E 02 | 0.16478E 05 | 0.12131E 00 | 0.26122E-01 |
| 9 | 15000. | 2460. | 110.00 | 0.37312E 05 | 0.49364E 02 | 0.12587E 05 | 0.34799E 00 | 0.76154E-01 |
| 10 | 14000. | 2460. | 130.00 | 0.37490E 05 | 0.67727E 02 | 0.12133E 05 | 0.28318E 00 | 0.62165E-01 |
| 11 | 13000. | 2460. | 130.00 | 0.37490E 05 | 0.94983E 02 | 0.12133E 05 | 0.13630E 00 | 0.29980E-01 |
| 12 | 10000. | 2460. | 600.00 | 0.39124E 05 | 0.31147E 03 | 0.86428E 04 | 0.28473E 00 | 0.63345E-01 |
| 13 | 17500. | 2660. | 0.75 | 0.34583E 05 | 0.19628E 01 | 0.21818E 05 | -0.41782E 00 | -0.95772E-01 |
| 14 | 16000. | 2660. | 2.50 | 0.35974E 05 | 0.28826E 01 | 0.16542E 05 | -0.61841E-01 | -0.14460E-01 |
| 15 | 15000. | 2660. | 4.00 | 0.36517E 05 | 0.37946E 01 | 0.14815E 05 | 0.22890E-01 | 0.53835E-02 |
| 16 | 12500. | 2660. | 15.00 | 0.38044E 05 | 0.81978E 01 | 0.10821E 05 | 0.26239E 00 | 0.62632E-01 |
| 17 | 10000. | 2660. | 55.00 | 0.39544E 05 | 0.20846E 02 | 0.79165E 04 | 0.42133E 00 | 0.10147E 00 |
| 18 | 9000. | 2660. | 80.00 | 0.39977E 05 | 0.32309E 02 | 0.72310E 04 | 0.39376E 00 | 0.95044E-01 |
| 19 | 7000. | 2660. | 121.00 | 0.40455E 05 | 0.91485E 02 | 0.65418E 04 | 0.12144E 00 | 0.29402E-01 |
| 20 | 7000. | 2660. | 200.00 | 0.41036E 05 | 0.91485E 02 | 0.57914E 04 | 0.33968E 00 | 0.82313E-01 |
| 21 | 5000. | 2660. | 152.00 | 0.40719E 05 | 0.36632E 03 | 0.61900E 04 | -0.38201E 00 | -0.92720E-01 |
| 22 | 5000. | 2660. | 205.00 | 0.41064E 05 | 0.36632E 03 | 0.57568E 04 | -0.25210E 00 | -0.61213E-01 |
| 23 | 2000. | 2660. | 1000.00 | 0.42895E 05 | 0.15792E 05 | 0.39168E 04 | -0.11984E 01 | -0.29190E 00 |
| 24 | 10000. | 3060. | 0.30 | 0.38565E 05 | 0.26964E 00 | 0.97097E 04 | 0.46338E-01 | 0.12795E-01 |
| 25 | 8000. | 3060. | 1.10 | 0.40292E 05 | 0.60340E 00 | 0.67694E 04 | 0.26079E 00 | 0.72537E-01 |
| 26 | 6000. | 3060. | 3.60 | 0.41868E 05 | 0.16956E 01 | 0.48623E 04 | 0.32697E 00 | 0.91306E-01 |
| 27 | 3500. | 3060. | 20.00 | 0.44147E 05 | 0.11656E 02 | 0.30088E 04 | 0.23449E 00 | 0.65669E-01 |
| 28 | 1000. | 3060. | 280.00 | 0.47654E 05 | 0.10165E 04 | 0.14357E 04 | -0.55995E 00 | -0.15707E 00 |

B2. MOPT = 3

This problem requires determination of the dimensions of a capsule for $^{244}Cm_2O_3$ fuel having maximum power per unit of volume. Restrictions on the capsule are: (1) The primary container material, fuel liners, and the bushing are to be made of Hastelloy N. (2) The fuel liners and bushing are to be 0.010 in. thick. (3) The outer surface temperature of the capsule is not to exceed 2100°R if the capsule becomes buried in dry sand [XK(10) = 0.0167] immediately after incapsulation. (4) The capsule is to be designed such that the probability of rupture in 5 years is 0.001. This condition (see Sect. 4) is approximately satisfied by designing for rupture in 5 years with a stress safety factor of 2.0. (5) $0.05 \leq X(4) \leq 0.5$, $0.01 \leq R(3) \leq 3.0$.

The input data for this problem are tabulated in the first 19 rows of the printed output (Table B2). Initial estimates of the variables R), X(2), and X(4) are 0.3, 0.1, and 0.16, respectively. In this problem, it is chosen to use only outer iterations in the MAX calculations (LMOST = 0). The variables and their increments for use in this calculation are so small that inner iterations would sometimes fail because of loss of significance (the IBM 360/75 carries only seven significant figures in this program).

The minimum value of YSQ (57.36) for which the constraint (D = 1.0000) is satisfied and P is maximum (59.25 Btu/hr-in$^3$) occurs in iteration number 48 (M = 48). The iterations that follow produce the same computed values indicating that YSQ is trapped at either a stationary point or an absolute minimum that may be obtained with the current set of variables and their increments. Calculations with higher and lower starting estimates of X(4) indicated that this solution provides maximum power within the range of interest in the variables R(3) and X(4). The last array of computed values in the output is the same as in the last iteration produced by MAX, indicating that LIMIT did not call DF because R(3) and X(4) were within the preselected limits.

Table B2

```
T8 = 2100.0      A =    1454.0      ETA =  0.4      BETA =  0.0
DELTA =  0.0       C =  1.00      G =  1.00      GAMMA = 2.00
RR =   18520.0     E =  1.000     PS =  0.0      TS =.560.0
M =        0.865     ZM =  244.0      TA =  560.0      TAI =     0.0
LAMBDA =    0.43680E-05     N = 100      NN = 100

NOPT =  3     NMAX =  5     NQ =  2

THE FOLLOWING THREE ROWS ARE THE CONTENTS OF THE X, P, AND XK ARRAYS

     0.0100      0.0         0.0100      0.0         0.0         0.0         0.0         0.0         0.0
     0.3250      0.3860      0.3250      0.3250      0.0         0.0         0.0         0.0         0.0
     0.1000      2.0000      2.0000      1.0000      1.0000      1.0000      1.0000      0.0100      0.0167

ALPHA =    0.72610E 08
XM =    0.10740E-03
YM =    0.93010E 16

X4U = 0.500     X4L = 0.050     R3U = 3.000     R3L = 0.010

SIGU =    110000.     SC = 2.00     SU = 2.00     PHI =  0.10E-04
Q =  0.10E-04     DELRO =  0.10E-05     DELX2 =  0.10E-05     DELX4 =  0.10E-05
TAU =  0.0          T81 =   2100.     T82 =   2100.     XLD =   5.0
RO = 0.300     X(2) = 0.100     X(4) = 0.160     THETA =  0.4383E 05

IX =  2     NLAM =  0     NITER =  0     ITER =  50     LMOST =  0
G = 0.990     C = 1.000     CRIT = 1.00000     ALAM = 0.0

   N=     1   D=  0.390253E 01   W=  0.753760E 02   YSQ=  0.157817E 06   Z=  0.157817E 06   LSTOP=     0

    0.395568E 03    0.358484E-01

    0.365267E 02    0.160000E 00

   -0.292253E 01    0.0

RADII ARE   0.26557   0.27557   0.31142   0.32142   0.48142   0.48142   0.48142   0.48142   0.48142
L = 0.48142E 01   AV2 = 0.33641E 03   W1 = 0.72575E 02   W2 = 0.75376E 02   W3 = 0.44226E 03   WT = 0.76067E 00

   N=     2   D=  0.179112E 01   W=  0.65E371E 02   YSQ=  0.406695E 10   Z=  0.406695E 10   LSTOP=     0

    0.634969E 05    0.320644E-01

   -0.592328E 04    0.164109E 00

   -0.791122E 00    0.187450E 03

RADII ARE   0.29895   0.30895   0.34101   0.35101   0.51512   0.51512   0.51512   0.51512   0.51512
L = 0.51512E 01   AV2 = 0.35996E 03   W1 = 0.67828E 02   W2 = 0.65837E 02   W3 = 0.40391E 03   WT = 0.89119E 00

   N=     3   D=  0.114299E 01   W=  0.606518E 02   YSQ=  0.464751E 13   Z=  0.464751E 13   LSTOP=     0

   -0.214592E 07    0.297294E-01

    0.206227E 06    0.163247E 00

   -0.142992E 00   -0.919341E 04

RADII ARE   0.32371   0.33371   0.36344   0.37344   0.53669   0.53669   0.53669   0.53669   0.53669
L = 0.53669E 01   AV2 = 0.37503E 03   W1 = 0.65102E 02   W2 = 0.60652E 02   W3 = 0.38721E 03   WT = 0.96855E 00

   N=     4   D=  0.100614E 01   W=  0.592224E 02   YSQ=  0.127412E 17   Z=  0.127412E 17   LSTOP=     0
```

Table B2 (continued)

```
L = 0.54344E 01   AV2 = 0.37975E 03   W1 = 0.64293E 02   W2 = 0.59154E 02   W3 = 0.38135E 03   WT = 0.99581E 00
   N=    44   D=  0.1C0000E 01   W=  0.591540E 02   YSQ=  0.112344E 04   Z=  0.112344E 04   LSTOP=     0
    C.138723E C2     0.291225E-01
   -C.305122E C2     C.163619E 0C
    C.476837E-06     0.111302E C2
RADII ARE   0.33070   0.3407C   0.36982   0.37982   0.54344   0.54344   0.54344   0.54344   0.54344
L = 0.54344E 01   AV2 = 0.37975E 03   W1 = 0.64293E 02   W2 = 0.59154E 02   W3 = 0.38135E 03   WT = 0.99581E 00
   N=    45   D=  0.10C0C0E C1   W=  0.591540E 02   YSQ=  0.104162E 04   Z=  0.104162E 04   LSTOP=     0
   -0.315794E C2     0.291225E-01
    0.47C410E C1     0.163618E 0C
   -C.953674E-C6     0.11C735E 02
RADII ARE   0.33070   0.3407C   0.36982   0.37982   0.54344   0.54344   0.54344   0.54344   0.54344
L = 0.54344E 01   AV2 = 0.37975E 03   W1 = 0.64293E 02   W2 = 0.59154E 02   W3 = 0.38135E 03   WT = 0.99581E 00
   N=    46   D=  0.100000E 01   W=  0.591540E 02   YSQ=  0.302119E 04   Z=  0.302119E 04   LSTOP=    . 0
    0.549556E C2     0.291225E-01
    0.1C3760E 01     0.163618E 0C
   -0.953674E-C6     0.112239E 02
RADII ARE   C.33070   0.3407C   0.36982   0.37982   0.54344   0.54344   0.54344   0.54344   0.54344
L = 0.54344E 01   AV2 = 0.37975E C3   W1 = 0.64293E 02   W2 = 0.59154E 02   W3 = 0.38135E 03   WT = 0.99581E 00
   N=    47   D=  0.100000E 01   W=  0.59154CE 02   YSQ=  0.574727E 02   Z=  0.574727E 02   LSTOP=     0
   -0.663330E CC     0.291225E-01
    0.7552CCE 01     0.163618E 0C
   -0.953674E-C6     0.109567E 02
RADII ARE   0.33070   0.34070   0.36982   0.37982   0.54344   0.54344   0.54344   0.54344   0.54344
L = 0.54344E 01   AV2 = 0.37975E 03   W1 = 0.64293E 02   W2 = 0.59154E 02   W3 = 0.38135E 03   WT = 0.99581E 00
   N=    48   D=  0.100000E 01   W=  0.591540E 02   YSQ=  0.573656E 02   Z=  0.573656E 02   LSTOP=     0
   -0.628906E 00     0.291225E-01
    0.754785E C1     0.163618E 00
   -0.953674E-C6     0.109569E 02
RADII ARE   0.33070   C.34C70   C.36982   0.37982   C.54344   0.54344   0.54344   0.54344   0.54344
L = 0.54344E 01   AV2 = 0.37975E 03   W1 = 0.64293E 02   W2 = 0.59154E 02   W3 = 0.38135E 03   WT = 0.99581E 00
   N=    49   D=  0.1000C0E 01   W=  0.59154CE 02   YSQ=  0.573656E 02   Z=  0.573656E 02   LSTOP=     0
   -0.628906E CO     0.291225E-01
    0.754785E C1     0.163618E 00
```

Table B2 (continued)

```
   -0.953674E-06     0.109569E 02

RADII ARE    0.33070    0.34070    0.36982    0.37982    0.54344    0.54344    0.54344    0.54344    0.54344
L = 0.54344E 01    AV2 = 0.37975E 03    W1 = 0.64293E 02    W2 = 0.59154E 02    W3 = 0.38135E 03    WT = 0.99581E 00

   M=    50    D= 0.100000E 01    W= 0.591540E 02    YSQ= 0.573656E 02    Z= 0.573656E 02    LSTOP=     0

   -0.628906E 00     0.291225E-01

    0.754785E 01     0.163618E 00

   -0.953674E-06     0.109569E 02

RADII ARE    0.33070    0.34070    0.36982    0.37982    0.54344    0.54344    0.54344    0.54344    0.54344
L = 0.54344E 01    AV2 = 0.37975E 03    W1 = 0.64293E 02    W2 = 0.59154E 02    W3 = 0.38135E 03    WT = 0.99581E 00

   M=    51    D= 0.100000E 01    W= 0.591540E 02    YSQ= 0.573656E 02    Z= 0.573656E 02    LSTOP=     0

   -0.628906E 00     0.291225E-01

    0.754785E 01     0.163618E 00

   -0.953674E-06     0.109569E 02

RADII ARE    0.33070    0.34070    0.36982    0.37982    0.54344    0.54344    0.54344    0.54344    0.54344
L = 0.54344E 01    AV2 = 0.37975E 03    W1 = 0.64293E 02    W2 = 0.59154E 02    W3 = 0.38135E 03    WT = 0.99581E 00

RADII ARE    0.33070    0.34070    0.36982    0.37982    0.54344    0.54344    0.54344    0.54344    0.54344
L = 0.54344E 01    AV2 = 0.37975E 03    W1 = 0.64293E 02    W2 = 0.59154E 02    W3 = 0.38135E 03    WT = 0.99581E 00
```

B3. MOPT = 4

We will illustrate the use of this sequence to calculate the rupture life of an alloy exposed to a constant stress rate, a problem different from the normal one of calculating the life of a capsule. The alloy T-222 is subjected to a constantly increasing stress rate, $\sigma$, of 3500 psi/hr at a temperature of 2460°R. The input constants (some are redefined), printed on the first 14 rows of the output (Table B3) are determined as follows:

1. Let TA = A = BETA = DELTA = PS = TA = NMAX = NQ = TAU = RC = 0. Let ETA = C = G = T[illegible] = X(I), I = 1,9 but ≠ 4 = P(I), I = 1,9 = XK(I), I = 2,10 but ≠ 9 = SC = SU = [illegible]LD = 1.0.

2. The constants GAMMA, SIGU, ALPHA, YK, and XM are chosen from a fit of T-222 rupture data.

3. Let TA1 = T81 = ZM = 2460. ZM cancels TA1 in the expression for SIGT.

4. Choose THET such that $\dot{\sigma}$·THET < SIGU; therefore, THET = 25.

5. Choose LAMBDA such that LAMBDA·THET < 0.01; therefore LAMBDA = $10^{-4}$. This causes the expression [1 - EXP(-LAMBDA·T)] to produce LAMBDA·T.

6. Choose H = $10^{4}$ to cancel LAMBDA.

7. Let RO = 0 and XK(9) = 3; therefore R(3) = XK(9) and VO = 0.

8. Let X(4) = 2 and E = 2.125. This causes E to cancel the term $[R(4)^2 + R(3)^2]/[R(4)^2 - R(3)^2]$.

9. Let RR = $\dot{\sigma}$ = 3500.

10. Let PHI = $10^{-4}$, Q = $10^{-3}$, DELTH = DELOME = $10^{-2}$, and THMAX = $10^{6}$.

The computed value of THETA, determined in five iterations, is 19.33 hours; this compares with an experimentally determined value of 19.1 hours. The value of OMEGA is one since THMAX was not exceeded.

Table B3

```
T8 =    0.0     A =       0.0     ETA =  1.0     BETA =  0.0
DELTA =  0.0        C =  1.00     G =  1.00     GAMMA = 2.00
RR =    3500.0      E =  2.125    PS =  0.0      TS =    1.0
R =   10000.000     ZM = 2460.0     TA =    0.0     TAI = 2460.0
LAMBDA =    0.10000E-03     N = 100      NN = 100

MOPT =  4     NMAX =  0     NQ =  0

THE FOLLOWING THREE ROWS ARE THE CONTENTS OF THE X, P, AND XK ARRAYS

     1.0000    1.0000    1.0000    2.0000    1.0000    1.0000    1.0000    1.0000    1.0000
     1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
     1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    3.0000    1.0000

SIGU =   130000.     SC = 1.00     SU = 1.00     PHI =  0.10E-03
Q =  0.10E-02     DELTH =  0.10E-01     DELOMG =  0.10E-01     THMAX = 0.100E 07
TAU = 0.0          T81 =  2460.0     XLC =  1.000     RO = 0.0
X(2) = 1.000     X(4) = 2.000     THET =   0.250E 02
ALPHA =  0.9535E 07     YK =  0.4763E 13     XM =  0.6503E-04
      0.2500000E 02     0.1276997E 02     0.1276997E 02
      0.2293690E 02     0.5114078E 01     0.5114078E 01
      0.2110226E 02     0.2253933E 01     0.2253933E 01
      0.1986649E 02     0.1283113E 01     0.1283113E 01
      0.1936927E 02     0.1028359E 01     0.1028359E 01
      0.1933015E 02     0.1000364E 01     0.1000364E 01

THETA=   0.193302E 02     OMEGA =   0.100000E 01
```

# APPENDIX C

## List of CAPSUL Program

```
      PROGRAM CAPSUL
      DIMENSION SIGMA(500), T(500), THETA(500), THETB(500),  SIGB(500)
      REAL LMP(500), DLTH(500), DLSG(500)
      COMMON/CON/ K, SICU, ALPHA, YK, XM, PHI, C, DELTA, DELOME, THMAX,
     1TAU, T81, T92, XLC, RO, R(8), GAMMA, RR, E, PS, TS, H, ZM, LAMBDA,
     2TA, NO, X(9), P(9), XK(10), A, ETA  BETA, TNIT, TERM, T8, SC, SU,
     3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
      REAL LAMBDA, L
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSO,DH,WH
      EQUIVALENCE (XK(1),OMEGA)
      COMMON/LIM/X4U, X4L, R3U, R3L
      COMMON/NIC/CRIT, NITER, ITER
      COMMON/DELS/DELRO, DELX2, DELX4
      COMMON/CONVG/XX(19,1),RK,LMOST,LSTOP,JWAY,G
C*** VARIABLES USED IN THIS PROGRAM FOR DECIDING OPTIONS
C     MOPT DECIDES ON SEQUENCE TO USE
C       MOPT=1  LSTSQ IS USED ALONE
C       MOPT=2  LSTSQ, MAX, LIMIT, AND DF ARE USED
C       MOPT=3  MAX, LIMIT, AND DF ARE USED
C       MOPT=4  THETA ALONE IS USED
C     NMAX DECIDES ON SEQUENCE TO USE IN MAX
C       NMAX=1  THREE INDEPENDENT VARIABLES
C       NMAX=2  X(4) IS SPECIFIED
C       NMAX=3  R(8) IS CONSTANT
C       NMAX=4  R(3) IS CONSTANT
C       NMAX=5  R7FRO IS TO BE COMPUTED
C     NO DECIDES FUNCTION TO BE MAXIMIZED
C       NO=1  POWER/PROJECTED AREA
C       NO=2  POWER/VOLUME OF CIRCUMSCRIBED RECTANGULAR PARALLELEPIPED
C       NO=3  POWER/WEIGHT
   90 READ(5C,100) MOPT, NMAX, NO
      NNN=1
      IF(MOPT-1) 13,10,91
C*** COMMON CONSTANTS FOR MOPT=2,3,AND 4 FOLLOW
   91 READ(50,111) GAMMA,RR,E,PS,TS,H,ZM,TA
      READ(50,111) T8,A,ETA,BETA,DELTA,C,G,TA1
      READ(5C,150) LAMBDA,N,NN
      READ(50,112) (X(I),I=1,9),(P(I),I=1,9),(XK(I),I=2,10)
  100 FORMAT(3I2)
  110 FORMAT(3E10.0)
  111 FORMAT(8F10.0)
  112 FORMAT(9F8.0)
  150 FORMAT(E10.3, 2I3)
      WRITE(51,892) T8, A, ETA, BETA, DELTA, C, G, GAMMA, RR, E, PS, TS,
     1 H, ZM, TA, TA1, LAMBDA, N, NN
  892 FORMAT('1T8 =',F7.1,5X,'A =',F10.1,5X,'ETA =',F5.1,5X,'BETA =',
     1F8.4,/,' DELTA =',F8.4,5X,'C =',F6.2,5X,'G =',F6.2,5X,'GAMMA =',
     2F5.2,/,' RR =',F10.1,5X,'E =',F7.3,5X,'PS =',F6.2,5X,'TS =',F6.1,
     3/,' H =',F12.3,5X,'ZM =',F7.1,5X,'TA =',F7.1,5X,'TA1 =',F7.1,
```

```
     4/,' LAMBDA =',E15.5,5X,'N =',I4,5X,' NN =',I4)
      WRITE(51,890) MOPT,NMAX,NO
  103 FORMAT(8HOALPHA =,E15.5,/,5H XM =,E15.5,/,5H YK =,E15.5)
  890 FORMAT('0MOPT =',I3,5X,'NMAX =',I3,5X,'NO =',I3)
      WRITE(51,891) (X(I),I=1,9),(P(I),I=1,9),(XK(I),I=2,10)
  891 FORMAT('0THE FOLLOWING THREE ROWS ARE THE CONTENTS OF THE X, P, AN
     1D XK ARRAYS',//,(9F12.4))
      GO TO (10,10,20,30), MOPT
C*** CONSTANTS FOR LSTSG ROUTINE
   10 READ(50,101) K, SIGU, GAMMA, (SIGMA(I),T(I),THETA(I),I=NNN,K)
      IF(K.EQ.0) GO TO 95
  101 FORMAT (I3,2F10.0,/(3F20.0))
      CALL LSTSQ (K,SIGU,GAMMA,SIGMA,T,THETA,ALPHA,XM,XKO)
      SELTH=0.0
      SELSG=0.0
      RESIG=0.0
      DO1109 I=1,K
      EX=1.0/XM/T(I)
      THETB(I)=1.0/XKO*(.01*ALPHA/SIGMA(I)/SIGU)**EX*(SIGU**
     1 GAMMA-SIGMA(I)**GAMMA)**(EX/GAMMA)*100.**EX
      SIGB(I)= ((ALPHA**GAMMA*SIGU**GAMMA)/(ALPHA**GAMMA + SIGU**GAMMA*
     1(XKO*THETA(I))**(GAMMA*XM*T(I))))**(1.0/GAMMA)
      LMP(I)=T(I)*ALOG10(XKO*THETA(I))
      DLSG(I)=ALOG10(SIGMA(I)/SIGB(I))
      SELSG=SELSG+(DLSG(I))**2
      DLTH(I)=ALOG10(THETA(I)/THETB(I))
      SELTH=SELTH+(DLTH(I))**2
      RESIG=RESIG +((SIGMA(I)-SIGB(I))**2)/SIGB(I)**2
 1109 CONTINUE
      SELTH=SQRT(SELTH/(K-3.0))
      SELSG=SQRT(SELSG/(K-3.0))
      RESIG=SQRT(RESIG/(K-3))
      WRITE(51,522) K, SIGU, GAMMA, ALPHA, XM, XKO, SELTH, SELSG,
     1RESIG,(I, SIGMA(I), T(I), THETA(I), LMP(I), THETB(I), SIGB(I),
     2DLTH(I), DLSG(I), I=1,K)
  522 FORMAT('1K =',I4,5X,'SIGU =',F9.0,5X,'GAMMA =',F5.2 ,/,
     1' ALPHA =',E15.6,5X,'XM =',E15.6,5X,'XKO =',E15.6,/,
     2' SELTH =',E15.6,5X,'SELSG =',E15.6,5X,'RESIG =',E15.6,//,
     3'  I',T6,'SIGMA(I)',T16,'T(I)',T23,'THETA(I)',T38,'LMP(I)',
     4T51,'THETB(I)',T66,'SIGB(I)',T80,'DLTH(I)',T94,'DLSG(I)',/,
     5(I4,F9.0,F7.0,F11.2,2X,E12.5,2X,E12.5,2X,E12.5,2X,E12.5,2X,E12.5))
      NNN=K+1
      GO TO 10
   95 GO TO (90,25),MOPT
C*** CONSTANTS FOR USE IN MAX
   20 READ(50,110) ALPHA,YK,XM
      WRITE(51,103) ALPHA, XM, YK
   25 CONTINUE
      READ(50,111) X4U, X4L, R3U, R3L
```

```
      WRITE(51,126) X4U, X4L, R3U, R3L
  126 FORMAT('0X4U =',F6.3,5X,'X4L =',F6.3,5X,'R3U =',F6.3,5X,'R3L =',
     1 F6.3)
      READ(50,241) SIGU, SC, SU, PHI, C, DELRO, DELX2, DELX4, TAU, T81,
     1T82, XLU, RO, X(2), X(4), THET
  241 FORMAT(4E20.0)
      WRITE(51,242) SIGU, SC, SU, PHI, 0, DELRO, DELX2, DELX4, TAU, T81,
     1T82, XLD, RO, X(2), X(4), THET
  242 FORMAT('0SIGU ='F10.0,5X,'SC ='F5.2,5X,'SU ='F5.2,5X,'PHI ='
     1E10.2,/,' 0 ='E10.2,5X,'DELRO ='E10.2,5X,'DELX2 ='E10.2,5X,
     2'DELX4 ='E10.2,/,' TAU ='E11.3,5X,'T81 ='F8.0,5X,'T82 ='F8.0,5X
     3'XLD ='F6.1,/,' RO ='F6.3,5X,'X(2) ='F6.3,5X,'X(4) ='F6.3,5X,
     4'THETA ='E12.4)
      READ(50,402) IX, NLAM, NITER, ITER, LMOST, G, C, CRIT, ALAM
  402 FORMAT(5I4,/,4F10.0)
      WRITE(51,403)IX, NLAM, NITER, ITER, LMOST, G, C, CRIT, ALAM
  403 FORMAT('0IX ='I4,5X,'NLAM ='I4,5X,'NITER ='I4,5X,'ITER ='I4,5X,
     1'LMOST ='I4,/' G ='F6.3,5X,'C ='F6.3,5X,'CRIT ='F8.5,5X,'ALAM ='
     2 F6.3)
  90C NMAX1=NMAX
      JAM=0
      CALL VSU
   61 CALL DR
      JAM=JAM+1
      IF(JAM.GT.100) GO TO 66
      IF(DH.GT.4.0) 62,62
   63 IF(DH.LT.0.3) 64,65
   66 WRITE(51,4) RO, (R(I),I=1,8), L, AV2, W1, W2, W3, WT
      GO TO 90
   62 X(2) = 0.95*X(2)
      CALL VSU
      GO TO 61
   64 X(2) = 1.05*X(2)
      X(4) = 0.95*X(4)
      CALL VSU
      GO TO 61
   65 CONTINUE
      CALL MAX(NLAM)
      NO1=NC $ NC=0
      CALL WR
      CALL LIMIT
      NC=NO1
      WRITE(51,4) RO, (R(I),I=1,8), L, AV2, W1, W2, W3, WT
    4 FORMAT('0RADII ARE',9F10.5,/,' L ='E12.5,3X,'AV2 ='E12.5,3X,
     1'W1 ='F12.5,3X,'W2 ='E12.5,3X,'W3 ='E12.5,3X,'WT ='E12.5)
      IF(NMAX.EQ.NMAX1) 9C,900
C***  THE FOLLOWING ARE THE CONSTANTS USED IN CALCULATING THETA
   30 READ(5C,110) ALPHA,YK,XM
      READ(5C,210) SIGU, SC, SU, PHI, 0, DELTH, DELOME, THMAX, TAU, TR1,
```

```
     1       XLO, RO, X(2), X(4),THET
210 FORMAT(4F20.0)
    WRITE(51,211) SIGU, SC, SU, PHI, Q, DELTH, DELOME, THMAX, TAU,
     1 TA1, X_O, RC, X(2), X(4), THET, ALPHA, YK, XM
211 FORMAT('0SIGU =',F10.0,5X,'SC =',F5.2,5X,'SU =',F5.2,5X,'PHI =',
     1 F10.2,/,' Q =', F10.2,5X,'DELTH =',F10.2,5X,'DELOME =',E10.2,
     25X,'THMAX =', E10.3,/,' TAU =', E10.3,5X,'T81 =',F8.1,5X,'XLO =',
     3F7.3,5X,'RC =',F6.3,/,' X(2) =',F6.3,5X,'X(4) =',F6.3,5X,'THET =',
     4 F12.3,/,' ALPHA =',E12.4,5X,'YK =',E12.4,5X,'XM =',E12.4)
    CALL THETAC
    WRITE(51,72) THET,OMEGA
 72 FORMAT('0THETA=',F15.6,5X,'OMEGA =',E15.6)
    GO TO 90
 13 END

    SUBROUTINE LSTSQ (K,SIGU,GAMMA,SIGMA,T,THETA,ALPHA,XM,XKO)
    DIMENSION SIGMA(500), T(500), THETA(500), Y(500), Z(500),
     1 A(3,3), X(3)
    SIG = SIGU**GAMMA
    DO 10 I=1,K
    SIGI = SIGMA(I)**GAMMA
    Y(I) = ALOG(SIG*SIGI/(SIG-SIGI))
 10 Z(I) = T(I)*ALOG(THETA(I))
    DO 20 I=1,3 $ X(I) = 0.0
    DO 20 J=1,3
 20 A(I,J)=0.0
    DO 30 I=1,K
    A(1,2) = A(1,2) + T(I)
    A(1,3) = A(1,3) + Z(I)
    A(2,2) = A(2,2) + T(I)**2
    A(2,3) = A(2,3) + T(I)*Z(I)
    A(3,3) = A(3,3) + Z(I)**2
    X(1) = X(1) - Y(I)
    X(2) = X(2) - T(I)*Y(I)
 30 X(3) = X(3) - Z(I)*Y(I)
    A(1,1) = -K $ A(2,1) = -A(1,2)
    A(3,1) = -A(1,3) $ A(3,2) = A(2,3)
    CALL MATC(A,X,3,1,DET,3,3)
    A1=X(1) $ A2=X(2) $ A3=X(3)
    ALPHA = EXP(A1/GAMMA)
    XM= A3/GAMMA
    XKO=EXP(A2/GAMMA/XM)
    RETURN
    END

    SUBROUTINE CONVERT(NMAX,X,ALF,R,RO)
    DIMENSION X(9), ALF(19),R(8)
    GO TO (10,20,30,40,50),NMAX
 10 RO=ALF(1)
```

```
      X(2)=ALF(2)
      X(4)=ALF(3)
      RETURN
   20 RO=ALF(1)
      X(2)=ALF(2)
      RETURN
   30 X(2)=ALF(1)
      X(4)=ALF(2)
      RO=R(8)-X(1)-X(2)-X(3)-X(4)-X(5)-X(6)-X(7)-X(8)
      RETURN
   40 X(2)=ALF(1)
      X(4)=ALF(2)
      RO=R(3)-X(1)-X(2)-X(3)
      RETURN
   50 X(2)=ALF(1)
      X(4)=ALF(2)
      CALL RZERO
      RETURN
      END

      SUBROUTINE WR
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      COMMON/CON/ K, SIGO, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
     1TAU, TR1, TR2, XLD, RO, R(8), GAMMA, RR, E, PS, TS, H, ZM, LAMBDA,
     2TA, NO, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
     3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
      REAL LAMBDA, L
      CALL CONVERT(NMAX,X,ALF,K,RO)
      R(1)=RO+X(1)
      DO 40 I=2,8
   40 R(I)=R(I-1)+X(I)
      XL2=2.0*(XLD-.575)*R(8)-.5*(1.0+R(4))-2.0*(X(1)+X(9))
      AV2=A*XL2*3.14159*(R(2)**2-R(1)**2)
      GO TO (10,20,30) NC
   10 WH = AV2/(4.0*XLD*R(8)*(R(8) + BETA))
      IF(NC.NE.0) RETURN
      W1=WH
   20 WH = AV2/(4.0*XLD*R(8)*(R(8) + BETA)*(R(8) + DELTA))
      IF(NO.NE.0) RETURN
      W2=WH
   30 CALL WHT
      WH = AV2/WT
      W3=WH
      RETURN
      END

      SUBROUTINE WHT
      COMMON/CON/ K, SIGO, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
     1TAU, TR1, TR2, XLD, RO, R(8), GAMMA, RR, E, PS, TS, H, ZM, LAMBDA,
```

```
     2TA, NQ, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
     3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3  THET, WT, TA1
      REAL L,L2,L4,V(9)
      L = 2.0*XLD*R(8)
      L4 = 2.0*(XLD - .575)*R(8)
      L2 = L4 - .5*(1.0 + R(4)) - 2.0*(X(1) + X(9))
      R(1) = RC + X(1)
      DO 60 I = 2,8
   60 R(I) = R(I-1) + X(I)
      VO = 3.1416*RO**2*L2 + 2.409*((R(3) - XK(9))**3) + 1.5708*((R(3)-
     1 XK(9))**2)*(1.0+R(4))
      V(1) = 3.1416*(L2 + 2.0*X(1))*(R(1)**2 - RO**2)
      V(2) = 3.1416*L2*(R(2)**2 - R(1)**2)
      V(3) = 3.1416*(L2 + 2.0*X(1))*(R(3)**2 - R(2)**2)
      V(4) = 3.1416*L4*R(4)**2 + 2.409*R(4)**3 - V(1) - V(2) - V(3)
     1 - 6.283*R(3)**2*X(9) - VO
      V(5) = 3.1416*L4*(R(5)**2 - R(4)**2) + 2.409*(R(5)**3 - R(4)**3)
      V(6) = 3.1416*L4*(R(6)**2 - R(5)**2) + 2.409*(R(6)**3 - R(5)**3)
      V(7) = 3.1416*(L -.575*R(8))*(R(7)**2 - R(6)**2) + 3.1416*R(7)**2
     1 *X(7) + 1.204*(R(7)**3 - R(6)**3)
      V(8) = 3.1416*(L -.575*R(8))*(R(8)**2 - R(7)**2) + 3.1416*R(8)**2
     1 *X(8) + 1.204*(R(8)**3 - R(7)**3)
      V(9) = 6.283*R(3)**2*X(9)
      WT = 0.0
      DO 70 I=1,9
   70 WT = WT + V(I)*P(I)
      RETURN
      END

      SUBROUTINE VSU
      COMMON/CON/ K, SIGU, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
     1TAU, T81, T82, XLD, RO, R(8), GAMMA, RR, E, PS, TS, H, ZM, LAMBDA,
     2TA, NQ, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
     3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
      COMMON/SETUP/W,D,WEL(19),DEL(19),WSD(19,19),DSD(19,19),DEL(19),WEL
     1(19)
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,CH,WH
      COMMON/DELS/DELRO, DELX2, DELX4
      TERM=THET$TNIT=0.0
      GO TO (10,20,30,40,50),NMAX
   10 DEL(1)=WEL(1)=DELRO
      DEL(2)=WEL(2)=DELX2
      DEL(3)=WEL(3)=DELX4
      ALF(1)=RO
      ALF(2)=X(2)
      ALF(3)=X(4)
      IX=3
      RETURN
   20 DEL(1)=WEL(1)=DELRO
```

```
      DEL(2)=WEL(2)=DELX2
      ALF(1)=RO
      ALF(2)=X(2)
      IX=2
      RETURN
   30 DEL(1)=WEL(1)=DELX2
      DEL(2)=WEL(2)=DELX4
      ALF(1)=X(2)
      ALF(2)=X(4)
      IX=2
      RETURN
   40 DEL(1)=WEL(1)=DELX2
      DEL(2)=WEL(2)=DELX4
      RO=R(8)-X(1)-X(2)-X(3)
      ALF(1)=X(2)
      ALF(2)=X(4)
      IX=2
      RETURN
   50 DEL(1)=WEL(1)=DELX2
      DEL(2)=WEL(2)=DELX4
      ALF(1)=X(2)
      ALF(2)=X(4)
      IX=2
      RETURN
      END

      SUBROUTINE RZERO
      COMMON/CON/ K, SIGU, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
     1TAU, T81, T82, XLD, RO, R(8), GAMMA, RR, E, PS, TS, H, Z, LAMBDA,
     2TT  NQ, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
     3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
      REAL L2
C*** SUBROUTINE CALCULATES A RO BY NEWTONS METHOD
      M=0
      HSIN = ALOG(XLD + SQRT(XLD**2 + 1.0))
      B = 4.0*XK(10)*XLD*(T82 - TA)/A /HSIN
      RO = B/(4.0*X(2)*(XLD-.825)) - X(1) - X(2)/2.0
   35 R(8) = RO
      DO 10 I=1,8
   10 R(8) = R(8) + X(I)
      L2 = 2.0*(XLD - .575)*R(8) - .5*(1.0 + RO + X(1) + X(2) + X(3) +
     1X(4)) - 2.0*(X(1) + X(9))
      V2PIE = L2*(2.0*X(2)*(RO + X(1)) + X(2)**2)
      IF(ABS((B*R(8)-V2PIE)/B/R(8)).GT.1.0E-6) 30, 31
   30 RO = RO -(B*R(8)-V2PIE)/(B-2.0*(XLD-.825)*2.*X(2)*(RO+X(1))+
     1X(2)**2) - 2.0*X(2)*L2)
      M=M+1
      IF(M.GT.10) GO TO 31
      GO TO 35
```

```
 31 CONTINUE
    IF(RO.GT.0.0) RETURN
    RO=.20
    RETURN
    END

    SUBROUTINE LIMIT
    COMMON/LIM/X4U, X4L, R3U, R3L
    COMMON/CON/ K, SIGU, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
   1TAU, T81, T82, XLD, RO, R(8), GAMMA, RR, E, PS, TS, H, Z, LAMBDA,
   2TA, NQ, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
   3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
    GO TO (10,20,30),NMAX
 10 IF(X(4).GT.X4U) 40,50
 40 X(4) = X4U $ NMAX = 2
    RETURN
 50 IF(X(4).LT.X4L) 60,70
 60 X(4) = X4L $ NMAX= 2
    RETURN
 70 IF(R(3).GT.R3U) 80, 90
 80 R(3) = R3U $ NMAX = 4
    RETURN
 90 IF(R(3).LT.R3L) 100, 110
100 R(3) = R3L $ NMAX=4
110 RETURN
 20 IF(R(3).GT.R3U) 120, 130
120 R(3) = R3U $ GO TO 160
130 IF(R(3).LT.R3L) 140, 150
140 R(3) = R3L $ GO TO 160
150 RETURN
160 RO = R(3) - X(1) - X(2) - X(3)
    CALL DF
    RETURN
 30 IF(X(4).GT.X4U) 170,180
170 X(4) = X4U $ GO TO 210
180 IF(X(4).LT.X4L) 190,200
190 X(4) = X4L $ GO TO 210
200 RETURN
210 CALL DF
    RETURN
    END

    SUBROUTINE DF
    COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
    COMMON/CON/ K, SIGU, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
   1TAU, T81, T82, XLD, RO, R(8), GAMMA, RR, E, PS, TS, H, Z, LAMBDA,
   2TA, NQ, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
   3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
    COMMON/DELS/DELRO, DELX2, DELX4
```

```
33 CALL DR
   D1=DH
   IF(ABS(D1-1.0).LE.Q) 30, 31
31 X(2) = X(2) + DELX2
   CALL DR
   X(2) = X(2) - DELX2 - (D1-1.0)/(DH-D1)*DELX2
   GO TO(10,10,33),NMAX
10 RO=R(3)-X(1)-X(2)-X(3)
   GO TO 33
30 RETURN
   END

   SUBROUTINE THETAC
   COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
   EQUIVALENCE (XK(1),OMEGA)
   COMMON/CON/ K, SIGU, ALPPA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
  1TAU, T81, T82, XLD, RO, R(8), GAMMA, RR, E, PS, TS, H, ZM, LAMBDA,
  2TA, NQ, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
  3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
   REAL LAMPDA, L
   NSTOR=N $ T80=T8
   TAO=TA
   D3=D4=TNIT=0.0
   L=0
   OMEGA =1.0
   SC1 = SC $ SU1 = SU
   TERM=TAU $ N=NN
   IF(TNIT.EQ.TERM) GO TO 10
   CALL DR
   D3 = DH
10 TNIT=TAU $ T8=T81
   TA=TA1
20 TERM=THET $ N=NSTOR
   CALL DR
   D4 = DH+D3
   WRITE(51,100) THET,D4,DH
100 FORMAT(3E20.7)
   IF(D4.LT.1.0.AND.THET.GE.THMAX) GO TO 40
   IF(ABS(D4-1.0).LE.Q) RETURN
   TERM = TERM + DELTH
   CALL DR
   D5=D3+DH
   THET = TERM - DELTH - (D4-1.0)*DELTH/(D5-D4)
   GO TO 20
40 TNIT=0.0 $ T8=T80
   TA=TAO
   TERM=TAU $ N=NN
   OMEGA1=OMEGA+DELOME
   SC=OMEGA1*SC1 $ SU=OMEGA1*SU1
```

```
      D3=D4=0.C
      IF(INIT.EC.TERM) GC TO 41
      CALL DR
      D3=DH
      SC=OMEGA*SC1 $ SU=CMEGA*SU1
      CALL DR
      D4=DH
   41 TNIT=TAU $ T8=T81
      TA=TA1
      TERM=THMAX $ N=NSTCR
      SC=OMEGA1*SC1 $ SU=OMEGA1*SU1
      CALL DR
      D5=DH
      SC=OMEGA*SC1 $ SU=CMEGA*SU1
      CALL DR
      D6=DH
      D9=D4+D6
      WRITE(51,100) TERM, D9, OMEGA
      IF(ABS(D4+D6-1.0).GT.0) 30, 31
   30 OMEGA=CMEGA - (D4+D6-1.0)*DELOME/(D3+D5-D4-D6)
      L=L+1
      IF(L.GT.10) 31, 40
   31 THET = TERM
      RETURN
      END

      SUBROUTINE DR
      COMMON/CON/ K, SIGU, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
     1TAU, T81, T82, XLD, RO, R(8), GAMMA, RK, E, PS, TS, H, ZM, LAMBDA,
     2TA, NO, X(9), P(9), XK(10), A, ETA, BETA, TNIT, TERM, T8, SC, SU,
     3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      REAL LAMBDA, L,L2,L4
      EQUIVALENCE (XKC,YK), (A0,A,A2)
      DIFF=TERM-TNIT
      IF(NMAX.NE.0.AND.NC.NE.0) CALL CONVERT(NMAX,X,ALF,R,RO)
      D1=SUM=C.0 $ NT=1
      R(1) = X(1) + RO
  110 DO 80 I=2,8
   80 R(I) = R(I-1) + X(I)
      DO 190 J=4,8
  190 SUM = SUM + 1.0/XK(J)*ALOG(R(J)/R(J-1))
      L2 = 2.0*(XLO -.575)*R(9) - .5*(1.0+R(4)) - 2.0*(X(1) + X(9))
      V0 = 3.1416*RC**2*L2 + 2.409*((R(3)-XK(9))**3) + 1.5708*((R(3)-
     1 XK(9))**2)*(1.C+R(4))
      V2 = 3.14159*L2*(R(2)**2 - R(1)**2)
      T40=TA+(T8-TA+A*V2/6.2832/L2*SUM)
      TO = T40 + A0*V2/(6.28318*L2*XK(3))*ALOG(R(3)/R(2)) + A0/4.0/XK(2)
     1*(R(2)**2 - R(1)**2 - 2*R(1)**2*ALOG(R(2)/R(1)))
```

```
      C1 =RR/E/(VO + ETA*V2)*(R(4)**2 + R(3)**2)/(R(4)**2 - R(3)**2)
      C2 = T8 - TA + A2*V2/6.28318/L2*SUM
      D2 = 0.C $ N1 = N
      C3=PS*(V0+ETA*V2)/RR/TS
      C4=P(2)*V2*H/ZM
      CALL SETFAULT(5,5HEU=-1)
      DO 9C T=INIT,TERM,DIFF
      SIGT = C1*(C3 + C4*(1.0 - EXP(-LAMBDA*T)))*(TA +(TO - TA)*
     1EXP(-LAMBDA*T))
      T4 = TA + C2*EXP(-LAMBDA*T)
   90 D1 = D1 + XKO*(SC**GAMMA*SIGU**GAMMA*SIGT**GAMMA/ALPHA**GAMMA/
     1 (SIGU**GAMMA - SU**GAMMA*SIGT**GAMMA))**(1.0/GAMMA/XM/T4)
      SIGG = SIGU**GAMMA $ SUGA = SU**GAMMA
  302 NPT = 1 $ D2 = 0.0
      DO 10 J=1,N1-1
      EX=EXP(-LAMBDA*(TNIT+J*DIFF/N1))
   29 SIGT = C1*(C3 + C4*(1.0 - EX))*(TA +(TO - TA)*EX)
      T4 = TA + C2*EX
      F =XKO*(SC**GAMMA*SIGG*SIGT**GAMMA/ALPHA**GAMMA/(SIGG - SUGA*
     1 SIGT**GAMMA))**(1.0/GAMMA/XM/T4)
      GO TO (11,12),NPT
   11 D2 = D2 + 4.C*F $ NPT = 2 $ GO TO 10
   12 D2 = D2 + 2.0*F $ NPT = 1
   10 CONTINUE
      GO TO (300,301),NT
  300 D3=DIFF*(D1+D2)/3.0/N1 $ N1=2*N1 $NT=2 $ GO TO 302
  301 D2=D1+D2 $ D2=DIFF*D2/3.0/N1
      IF(ABS((C2-C3)/D2).LT.PHI) 304, 305
  305 D3 = D2 $ N1=2*N1
      IF(N1.GT.16*N) 306, 302
  306 WRITE(51,247)
  247 FORMAT(' FAILED TO CONVERGE')
  304 DH = D2
      CALL SETFAULT(4,4HEU=0)
      END

      SUBROUTINE MAX(NLAM)
      COMMON/CONVG/XX(19,1),RK,LMOST,LSTOP,JWAY,G
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      COMMON/MATQ/A(19,19),X(19,1),NR,YSQ,Z
      COMMON/NIC/CRIT, NITER, ITER
      CRITSQ=CRIT**2
      CSQ=C*C
      NR=IX+1
      M=1
    4 CALL SETUP
      IF(NLAM)10,5,10
    5 CALL GTALAM
C       FIRST LAMBDA COMPUTED IN SUBR. GTALAM
```

```
      NLAM=1
C     FIRST LAMBDA FROM INPUT
   10 CALL VECT
      CALL CONVG
      CALL OUTPUT
      IF(JWAY)100,30,12
   12 IF(NITER)15,15,20
C     TEST ON NUMBER OF OUTER ITERATION
   15 IF(M-ITER)25,25,100
C     CONVERENCE CRITERIA TEST
   20 IF(Z-CRITSQ)100,100,15
   25 CALL ARITH
      CALL MATC(A,X,NR,1,DET,IS,19)
      M=M+1
      LSTOP=0
   30 CALL STEP
      GO TO 4
  100 RETURN
      END MAX

      SUBROUTINE SETUP
      COMMON/SETUP/W,D,WFD(19),DFD(19),WSD(19,19),DSD(19,19),DEL(19),WEL
     1(19)
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSC,DH,WH
C     COMPUTES QUANTITIES IN LABELED COMMON SETUP
      CALL DR
C     COMPUTE FIRST AND SECOND DERIVATIVES OF FCN D
      D=DH
      DO 40 I=1,IX
      STORI=ALF(I)
      ALF(I)=ALF(I)+DEL(I)
      CALL DR
      DP=DH
      ALF(I)=STORI-DEL(I)
      CALL DR
      DM=DH
      DFD(I)=(DP-DM)/(2.*DEL(I))
      DSD(I,I)=(DP+DM-2.*D)/(DEL(I)**2)
      J=I
    4 IF(IX-J)40,1,1
    1 IF(J-I)13,3,2
   13 DSD(I,J)=DSD(J,I)
    3 J=J+1
      GO TO 4
    2 STORJ=ALF(J)
      ALF(J)=ALF(J)-DEL(J)
      CALL DR
      D3=DH
      ALF(J)=STORJ+DEL(J)
```

```
      CALL DR
      D2=DH
      ALF(I)=STORI+DEL(I)
      CALL DR
      D1=DH
      ALF(J)=STORJ-DEL(J)
      CALL DR
      D4=DH
      ALF(I)=STORI-DEL(I)
      ALF(J)=STORJ
      DSD(I,J)=(D1+D3-D2-D4)/(4.*DEL(I)*DEL(J))
      GO TO 3
   40 ALF(I)=STORI
      CALL WR
C     COMPUTE FIRST AND SECOND DERIVATIVES OF FCN W
      W=WH
      DO 60 I=1,IX
      STORI=ALF(I)
      ALF(I)=ALF(I)+WEL(I)
      CALL WR
      WP=WH
      ALF(I)=STORI-WEL(I)
      CALL WR
      WM=WH
      WFD(I)=(WP-WM)/(2.*WEL(I))
      WSD(I,I)=(WP+WM-2.*W)/(WEL(I)**2)
      J=1
   24 IF(IX-J)60,21,21
   21 IF(J-I)213,23,22
  213 WSD(I,J)=WSD(J,I)
   23 J=J+1
      GO TO 24
   22 STORJ=ALF(J)
      ALF(J)=ALF(J)-WEL(J)
      CALL WR
      W3=WH
      ALF(J)=STORJ+WEL(J)
      CALL WR
      W2=WH
      ALF(I)=STORI+WEL(I)
      CALL WR
      W1=WH
      ALF(J)=STORJ-WEL(J)
      CALL WR
      W4=WH
      ALF(I)=STORI-WEL(I)
      ALF(J)=STORJ
      WSD(I,J)=(W1+W3-W2-W4)/(4.*WEL(I)*WEL(J))
      GO TO 23
```

```
   60 ALF(I)=STORI
      RETURN
      END

      SUBROUTINE GTALAM
      COMMON/SETUP/W,D,WFD(19),DFD(19),WSD(19,19),DSD(19,19),DEL(19),WEL
     1(19)
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
C     COMPUTES AVERAGE LAMBDA FOR 1ST ITERATION
      SUM=0.
      DEN=0.
      DO14J=1,IX
      IF(DFD(J))13,14,13
   13 SUM=SUM+WFD(J)/DFD(J)
      DEN=DEN+1.
   14 CONTINUE
      IF(DEN)16,15,16
   15 CALL ERROR
   16 ALAM=SUM/DEN
      RETURN
      END

      SUBROUTINE VECT
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      COMMON/SETUP/W,D,WFD(19),DFD(19),WSD(19,19),DSD(19,19),DEL(19),WEL
     1(19)
      COMMON/MATC/A(19,19),Y(19,1),NR,YSQ,Z
C     AX=Y     NR=IX+1
C     COMPUTES Y,YSQ,AND Z
      YSQ=0.
      DO15J=1,IX
      Y(J,1)=-(WFD(J)-ALAM*DFD(J))
   15 YSQ=YSQ+Y(J,1)*Y(J,1)
      Y(NR,1)=C-D
      YSQ=YSQ+Y(NR,1)**2
      Z=YSQ/CSQ
      RETURN
      END

      SUBROUTINE CONVG
      COMMON/CONVG/XX(19,1),RK,LMOST,LSTOP,JWAY,G
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      COMMON/MATC/A(19,19),X(19,1),NR,YSQ,Z
C     TESTS WHETHER MORE INNER ITERATIONS ARE REQUESTED AND COMPUTES A
C     VECTOR INCREMENT ONE HALF THE MAGNITUDE OF THE LAST SUCH VECTOR
C     INCREMENT TRIED, IF NEEDED
      IF(M-1) 1,2,8
    1 CALL ERROR
    8 IF(LMOST.GT.0) GO TO 3
```

```
    2 RK=YSQ*G
      JWAY=1
      GO TO 100
    3 IF(YSQ-RK)2,4,4
    4 LSTOP=LSTOP+1
      IF(LMOST-LSTOP)5,6,6
    5 JWAY=-1
      GO TO 100
    6 DO 7 K=1,NR
      XX(K,1)=.5*XX(K,1)
    7 X(K,1)=XX(K,1)
      JWAY=0
  100 RETURN
      END

      SUBROUTINE OUTPUT
      COMMON/CON/ K, SIGU, ALPHA, YK, XM, PHI, Q, DELTA, DELOME, THMAX,
     1TAU, T81, T82, XLD, RO, R(8), GAMMA, RK, E, PS, TS, H, ZM, LAMBDA,
     2T7, NQ, X(9), P(9), XK(10), ABC, ETA, BETA, TNIT, TERM, T8, SC, SU,
     3N, NN, DELTH, AV2, NMAX, L, W1, W2, W3, THET, WT, TA1
      COMMON/SETUP/W,D,WFD(19),DFD(19),WSD(19,19),DSD(19,19),DEL(19),
     1WEL(19)
      COMMON/MATQ/A(19,19),Y(19,1),NR,YSQ,Z
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      COMMON/CONVG/XX(19,1),RK,LMOST,LSTOP,JWAY,G
      WRITE (51,1)M,D,W,YSQ,Z,LSTOP
    1 FORMAT(1H03X2HM=I6,5H    D=E14.6,5H    W=E14.6,7H    YSQ=E14.6,
     15H    Z=E14.6,9H    LSTOP=I6)
      WRITE(51,2)((Y(I,1),ALF(I)),I=1,NR)
    2 FORMAT(1H02F16.6)
      NQ1=NC $ NC=0
      CALL WR
      NQ=NQ1
      WRITE(51,4) RO, (R(I),I=1,8), L, AV2, W1, W2, W3, WT
    4 FORMAT('ORADII ARE',9F10.5,/,' L =',E12.5,3X,'AV2 =',E12.5,3X,
     1'W1 =',E12.5,3X,'W2 =',E12.5,3X,'W3 =',E12.5,3X,'WT =',E12.5)
      RETURN
      END

      SUBROUTINE ARITH
      COMMON/SETUP/W,D,WFD(19),DFD(19),WSD(19,19),DSD(19,19),DEL(19),WEL
     1(19)
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      COMMON/MATQ/A(19,19),X(19,1),NR,YSQ,Z
C     COMPUTES ELEMENTS OF MATRIX A, NR=IX+1
      DO 10 I=1,IX
      DO 11 J=1,IX
   11 A(I,J)=WSD(I,J)-ALAM*DSD(I,J)
      A(I,NR)=-DFD(I)
```

```
10 A(NP-I)=DFD(I)
   A(NR,NR)=0.
   RETURN
   END

   SUBROUTINE MATO(A,X,NR,NV,DET,NA,NX)
   DIMENSION A(10),X(10)
   DET=1.0
   NR1=NR-1
   DO 5 K=1,NR1
   IR1=K+1
   PIVOT=0.0
   DO 6 I=K,NR
   IK=(K-1)*NA+I
   Z=ABSF(A(IK))
   IF(Z-PIVOT)6,6,7
 7 PIVOT=Z
   IPR=I
 6 CONTINUE
   IF(PIVOT)8,9,8
 9 DET=0.0
   RETURN
 8 IF(IPR-K)10,11,10
10 DO 12 J=K,NR
   IPRJ=(J-1)*NA+IPR
   Z=A(IPRJ)
   KJ=(J-1)*NA+K
   A(IPRJ)=A(KJ)
12 A(KJ)=Z
   DO 13 J=1,NV
   IPRJ=(J-1)*NX+IPR
   Z=X(IPRJ)
   KJ=(J-1)*NX+K
   X(IPRJ)=X(KJ)
13 X(KJ)=Z
   DET=-DET
11 KK=(K-1)*NA+K
   DET=DET*A(KK)
   PIVOT=1.0/A(KK)
   DO 14 J=IR1,NR
   KJ=(J-1)*NA+K
   A(KJ)=A(KJ)*PIVOT
   DO 14 I=IR1,NR
   IJ=(J-1)*NA+I
   IK=(K-1)*NA+I
14 A(IJ)=A(IJ)-A(IK)*A(KJ)
   DO 5 J=1,NV
   KJ=(J-1)*NX+K
   IF(X(KJ)) 15,5,15
```

```
   15 X(KJ)=X(KJ)*PIVOT
      DO 16 I=IR1,NR
      IJ=(J-1)*NX+I
      IK=(K-1)*NA+I
   16 X(IJ)=X(IJ)-A(IK)*X(KJ)
    5 CONTINUE
      NRNR=(NR-1)*NA+NR
      IF(A(NRNR)) 17,9,17
   17 DET=DET*A(NRNR)
      PIVOT=1.0/A(NRNR)
      DO 18 J=1,NV
      NRJ=(J-1)*NX+NR
      X(NRJ)=X(NRJ)*PIVOT
      DO 18 K=1,NR1
      I=NR-K
      SUM=0.0
      DO 19 L=I,NR1
      IL=L*NA+I
      LJ=(J-1)*NX+(L+1)
   19 SUM=SUM+A(IL)*X(LJ)
      IJ=(J-1)*NX+I
   18 X(IJ)=X(IJ)-SUM
      RETURN
      END

      SUBROUTINE STEP
      COMMON/CONVG/XX(19,1),RK,LMOST,LSTOP,JWAY,G
      COMMON/MAX/IX,M,ALAM,ALF(19),C,CSQ,DH,WH
      COMMON/MATQ/A(19,19),X(19,1),NR,YSQ,Z
      DIMENSION ALFX(19)
C     COMPUTES NEW ALPHAS AND LAMBDA. NR=IX+1
      IF(JWAY)5,5,9
    5 DO7K=1,NR
    7 ALF(K)=ALF(K)-X(K,1)
      GO TO 11
    9 ALF(NR)=ALAM
      DO10K=1,NR
      XX(K,1)=X(K,1)
      ALFX(K)=ALF(K)
   10 ALF(K)=ALF(K)+X(K,1)
   11 ALAM=ALF(NR)
      DO 20 K=1,IX
      IF(ALF(K).LT.0.0) 19,20
   19 ALF(K)=0.5*ALFX(K)
      XX(K,1)=-1.0*ALF(K)
   20 CONTINUE
      RETURN
      END
```