MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

PREDICTING THE EFFECTS OF OVERLOADS
ON SUSTAINED-LOAD CRACK GROWTH
IN A HIGH-TEMPERATURE SUPERALLOY

THESIS

Robert L. Hastie Jr.
Captain, USAF

AFIT/GA/AA/85D-6

DTIC
ELECTE
FEB 1 2 1986

S

B

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

86 2 12 069

AFIT/GA/AA/85D-6

PREDICTING THE EFFECTS OF OVERLOADS
ON SUSTAINED-LOAD CRACK GROWTH
IN A HIGH-TEMPERATURE SUPERALLOY

THESIS

Robert L. Hastie Jr.
Captain, USAF

AFIT/GA/AA/85D-6

DTIC
ELECTE
S FEB 1 2 1986
D

B

Approved for public release; distribution unlimited

AFIT/GA/AA/85D-6

PREDICTING THE EFFECTS OF OVERLOADS

ON SUSTAINED-LOAD CRACK GROWTH

IN A HIGH-TEMPERATURE SUPERALLOY

THESIS

Presented to the Faculty of the School of Engoneering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Master of Science in Astronautical Engineering

Robert L. Hastie Jr., B.S.

Captain, USAF

December 1985

Preface

The purpose of this study was to support the Engine
Structural Integrity Program and the Retirement-for-Cause
maintenance program. Both programs have been initiated by
the USAF to extend the useful service life of future and
present engine components. Successful implementation of
both programs requires accurate analytical methods for
predicting crack growth. I personally found it rewarding to
develop a analytical technique for predicting sustained-load
crack growth after overloads in engine components.

This study, however, would not have been possible
without all the help and support I received. I wish to
express my sincere thanks and gratitude to Dr. T. Nicholas
AFWAL/MLLN and his department for the use of their
facilities and their time and efforts in assisting me. In
particular I would like to thank Mr. G. Ahrens and Mr. W.
Goddard, UDRI, for their help setting up the experimental
test apparatus. I also thank Mr.G. Hartman and Mr. D.
Johnson for helping me overcome the difficulties learning a
new computer system. In addition, I appreciated Capt K.
Harms explaining his previous work to me. I also greatly
appreciated the overall guidance and support my advisor
Major G. K. Haritos, AFIT/ENY, provided during this study.

I especially thank my wife Victoria who provided
immeasurable support and encouragement when I needed it.

ii

A-1

Table of Contents

iii

List of Figures

List of Tables

vi

Abstract

This study investigates methods of modeling the effects
of overloads on high-temperature sustained-load crack
growth. In addition to a model previously developed for
this specific problem, a computer program developed for
low-temperature, high-frequency cyclic load applications was
evaluated. Sustained-load hold times were converted to
equivalent fatigue cycles to analyze a load spectrum,
consisting of sustained-load with periodic overloads. The
CRACKS crack growth program was used with the Wheeler and
Willenborg models used to account for crack growth
retardation due to overloads.

Predictions were compared with experimental test data
generated on specimens of Inconel 718 at 650 C with periodic
overloads of either 20 or 50 percent. Crack measurements
were made using a electric potential system. The
application of the electric potential system to crack growth
measurement following overloads was extensively evaluated.
It was concluded that the system had to be recalibrated
after each overload due to a sudden advancement in crack
length.

The retardation models were found to require empirical
parameters that depend upon the stress intensity level for

each overload application. Using relationships developed
for these parameters, the CRACKS program using the Wheeler
model was found to be capable of predicting the
time-to-failure for sustained-loading with periodic
overloads within 20 percent of test data. The Willenborg
model was found to be inapplicable to this problem because
it depends solely on stress ratio which has no physical
meaning for sustained-loading. The Wheeler model, on the
other hand, could generally be applied to sustained-load
crack growth using equivalent fatigue cycles. In
conjunction with the CRACKS computer program, this could
provide a powerful new method for evaluating crack growth
under general engine mission spectra including the effects
of overloads.

PREDICTING THE EFFECTS OF OVERLOADS

ON SUSTAINED-LOAD CRACK GROWTH

IN A HIGH-TEMPERATURE SUPERALLOY

I. Introduction

Background

Design trends for modern engines have emphasized
increased performance with higher thrust/weight ratios,
while also requiring improved engine durability and
maintainability. Engine components must therefore be
designed to endure the increasing severity in operating
conditions. Until recently, the method used to predict the
useful service life of engine components was based on
statistical life models. This method was very conservative
and would retire an entire population of engine disks when
it was predicted that, statistically, 1 in 1000 disks would
develop a 0.03-inch fatigue-induced crack [1]. Although this
method helped preclude disk failures, significant useful
life remained in the other 999 disks retired. Estimates
placed this residual useful life at greater than 10 more
lifetimes for 80 percent of the disks retired [2].

-1-

Under the new "Retirement-for-Cause" concept initiated by the USAF, this additional useful life can be utilized by adopting an inspection criterion applied to components after a specific period of time. If the inspected components pass this criterion, they may be returned to service. The criterion is based on fracture mechanics calculations to determine what minimum crack size, if undetected, would grow to failure before the next inspection.

Fracture mechanics is also the basis for predicting crack growth as required by the Engine Structural Integrity Program (ENSIP) specification [3]. The ENSIP specification requires a damage tolerant design approach be applied to structural critical components on all future USAF engine designs. Under this approach, initial flaws or defects are assumed to exist in the components when they enter service. Analysis of components along with verification testing must demonstrate that the initial flaws will not grow to a catastrophic size within the design lifetime of the component.

It is clear that successful implementation of the Retirement-for-Cause program and the ENSIP design approach depends upon technical capability in two key areas. The first is a Nondestructive Evaluation (NDE) procedure used to determine the largest initial flaw size existing in a component after an inspection. This initial flaw size is then assumed to exist in all components. The second is the

-2-

capability to use analytical prediction models to accurately
predict the crack propagation from the initial crack size to
failure.

There are numerous crack growth rate prediction models
with varying degrees of complexity. Most of these models
have been developed for application to airframe components
under typical airframe spectra including large numbers of
cycles with periodic overloads. For engine applications,
the spectra are simpler, involving fewer cycles and only
occasional overloads. No complex interaction models have
been developed for engine spectrum loading which involves
both cyclic and sustained-loading.

Simple crack growth models usually calculate the growth
cycle-by-cycle by integrating a crack growth rate equation.
More complex models used in airframe analysis include
retardation routines to account for the decrease in growth
rate following a peak overload cycle. These models do not
address crack growth under sustained (hold-time) loading
which is present in a typical engine spectrum. CRACKS [4]
which represents the state-of-the-art in airframe spectrum
crack growth analysis is a complex prediction program used
to analytically calculate crack growth under large spectrums
of cyclic loading. This program includes the Wheeler and
Willenborg models for predicting retardation effects.

Objective

This thesis explores one aspect of the complex crack-growth-rate prediction problem. Specifically, this thesis explores the applicability of existing crack-growth retardation models, developed for high-frequency, low-temperature airframe applications, to high-temperature sustained-load crack growth retardation. Procedures will be developed to convert sustained-load time to equivalent fatigue cycles so that classical Wheeler and Willenborg retardation models can be used in the CRACKS computer program to predict sustained-load crack growth rates following overloads. The retardation models will be applied to data obtained from previous experimental work as well as new experimental proof tests to verify each model's capability.

II. Retardation Model Theory

In this study three plastic zone retardation models
were examined. First was the Wheeler model [5] which
reduces the crack growth rate da/dn to account for
retardation. Second was the Willenborg model [6] which
accounts for retardation by reducing the maximum and minimum
stress intensity factors. Also, the minimum stress
intensity factor, if negative, is truncated at zero.
Finally was the Overload model developed by K. Harms, T.
Weerasooriya, and T. Nicholas [7] [8] which accounts for
retardation by reducing the stress intensity factor K, to a
lower effective value K_{eff}, after each overload. The theory
behind each of these models is discussed in the following
sections.

Wheeler Model

The basis for the Wheeler, Willenborg, and Overload
models is that an overload cycle produces an extended
plastic zone that retards crack growth. Thus, in figure 1,
r_p represents the plastic zone due to sustained-loading and
$\bar{r}_p$ represents the plastic zone due to an applied overload.
As long as a_1 is less than a_2, the sustained-load crack is
growing in the overload plastic zone at a slower or retarded
rate. The Wheeler model predicts this retardation effect by
reducing the crack growth rate while growing through the

-5-

overload plastic zone. The crack growth rate da/dn is reduced by multiplication with a retardation parameter C_p. This parameter depends on the ratio of the current sustained-load plastic zone size to the previous overload plastic zone size raised to a shaping exponent m. If the sustained-load plastic zone grows past the prior overload plastic zone, no retardation is predicted. In terms of the symbols used in figure 1, the retardation parameter is defined as,

$$C_p = \left(\frac{r_p}{a_2 - a_1 + r_p} \right)^m \qquad \text{for } a_1 < a_2 \qquad (1)$$

and

$$C_p = 1 \qquad \text{for } a_1 \geq a_2 \qquad (2)$$

where r_p = extent of current yield zone

$a_2 - a_1 + r_p$ = distance from crack tip to elastic plastic interface

m = shaping exponent

The shaping exponent m is used to calibrate the retardation model with experimental data. Generally, m has been found to be a material dependent constant.

Once the shaping exponent is defined for the given material, the value of C_p is substituted into equation 3.

$$\frac{da}{dn} \text{(retarded)} = C_p \frac{da}{dn} \text{(non-retarded)} \qquad (3)$$

-6-
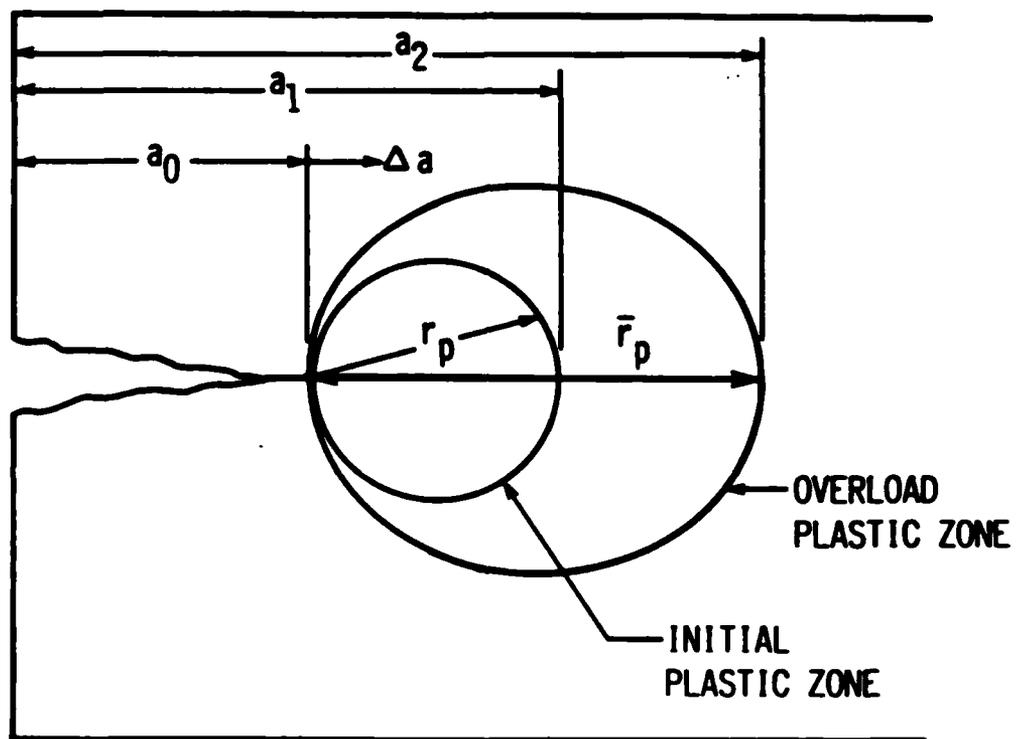
Figure 1 Schematic diagram of the plastic zones at the crack tip.

The Wheeler model accounts for retardation by substituting da/dn (retarded) for da/dn (non-retarded) while the crack is growing through the overload plastic zone. The numerical procedures used to implement the Wheeler model are contained in the CRACKS program.

Willenborg Model

The Willenborg model like the Wheeler model is based on yield zone analyses. The retardation effect is accounted for by a reduction in stress intensity factor. The reduction factor is calculated by first finding an equivalent stress, σ_{ap}, that will produce a plastic zone from the current crack tip location to the edge of the overload plastic zone. Refering to figure 1, this corresponds to a plastic zone of radius of $\overline{r}_p - \Delta a$. Second, a stress reduction factor, σ_{red}, is calculated by subtracting the currently applied maximum stress σ_{max} from σ_{ap}.

$$\sigma_{red} = \sigma_{ap} - \sigma_{max} \qquad (4)$$

When the crack growth amount Δa plus the current plastic zone size r_y equals the overload plastic zone size $\overline{r}_y$, the value of σ_{red} is set equal to zero, since the crack propagation is no longer retarded. Third, effective values of the currently applied stresses are calculated by:

-8-

$$[\sigma_{max}]_{eff} = \sigma_{max} - \sigma_{red} \tag{5}$$

$$[\sigma_{min}]_{eff} = \sigma_{min} - \sigma_{red} \tag{6}$$

If either effective stress is less than zero, it is set
equal to zero. Finally, using the effective maximum and
minimum stress values, effective stress intensity factors
are calculated. This produces the following retardation
relationships.

$$[K_{max}, K_{min}]_{eff} = [K_{max} - K_{red} , K_{min} - K_{red}] \tag{7}$$

where $K_{red} = K_{ap} - K_{max}$ for $a_1 < a_2$

      $K_{red} = 0$ No retardation for $a_1 \geq a_2$

After each crack growth increment a new value of σ_{ap} is
calculated. The corresponding new value of K_{red}, calculated
using equation (4), is substituted into equation (7),
yielding the new K_{eff} values for the next growth increment.

Overload Model

    This model uses a linear cumulative damage concept to
sum the growth contributions of a single overload fatigue
cycle and growth due to sustained load. The basis for the
Overload model, like the Wheeler and Willenborg models, is
that an overload cycle produces an extended plastic zone
that retards crack growth. The plastic zone concept is
illustrated schematically in figure 1. The sustained load
and overload fatigue cycle have stress intensity factors

-9-

K_s and K_m and plastic zone radii denoted by r_p and $\overline{r}_p$ respectively. The distances from the center of the crack to the edge of the plastic zones due to sustained loading and a fatigue overload cycle applied when the crack length was a_o are labeled a_1 and a_2. For a crack advancement $\triangle a$ from a_o, the crack tip will be in an overload plastic zone until $\triangle a = a_2 - a_1$. While in this plastic zone the growth rate will be retarded. The Overload model uses a reduced value of stress intensity factor, K_{eff}, to account for the retarded growth rate. For modeling purposes, K_{eff} is taken in the form:

$$K_{eff} = K_s [1 - \alpha \exp (-\beta \triangle a)] \qquad (8)$$

where K_{eff} = effective (reduced) stress intensity factor

K_s = sustained-load stress intensity factor

α, β = modeling parameters

$\triangle a$ = incremental crack extension

The parameter β is chosen such that steady-state crack growth will resume after the crack has traversed the overload plastic zone. Mathematically, it is desired to have K_{eff} approach K_s when $\triangle a$ approaches $(\overline{r}_p - r_p)$. This is accomplished by letting

$$\beta \triangle a = \pi \sqrt{2}. \qquad (9)$$

When $\triangle a = \overline{r}_p - r_p$, the resulting value of K_{eff}/K approaches

-10-

unity to within one percent as shown in figure 2. The plane
stress plastic zone sizes for the overload cycle and
sustained load are given by,

$$\overline{r}_p = [K_m / \sigma_y]^2 / \pi$$
$$r_p = [K_s / \sigma_y]^2 / \pi \qquad (10)$$

where σ_y is the uniaxial tension yield stress of the
material. Substituting $\Delta a = \overline{r}_p - r_p$ and equation (10) into
equation (9) yields an expression for β:

$$\beta = \frac{\sqrt{2} \ \pi^2 \ \sigma_y^2}{K_s^2 (\tau^2 - 1)} \qquad (11)$$

where the overload ratio τ is defined by:

$$\tau = \frac{K_m}{K_s} \qquad (12)$$

Observing equation (11) it is noted β is only a function
of material properties and test conditions. β therefore
cannot be used as an adjustable parameter to fit
experimental data. This leaves the parameter α given in
equation (8), to be adjusted to fit experimental data. The
value of α chosen determines K_{eff} immediately after an
overload application and therefore can be used to model the
reduced value of sustained-load crack growth rate. The
effect of varying α on the ratio of K_{eff}/K_s is seen in
figure 2. Also this figure demonstrates how the Overload

-11-
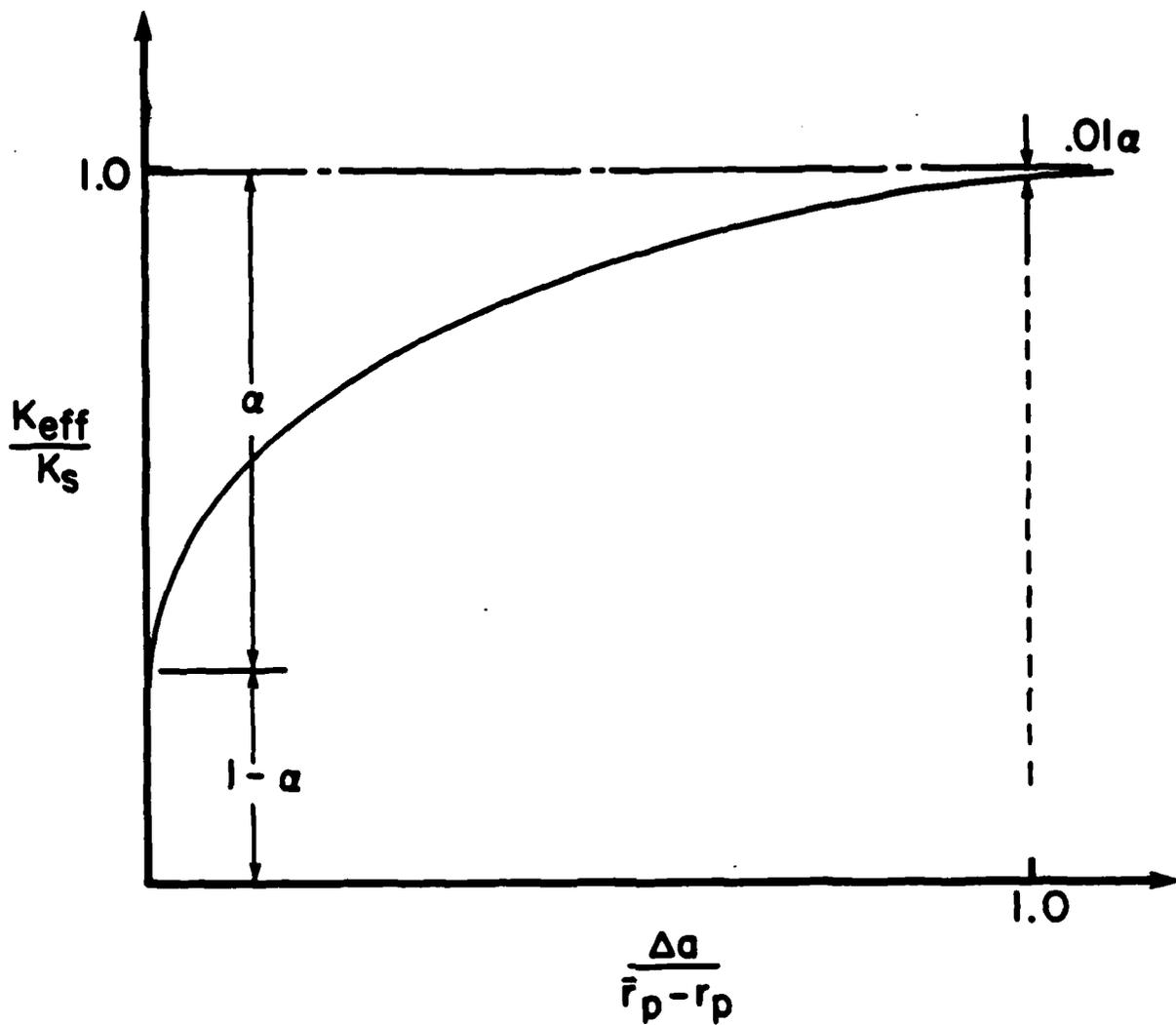
Figure 2 Variation of K_{eff}/K_s as a function of
normalized crack advancement from the
crack tip in the overload plastic zone.

model uses a decreasing exponential to asymptotically
approach the normal crack growth.

In a prior investigation, Harms made experimental
measurements of the delay time before normal sustained-load
crack growth resumed after an overload was applied. Harms
[7] noted that delay times and, hence, the value of K_{eff}
changed systematically with K. Plots of the delay times Δt_r
versus K level for the 20% and 50% overload cases tested by
Harms are shown in figures 3 and 4. The boundaries of the
data were used to approximate the average delay time
(Δt_{avg}) curve. Using an iteration process, values of α
were chosen with a specified value of K to generate a Δt_r
for the model. Adjustments in α were made until Δt_r
calculated and Δt_{avg} agreed resonably well. Harms repeated
this process over the full range of K values to define a
functional relationship for Δt_r, K, and α.

The functionals developed were expressed in
non-dimensional terms as α/α^* and K/K^* where α^* and K^* are
threshold values. K^*, a material property, was substituted
into equation (1) for Keff with $\Delta a = 0$ and $\alpha = \alpha^*$. Solving
for α^* yields $\alpha^* = 1 - (K^* / K_s)$ with the limiting values
being:

$$\frac{\alpha}{\alpha^*} = 1 \quad \text{total crack arrest} \qquad (13)$$

$$\frac{\alpha}{\alpha^*} = 0 \quad \text{no retardation.} \qquad (14)$$

-13-
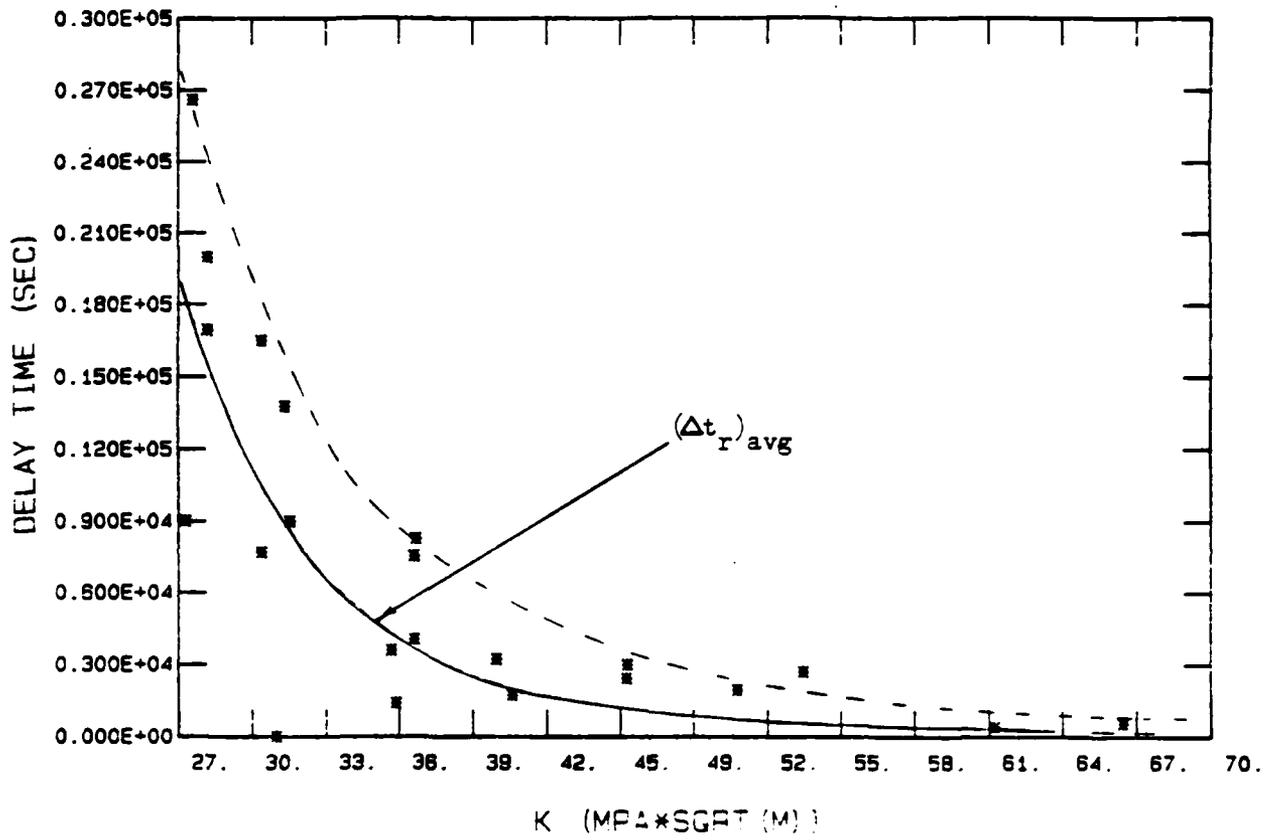
Figure 3 Delay Times Resulting from 20 Percent Overloads at
Various Stress Intensities.

-14-
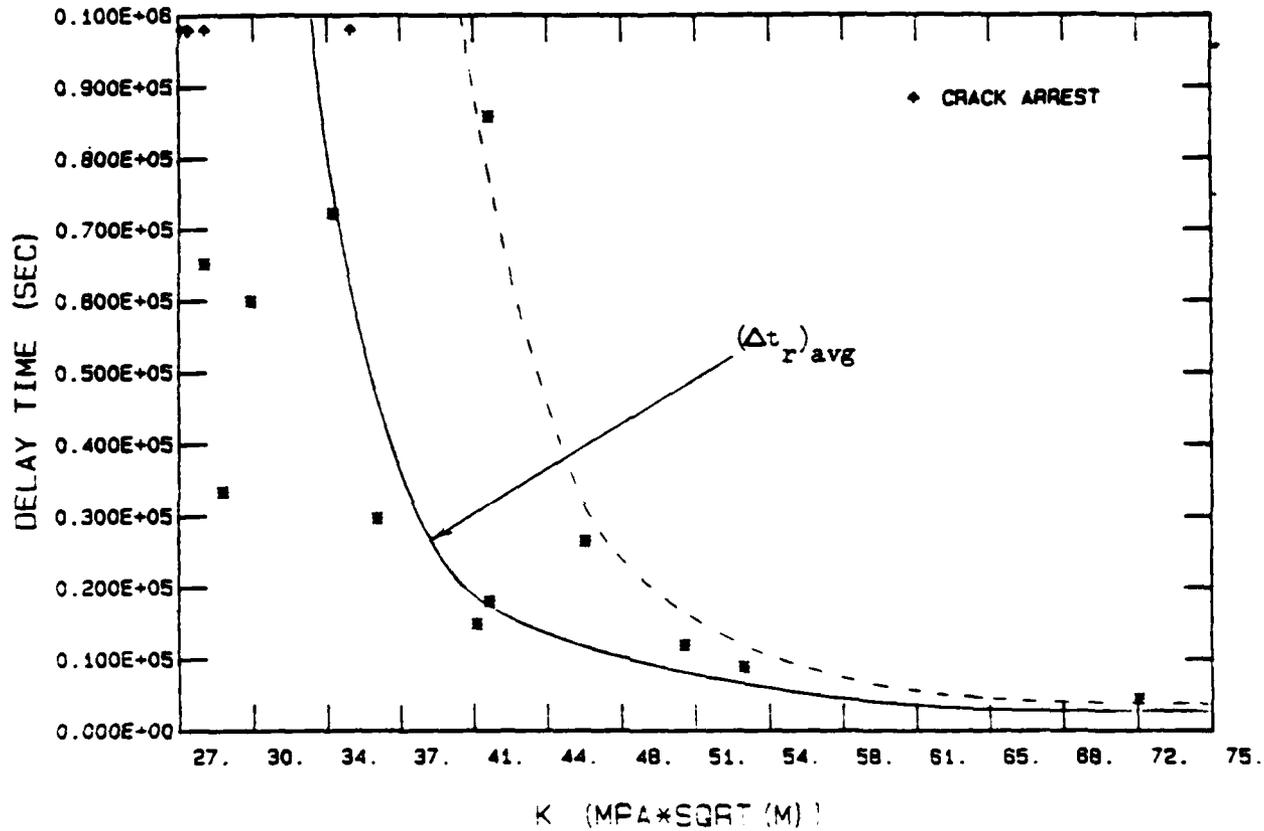
Figure 4 Delay Times Resulting from 50 Percent Overloads
at Various Stress Intensities.

-15-

The α versus K curves generated by Harms for 20% and 50% overloads, normalized to the threshold values, are shown in figure 5. The curves were fit with a polynomial equation. The equation for the 20% overload case is

$$\alpha = \alpha^* \ [(-.0730791E-01) \ (K/K^*)^3 + (0.303086) \ (K/K^*)^2$$
$$-(0.422108) \ (K/K^*) + (0.117517E-01) \ ] \qquad (15)$$

while that for the 50% overload case is

$$\alpha = \alpha^* \ [(-.121127E-01) \ (K/K^*)^2 + (0.239231E-01) \ (K/K^*)$$
$$+ (0.987133) \ ]. \qquad (16)$$

With α and β defined, the Overload model's retardation effect was calculated using equation (8). The K_{eff} value obtained from this equation is used to account retardation while within the overload plastic zone.

It is apparent all three of the retardation models discussed are numerically cumbersome. Therefore, computer programs were developed to implement the retardation calculations.

-16-
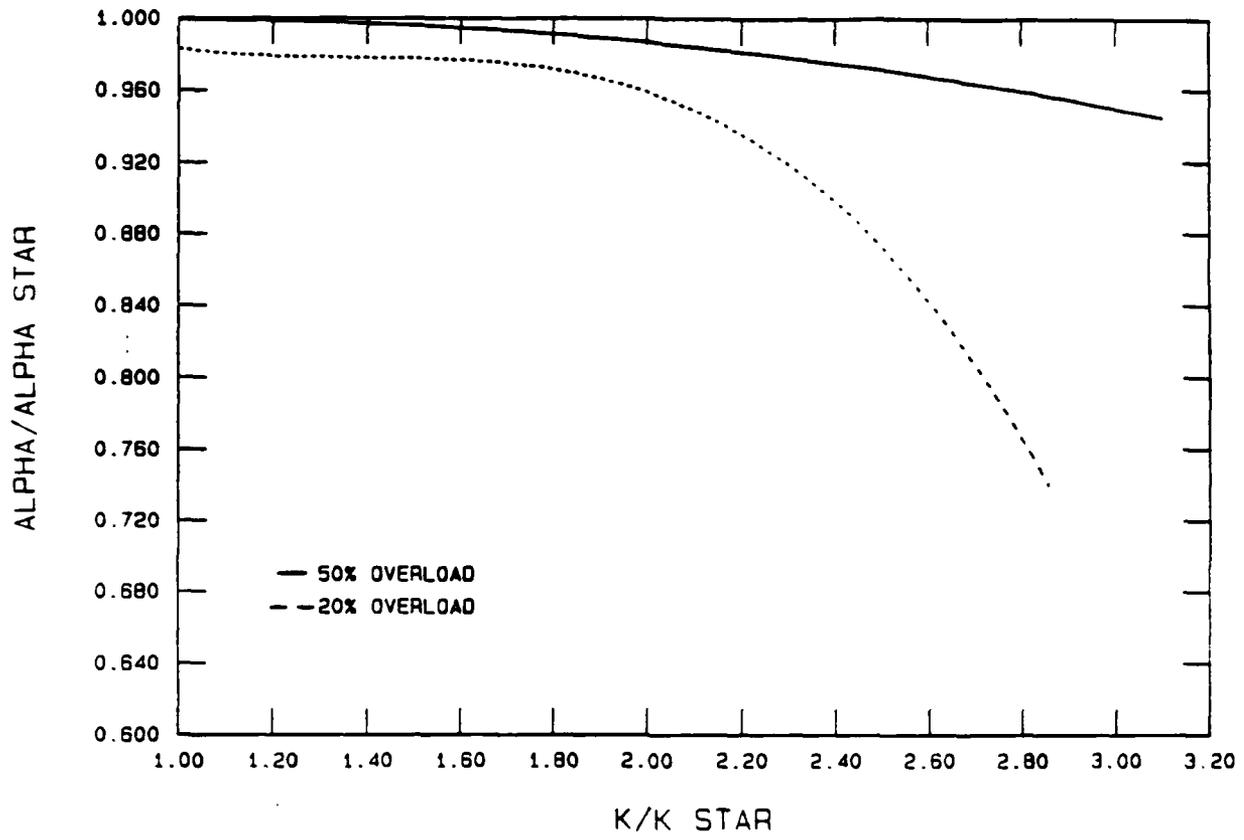
Figure 5 α Functions for 20 and 50 Percent Overloads.

-17-

III. Computer Model Development

In this study two computer programs were used to perform the numerical calculations required to predict the crack growth. First, a computer program called "Overload" was written to integrate a creep crack growth rate (da/dt) equation and incorporate the Overload retardation model. The second program used was the CRACKS program which integrates a fatigue crack growth (da/dn) equation. To use this program for sustained-load crack growth required converting sustained load time into equivalent fatigue cycles. The crack growth due to a simple spectrum of sustained-load (represented by equivalent fatigue cycles) with periodic overloads was analyzed using the CRACKS program. Both the Wheeler and Willenborg retardation schemes were used in CRACKS. After completing the growth predictions, equivalent cycles were converted back to sustained load time. The theory and assumptions associated with each model will be discussed in detail in the following sections.

Overload Program

This program was written to carry out the numerical calculations required to use the Overload retardation model. A listing of the program is contained in appendix 3. The input data for this program are the initial crack size,

load history, sustained-load crack growth rate, and compact
tension specimen dimensions. At anytime, for a given crack
length and sustained-load amplitude, the stress intensity
factor is calculated using the compact tension solution [9]
below:

$$K = \frac{P}{b \sqrt{w}} \frac{(2+a/w) \; f(a/w)}{(1-a/w)^{3/2}} \qquad (17)$$

where   K = Stress Intensity factor

$f(a/w)=(0.886+4.64(a/w)-13.32(a/w)^2+14.72(a/w)^3+5.6(a/w)^4)$

   a = Crack length

   P = Applied load

   b = Specimen thickness

   w = Specimen width

Once K is known, the crack growth rate is determined
from the crack growth rate equation relating da/dt to K. A
Modified Sigmoidal Equation (MSE) is used to represent this
relationship. The MSE model, developed by General Electric
[10], uses the various coefficients in the following
equation to fit a sigmoidal curve through the data.

$$da/dt = [\exp(B)] \; [K/K^*]^P \; [\ln(K/K^*)]^Q \; [\ln(K_c/K)]^D \qquad (18)$$

where   da/dt = Crack growth rate

       K = Current stress intensity value

       K^* = Threshold stress intensity value

       K_c = Critical Stress intensity value

   B,P,Q,D = Fitting parameters of the curve

-19-

In this equation, K^* and K_c are in units of $(MPa\ m^{1/2})$ and da/dt is calculated in units of (m/sec). The remaining constants are non-dimensional. Harms [3] determined the sigmoidal coefficients by fitting the data from his constant sustained-load baseline test. The test data, along with the best-fit MSE coefficients, are shown in figure 6.

With the relationship between a, K and da/dt known, it is possible to find the time it takes to grow from an initial crack size of a_i to final crack length a_f using:

$$\Delta t = \int_{a_i}^{a_f} \frac{da}{da/dt_{MSE}} \qquad (19)$$

This integration is carried out numerically by dividing the region of crack growth $(a_f - a_i)$ into a finite number of increments Δa and summing the time to grow each increment. For each small increment, Δa, the value of K is calculated for the end points and for any intermediate crack lengths from equation (17). Similarly, da/dt is determined for the end points and for discrete points in between using equation (18). The numerical integration of each increment Δa is performed via Simpson's rule [11]. The total time to grow from a_i to a_f is obtained by summing the time to grow each Δa interval. Since Simpson's is rule only a numerical approximation of the exact integral of the function, care was taken to ensure that proper accuracy was carried through the calculations.
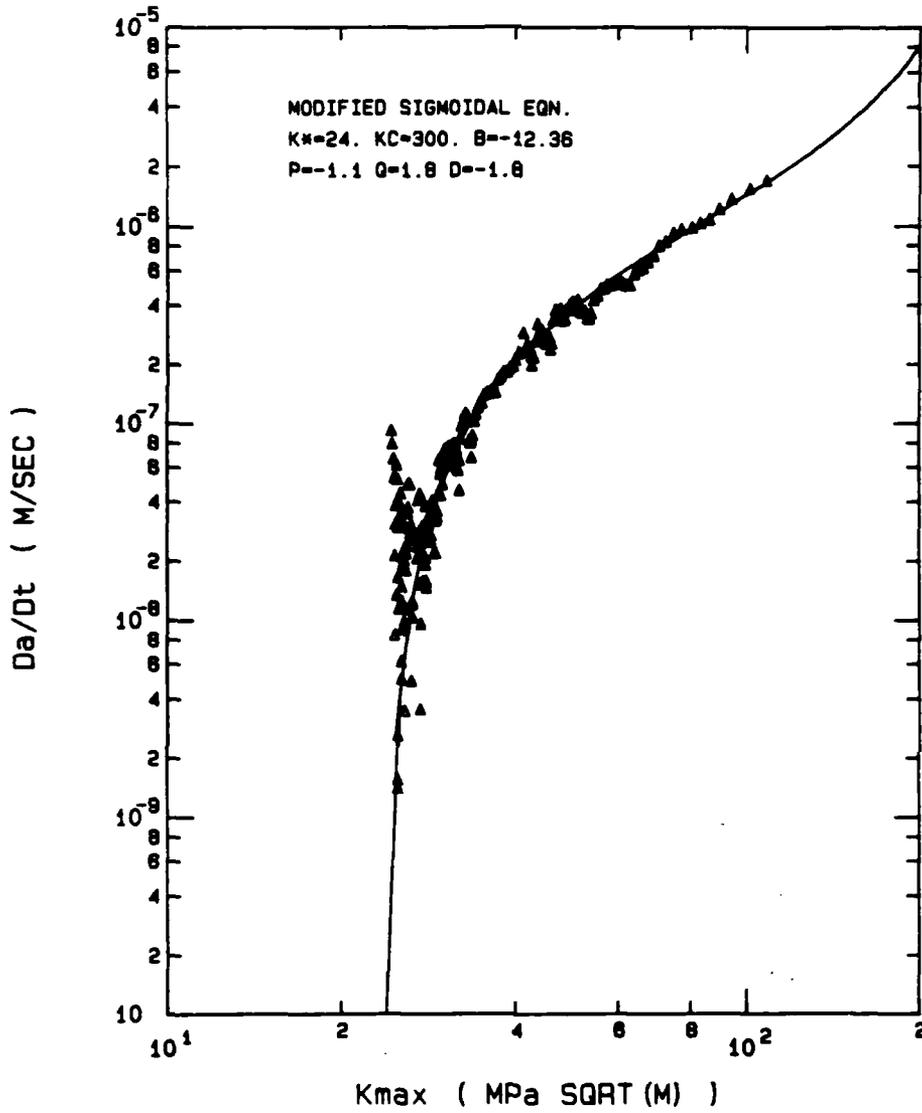
Figure 6 MSE Model Fitted to Sustained-Load Baseline Data.

The accuracy of the integration was controlled in two ways. First, the interval of total crack length $(a_f - a_i)$ was divided into a number of increments (Δa). The size of each increment was based on the initial stress intensity level at the beginning of the interval and was determined as follows. Values of $(da/dt)_{MSE}$ were calculated at various K-levels. These values were multiplied by 100 seconds to obtain the crack length interval (Δa_{100}) which produces 100 seconds of sustained load growth. Linear functions were used to represent Δa_{100} as a function of K, in a manner that insured the growth time interval was always less than 100 seconds. The effect of this first step was to use small Δa intervals at lower K values and larger Δa intervals as K increased, thus limiting the total time required for the crack to grow through any interval to less than 100 seconds, for any value of crack length or K.

Each Δa interval was then numerically integrated using Simpson's rule. The numerical integration errors were minimized by testing the Simpson's integration subroutine for convergence. This was accomplished by integrating each crack length interval at least twice. The program started by using two subintervals within each Δa interval. This interval was then integrated again doubling the number of subintervals. The difference of the values of the time calculated to grow through the interval using these two integrations was compared with a convergence value; this

-22-

value was calculated by multiplying a tolerance, set at 0.0001, by the time calculated to grow through the crack interval. Since this time increment was always near 100 seconds, the convergence value corresponded to approximately 0.01 seconds difference between the calculated times using the two successive integrations. If the difference exceeded 0.01 seconds, another integration was performed, with the number of subintervals being doubled again. This procedure was repeated until the convergence criterion was satisfied. Assuming that each $\triangle$a interval resulted in an error of 0.01 seconds, the cumulative error for the total time predictions is no greater than 10 seconds based on a maximum number of 1000 intervals corresponding to a total test time of approximately 10^5 seconds.

The numerical integration scheme was applied to a loading history which involved a constant sustained-load with periodic overloads. The input load history for each specimen specified the times at which overloads were applied and the percentage of overload. An example of how the load history was entered into the program is shown in figure 7.

The program integrates repeated $\triangle$a intervals, equal to approximately 100 seconds of sustained load time, until the total time exceeded the time when the next overload was applied. At this point, the time when the last $\triangle$a interval started and the time it took to grow the last $\triangle$a interval were known. Using a linear interpolation, the length of the

-23-

last $\triangle$a interval was reduced so that the total growth time
equaled the time when the overload was applied. At this
point, the retardation effect due to the overload was added
in the model. This was accomplished by calculating a
reduced stress intensity factor, K_{eff}, defined by equation
(8). In this equation, the modeling parameter β was related
to the overload plastic zone size, while the parameter α was
defined by fitting experimental data. During the retarded
growth K_{eff} was substituted for K in equation (18) until the
crack grew through the overload plastic zone.

Time	% Overload
0	0 %
T1	20 %
T2	20 %
T3	20 %
etc.	
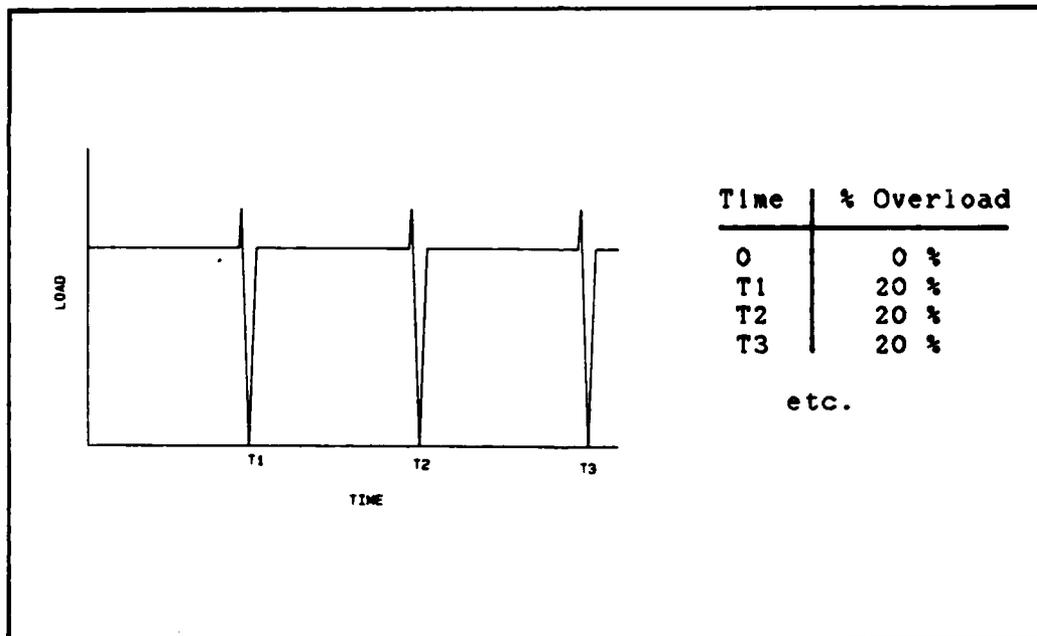
Figure 7 Overload Program Load History Input Example.

Harms [7] noted that each time an overload cycle was
applied an apparent jump in crack length occurred. This
same jump phenomenon was also noted by Larsen and Nicholas
[12] in their study of crack-growth transients at elevated
temperature. The amount of crack jump seemed to correlate
with the K level at overload application. However, Harms's
data for jump versus K level, shown in figure 8, contained a
large amount of scatter making it difficult to fit a
functional relationship. Harms therefore assumed a constant
value of 0.381 mm jump in crack length at each overload. In
an attempt to improve the estimation of the jump function,
several other functional relationships were investigated.

First, a function relating the amount of jump to the K
level at overload application, normalized to the threshold
K^* value of 24 MPa $m^{1/2}$, was tried. This function takes the
form of

$$Jump = 0.2 * (K/K^*) \quad mm \qquad (20)$$

and is labeled curve 1 in figure 8. The second function used
the Log of the K level at overload application, normalized
to the threshold K^* value of 24 MPa $m^{1/2}$. The resulting
function takes the form of

$$Jump = 1.25 * Log(K/K^*) \quad mm \qquad (21)$$

and is labeled curve 2 in figure 8. Finally, a linear
function with an initial jump of 0.381 mm at threshold

-25-

Increasing to .508 mm jump near the critical stress
intensity was tried. This function is labeled curve 3 in
figure 8. Also shown in this figure is the constant 0.381 mm
jump used by Harms, labeled curve 4.

The proof test, conducted by Harms, was analyzed using
each jump function to predict the increment in crack length
caused by the overload cycle. The resulting predictions are
shown in figure 9. Curves 1 and 2 correspond to jump
functions in equations 20 and 21, respectively these
functions underestimated the jump at low K values. The
resulting predictions had significant delay times. Curve 3
corresponded to the linearly increasing jump function, and
was the most exact prediction of the total time to failure
for this test. The constant jump of 0.381 mm, labeled curve
4, predicted the total time to failure to within 4 percent.
Since the constant jump used by Harms gave predictions
within normal test scatter, it was decided to use a constant
jump of 0.381mm for all further calculations. This also
allowed a direct comparison of Harms's work with the other
retardation models.
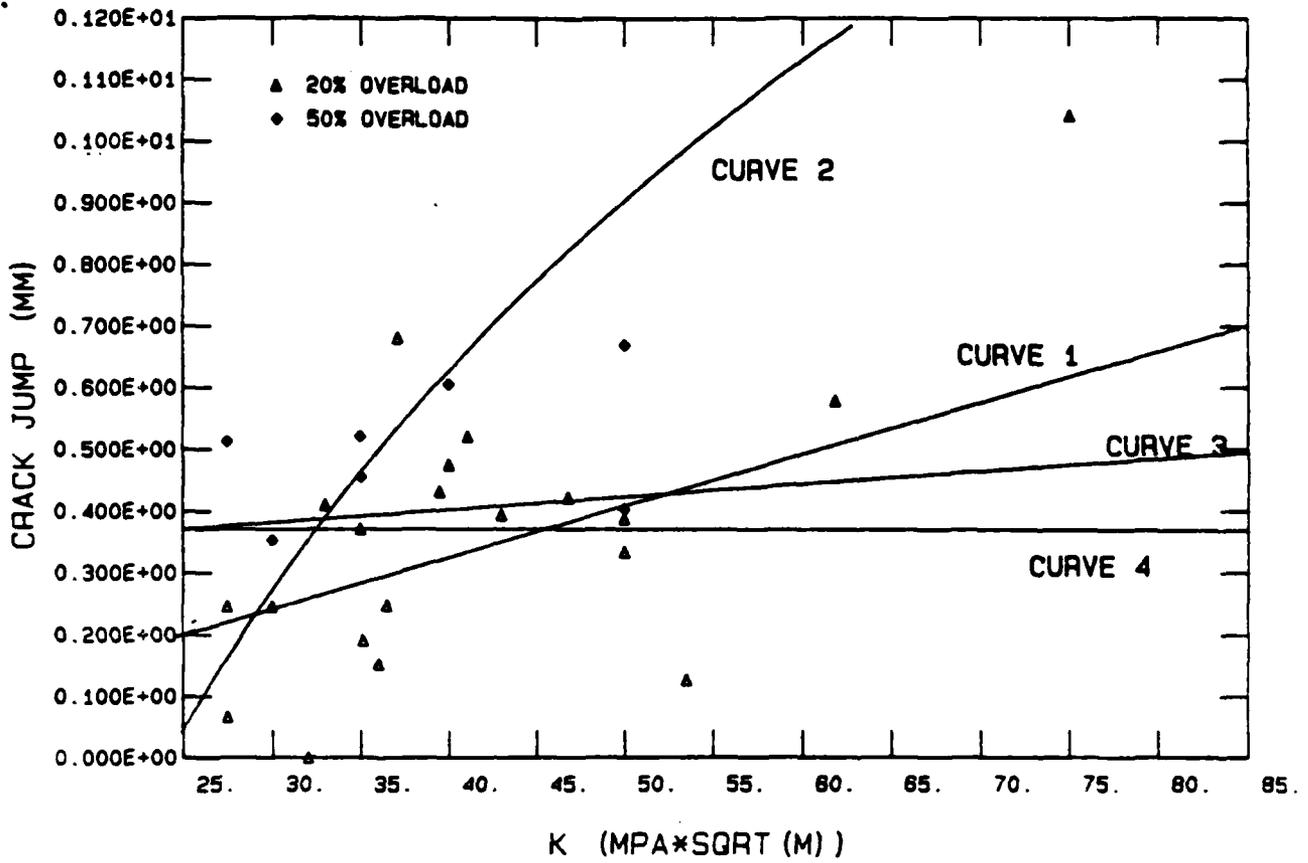
Figure 8 Crack Jumps from Overload Cycles at
Various K Levels.
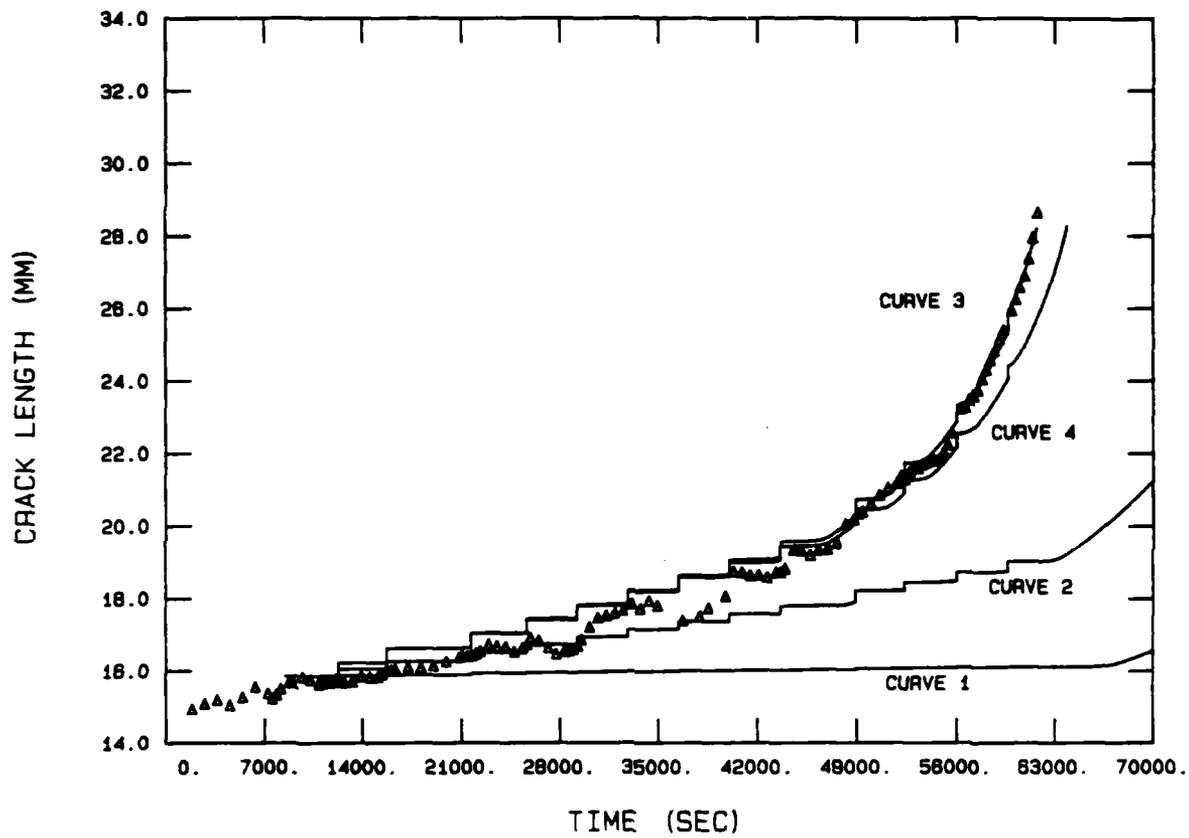
Figure 9 Overload Model Predictions to Proof Test
84-507 Using Various Jump Functions.

CRACKS Program

This program was developed by R. M. Engle [4] to predict crack growth in airframe applications where high frequency, low temperature spectrum loading occurs. The objective here was to modify the original program so that it could be used to analyze high temperature sustained loading with periodic overloads. A technique was developed for converting sustained loading to equivalent fatigue cycles. This technique and all required programing changes made within CRACKS are described next.

The process of converting sustained load creep crack growth to equivalent fatigue cycles was achieved by setting da/dn equal to da/dt at equivalent ΔK and K_{max} stress intensity levels. A one-to-one correspondence would equate one fatigue cycle to one second of sustained load. Since CRACKS can only use one crack growth rate equation, a method of representing both the rate of growth due to sustained loading and overload fatigue cycles with the same equation was needed. The growth rate due to an overload cycle was approximated using previously generated test data [13] for Inconel 718 at 650 C, with an R ratio of 0.1 and frequency of 0.01 Hz. This frequency is approximately that of the single overload cycle in the experimental part of the investigation. This data is labeled 83266G test data and shown in figure 10. Also shown in the figure is the baseline
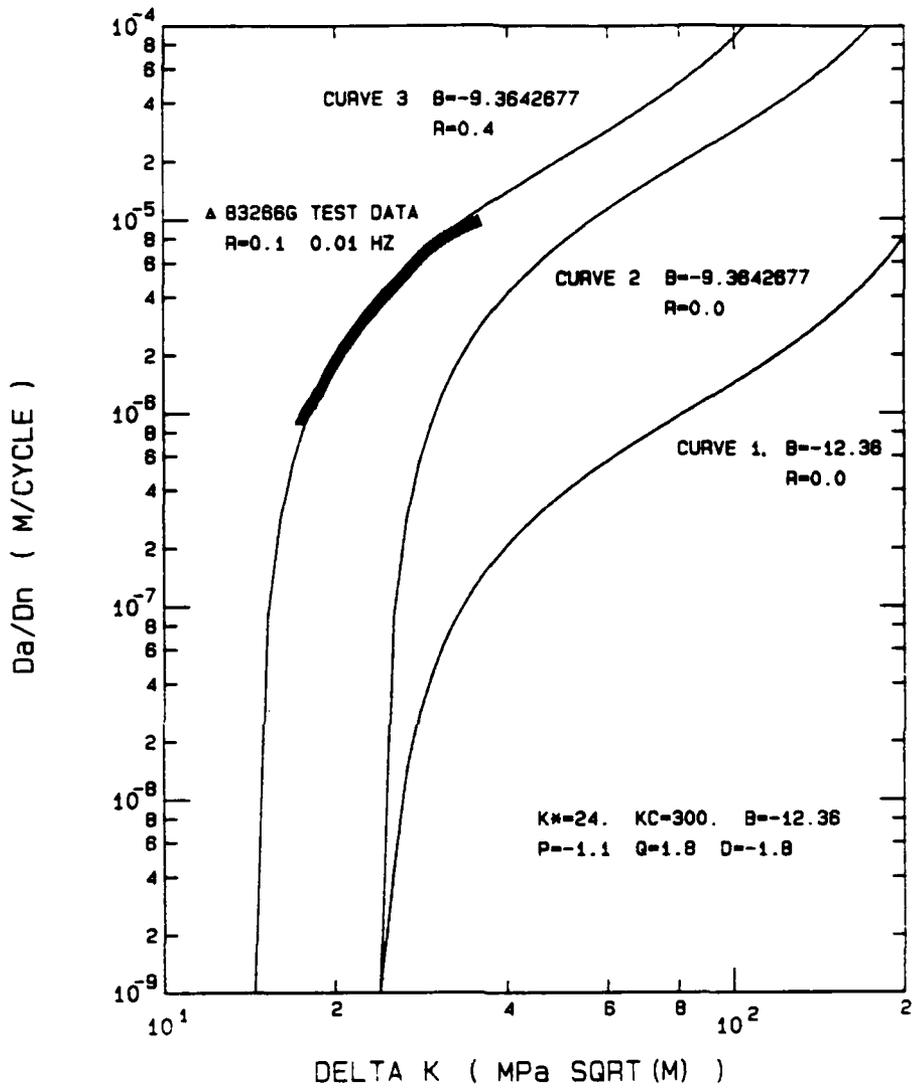
Figure 10 MSE Curve Shift Using the B Parameter and R Ratio.

sustained-load sigmoidal curve, developed by Harms, and
labeled curve 1. The sustained-load growth rate da/dt
(m/sec) given by curve 1 equals the cyclic growth rate da/dn
(m/cycle). Thus, one second of sustained-load is equal to
one fatigue cycle. In order to have the same curve
represent both the sustained loading and overload cycles a
vertical and horizontal shift of curve 1 was needed.
Refering to the sustained load sigmoidal equation below

$$da/dt = [\exp(B)] * [K/K^*]^P * [\ln(K/K^*)]^Q * [\ln(K_c/K)]^D \quad (22)$$

the parameter B can be used to absorb a rate multiplication
constant into the equation. Using a trial and error
procedure, a factor of 20 was found to vertically shift the
curve to a position where an additional horizontal shift
placed the curve on the crack growth rate data for overload
cycles. The new B value was found by solving the following
equation for B

$$\exp B = 20 \exp B_{old} \quad (23)$$

The new value for B and the associated new sigmoidal curve
is labeled curve 2 in figure 10. The cyclic growth da/dn
(m/cycle) now equals 20 times the sustained-load growth rate
da/dt (m/sec). Thus, 20 seconds of sustained-load is equal
to one fatigue cycle. The horizontal shift was accomplished
using an R ratio shift. The cyclic sigmoidal equation shown
below

-31-

$$da/dn=[\exp(B)]\ast[\triangle K/\triangle K^\ast]^P\ast[\ln(\triangle K/\triangle K^\ast)]^Q\ast[\ln(\triangle K_c/\triangle K)]^D \quad (24)$$

can be modified for R ratio effects by replacing $\triangle K$ with $\triangle K\ast(1-R)$, $\triangle K^\ast$ with $\triangle K^\ast\ast(1-R)$, and $\triangle K_c$ with $\triangle K_c\ast(1-R)$. Again, using trial and error, it was found that an R ratio of 0.4 produced the desired horizontal shift. The resulting sigmoidal equation with fixed a value for $\triangle K^\ast = \triangle K^\ast\ast(1-0.4)$ and $\triangle K_c = \triangle K_c\ast(1-0.4)$ is labeled curve 3 in figure 10. Both the sustained load and overload cycle can be modeled by varying the R ratio to represent the different cycles. The sustained loading is modeled using 1 cycle equals 20 seconds of sustained loading at an R ratio of 0.4. The overload cycle is modeled as 1 cycle equals 1 overload cycle at an R ratio of 0.0. A graphical representation of these cycles is shown in figure 11. Since $\triangle K = \triangle K\ast(1-R)$ and R = 0.4 for the sustained loading, it is easily seen that the R ratio dependence cancels out of equation (24) for sustained loading. The sustained load crack growth rate then in essence is defined by curve 2 in figure 10. Remembering that the threshold and critical stress intensity factors are fixed at values corresponding to an R ratio of 0.4 and the overload cycle is modeled with an R ratio of 0.0, the overload cycle crack growth rate is calculated using curve 3 in figure 10 with $\triangle K = \triangle K\ast(1-0.0)$ or just $\triangle K$.

-32-
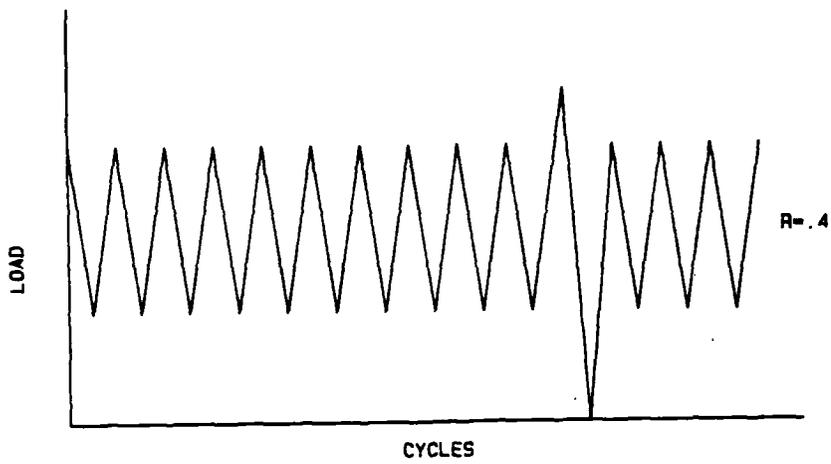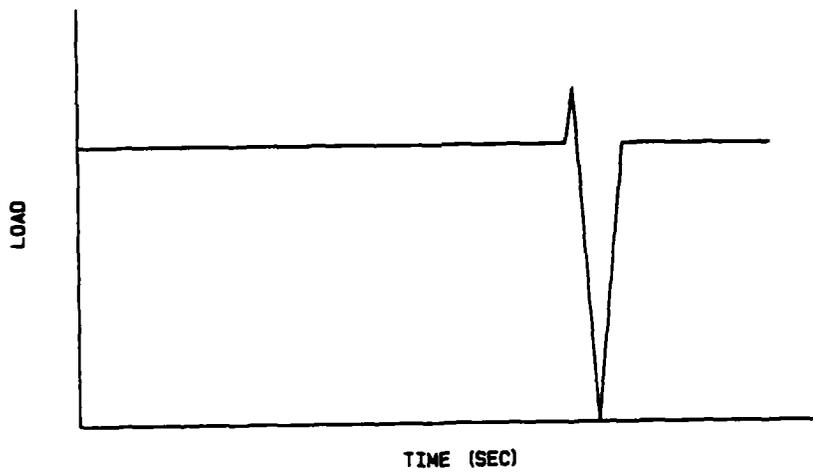
Figure 11 Graphical Representation of Equivalent
Sustained-Load Fatigue Cycles.

Implementation of the method developed to change sustained loading into equivalent fatigue cycles required several changes to the CRACKS program. The source listing of the program used in this study is contained in appendix 3. Before describing the detailed changes within the program, a brief description of the overall program will be presented.

The CRACKS program consists of twenty two routines of which sixteen are basic to crack growth calculations and six others are used to implement the retardation models. Detailed descriptions of each routine is contained in the CRACKS manual [4]. The overall supervisory routine CRACKS4 calls each of the other subroutines as needed during the calculations. Only the major changes made within each subroutine will be described, although each change usually required changing other subroutines that used the same common blocks.

The first change was to include the sigmoidal crack growth rate equation in the RATE subroutine. This required changing the input and output subroutines to read and write the sigmoidal coefficients. Also, equation (24) was programed in the RATE subroutine for use whenever the crack growth rate was required by the main CRACKS4 routine.

The second change was to include the ASTM compact tension stress intensity solution, given by equation (17), in the BETA subroutine. The CRACKS program calculates all K

-34-

values using the equation

$$K = \sigma Y \sqrt{\pi a} \qquad (25)$$

where   K = Stress intensity level

        σ = Stress or load

        Y = Variable defining case solution

        a = Current crack length

The variable Y is used to define the case solution for the type geometry being analyzed. Y was set equal to equation (17) divided by $\sigma \sqrt{\pi a}$. Whenever the stress intensity value for a given crack length was needed, the subroutine K was called. This subroutine called the BETA subroutine where a new value for Y was calculated for substitution into equation (25).

The final change made in CRACKS affected the numerical integration method used in the program. CRACKS was written to handle a very large number of fatigue cycles. Crack growth due to large spectra of cycles are calculated using a linear approximation technique in order to save computation time. The basis for the approximation is the assumption that the damage parameters remain constant over some small increment of crack growth $\triangle$a. Engle [14] found the linear approximation in CRACKS to be an excellent balance between accuracy and computational efficiency for very large spectra. However, when smaller constant amplitude spectra were analyzed, the program was more efficient using a

-35-

cycle-by-cycle Runge-Kutta numerical integration method
[14]. During the check-out phase of the CRACKS program, it
was found that the linear approximation did not provide the
same accuracy as the Overload program for constant sustained
loading. Therefore, the program was changed to eliminate
the linear approximation and use the Runge-Kutta
cycle-by-cycle integration method. With this change in
place both the Overload and CRACKS programs predicted
exactly the same results for constant sustained-load growth
containing no overloads.

Results from the CRACKS program were obtained in terms
of equivalent fatigue cycles of growth. This required a
separate Fortran program to convert cycles back into
sustained load time by multiplying the equivalent cycles by
20 seconds per cycle.

Both the Overload and CRACKS programs were now capable
of predicting sustained load crack growth with periodic
overloads. The retardation effect overload cycles produced
was estimated using the Overload model in the Overload
program and the Wheeler and Willenborg models in the CRACKS
program. The next step in developing the programs was to
compare the analytical predictions of each model to
experimental test data.

IV. Application of Retardation Models

Each retardation model was applied to typical test
segments using the computer programs described in section
III to predict how sustained load crack growth was affected
by overload cycles. Six segments of crack growth data with
average delay times were selected from Harms's work for
analysis. The segments included both 20 % and 50 %
overloads applied at low (30 MPa $m^{1/2}$), medium (40 MPa $m^{1/2}$)
and high (50 MPa $m^{1/2}$) stress intensity levels. The test
data for each segment, shown in figures 12 through 17, start
at an initial crack length which already includes the crack
length jump produced by the overload cycle. Crack growth
predictions using the Overload, Wheeler, and Willenborg
retardation models were computed for each segment to provide
a comparison between the predictive capabilities of each
model. Several observations were made on the flexibility of
each retardation model to predict the test data. Discussion
of each of the observations and their effects follows.

The Overload model's flexibility to match test data is
contained in the α parameter in equation (8). This parameter
controls the delay time before normal crack growth resumes
after an overload. Harms used his test data to develop
expressions for α as a function of stress intensity level
for 20 % and 50 % overload ratios. These expressions, given
by equations (15) and (16), were used in the Overload

-37-

model. Thus all the model variables were predefined before
the test segment were analyzed. The predictions for each of
the six segments are shown in figures 12 through 17.

The Wheeler model had flexibility to fit test data by
changing the shaping exponent m. The shaping exponent m is
an empirical parameter dependent upon material and stress
history [15]. The value of m for fatigue cycling generally
ranges from 1.0 to 3.5 . During analysis of the six test
segments the shaping exponent m was treated as a variable to
fit the Wheeler model to the test data. Figure 18 shows how
the best fit value of m was found by trial and error for a
typical test segment. A constant value of m equal to 6.0
accurately predicted the 20 % overload segments. The best
fit value of m for the 50 % overload segments was related to
the stress intensity level at overload application and
varied between 6.0 and 3.5 . This relationship between m and
K is shown in figure 19. In general, the shaping exponent
decreases as K increases at higher overload ratios. A
similar trend was seen in the α parameter which is used to
fit experimental data in the Overload model. Accurate
predictions of retardation at higher overload ratios depends
upon relating α or m to the stress intensity at overload
application. At lower overload ratios α was still related
to the stress intensity level but a constant value of m was
found to adequately predict the retardation affect. The
resulting Wheeler predictions using the best fit values of m

-38-

are shown in figures 12 through 17.

The Willenborg model does not have a parameter, like the Overload and Wheeler models, for use in fitting test data. Instead the model accounts for retardation by reducing the equivalent sustained load fatigue cycle of

$$\Delta K = K_{max} - K_{min} \qquad (26)$$
with $R = K_{min}/K_{max} = .4$

to an effective value calculated by substituting $[K_{max}, K_{min}]_{eff}$ from equation (7) into equation (26) to get

$$\Delta K_{eff} = [\Delta K_{max}]_{eff} - [\Delta K_{min}]_{eff} \qquad (27)$$
with $R_{eff} = [K_{min}]_{eff} / [K_{max}]_{eff}$

Application of the Willenborg model to the test segments showed that R_{eff} did not equal the equivalent fatigue cycle R ratio of 0.4. This is due to the truncation of the minimum K_{eff} value at zero and the reduction of both K_{max} and K_{min} by K_{red} as defined in equation (7). Therefore, the model would not predict the same retardation effect if the equivalent sustained load cycles were modeled at different R ratios. The resulting predictions using the Willenborg model with the equivalent fatigue cycle modeled with an R ratio of 0.4 are shown in figures 12 through 17. Since the Willenborg retardation model was dependent on the modeling technique used to represent the sustained load time it was eliminated from further consideration in this study.

-39-
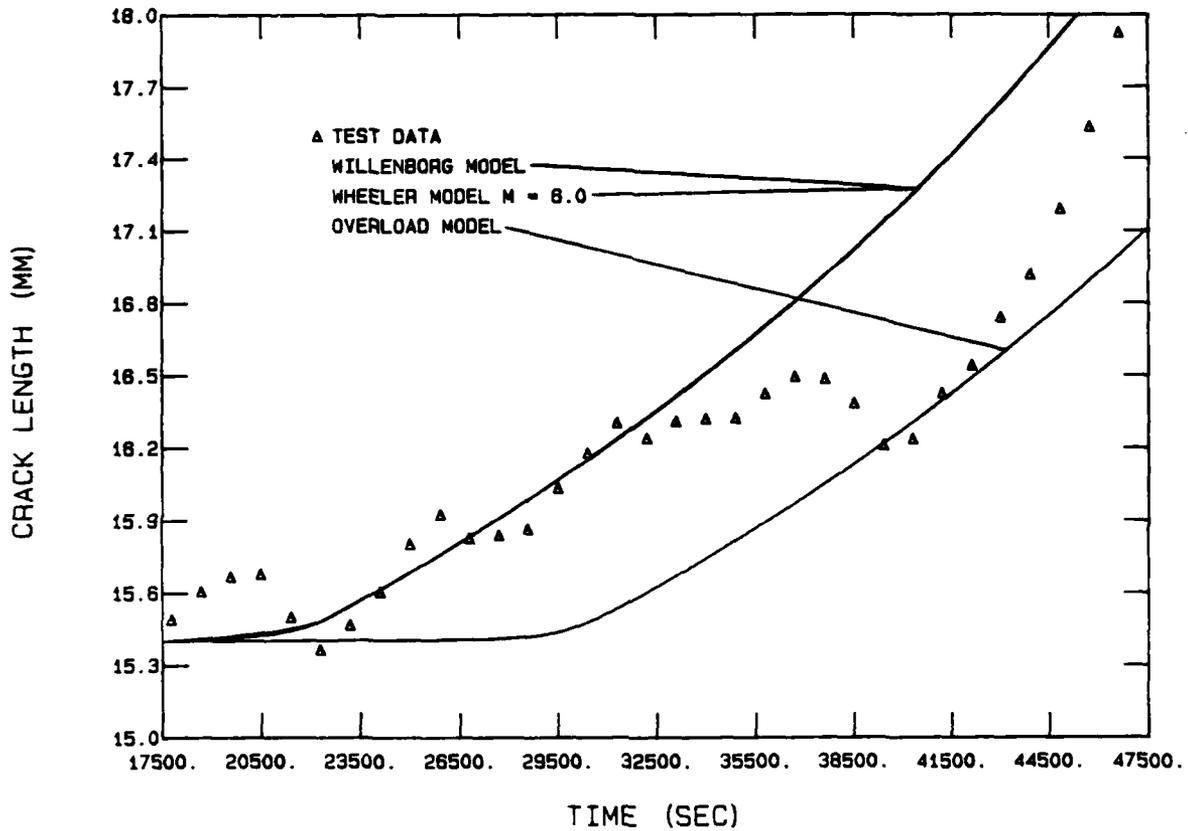
Figure 12 Crack Length versus Time for
20 Percent Overload at Low K.
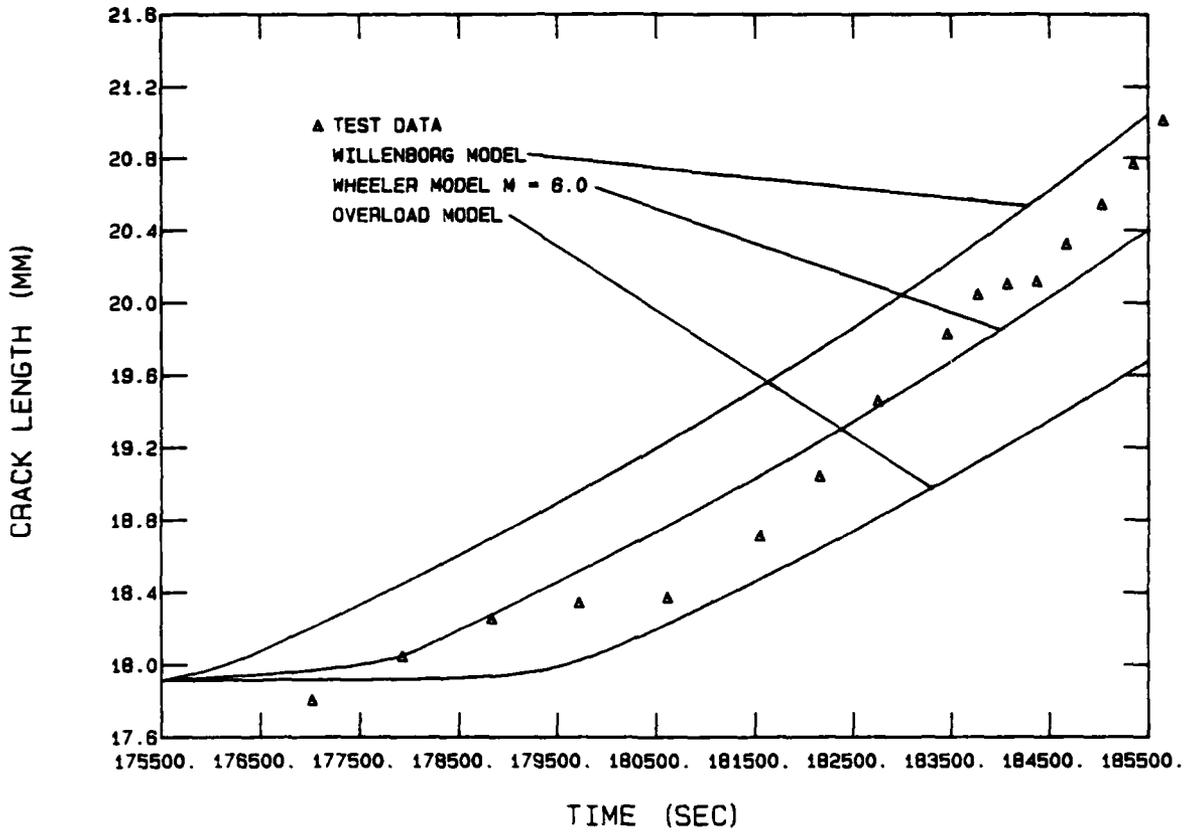
Figure 13 Crack Length versus Time for

20 Percent Overload at Medium K.

-41-
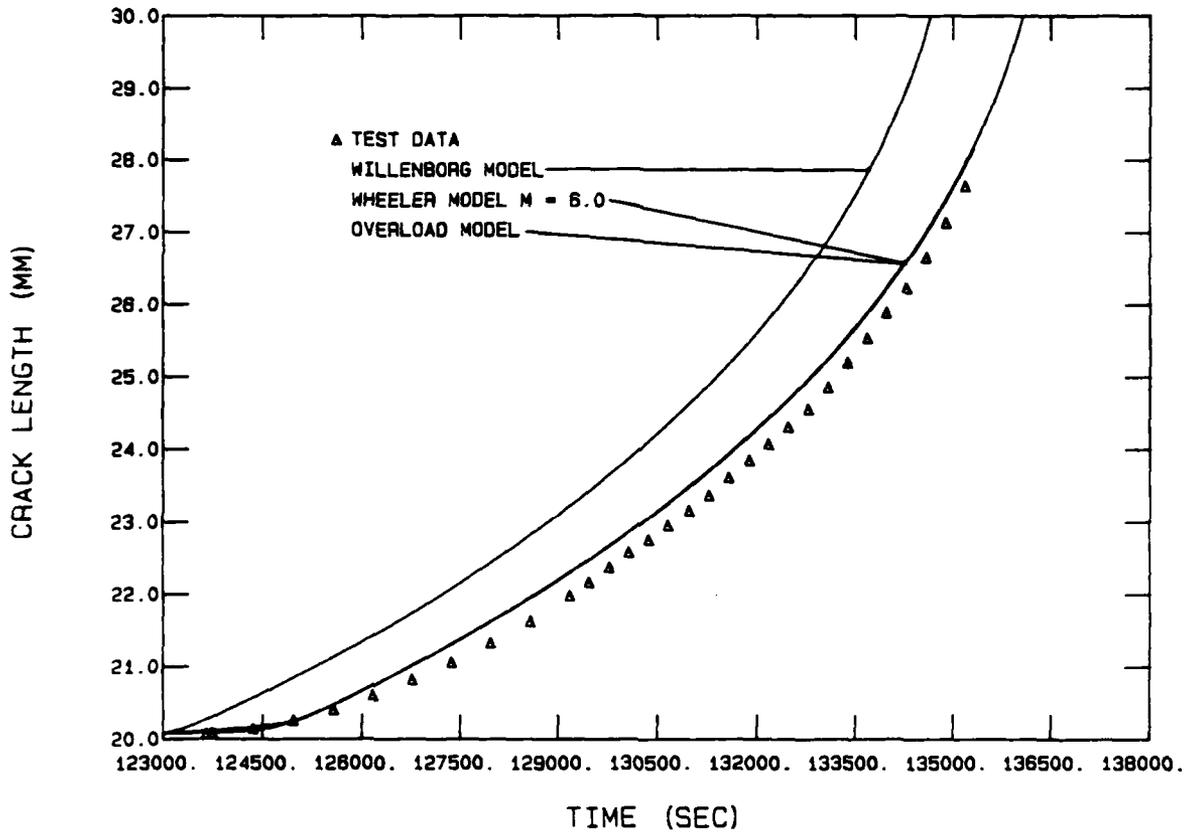
Figure 14 Crack Length versus Time for
        20 Percent Overload at High K.

-42-
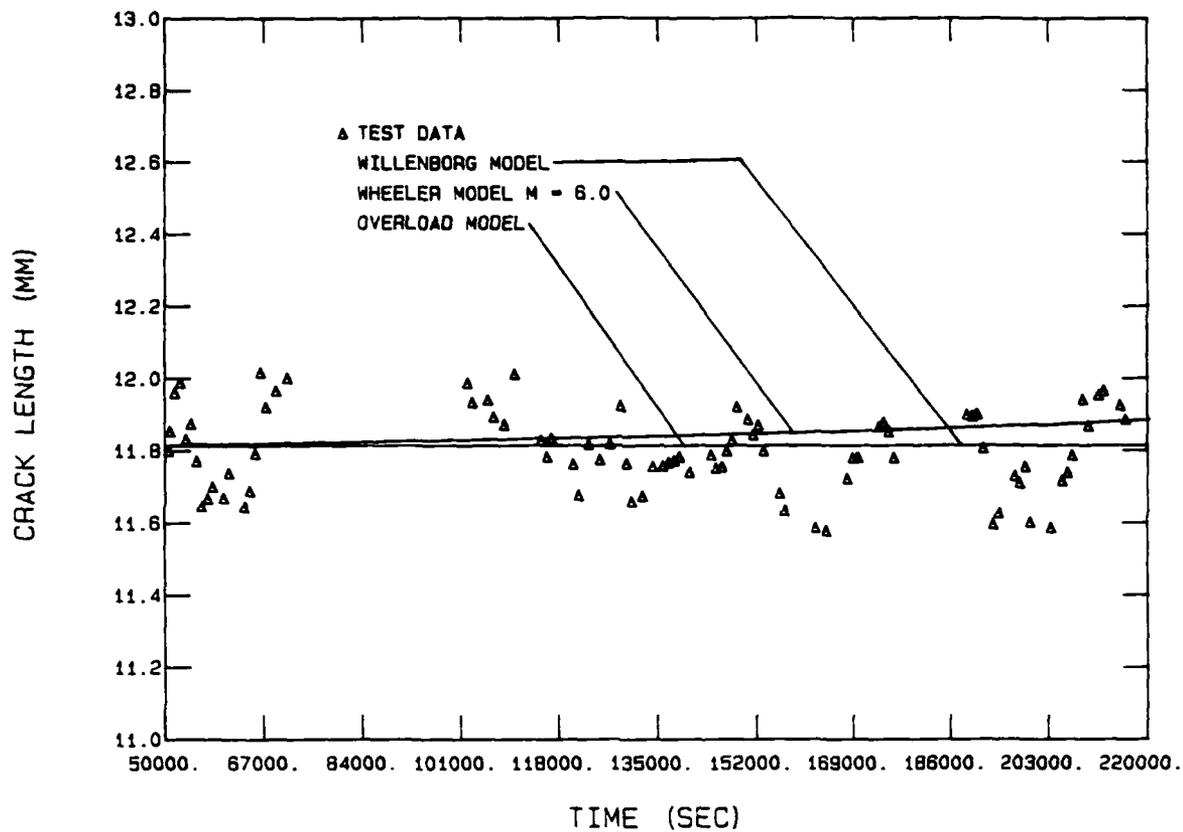
Figure 15 Crack Length versus Time for

50 Percent Overload at Low K.
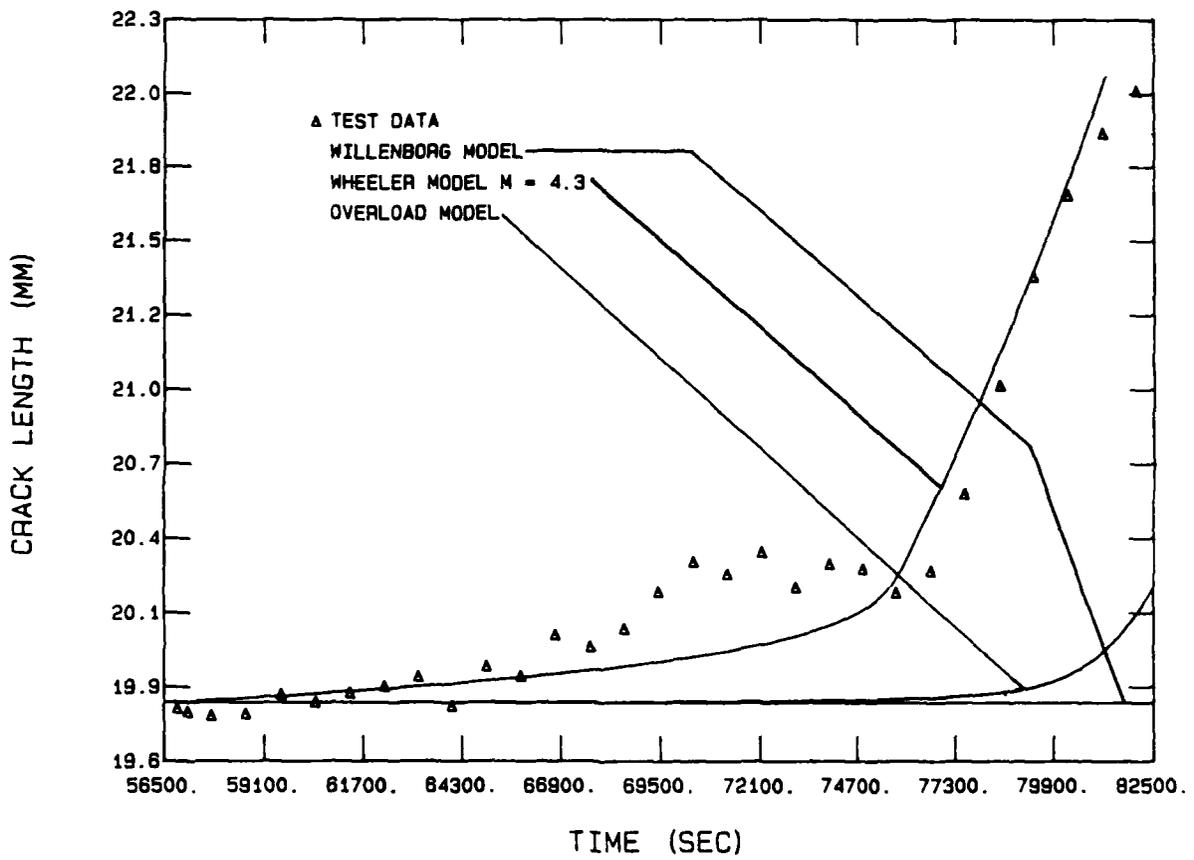
Figure 16 Crack Length versus Time for

50 Percent Overload at Medium K.

-44-
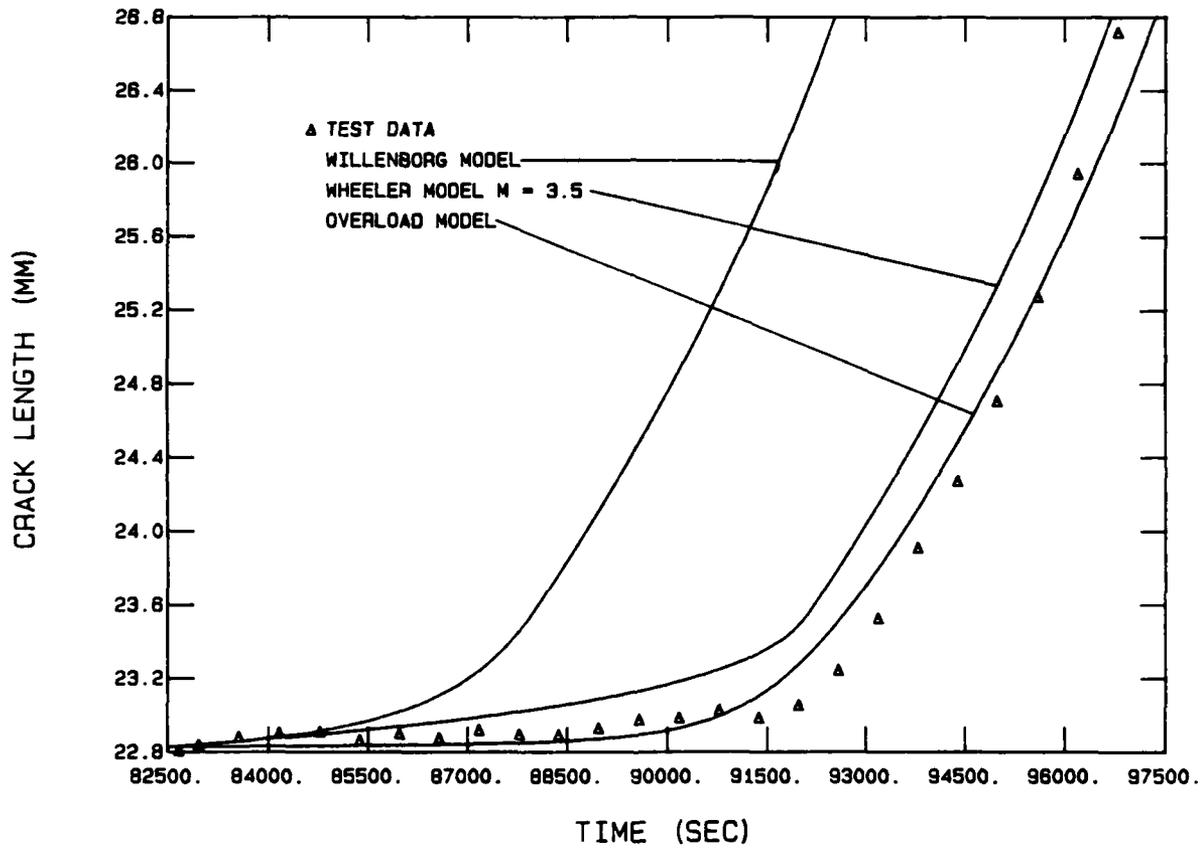
Figure 17 Crack Length versus Time for

50 Percent Overload at High K.

-45-
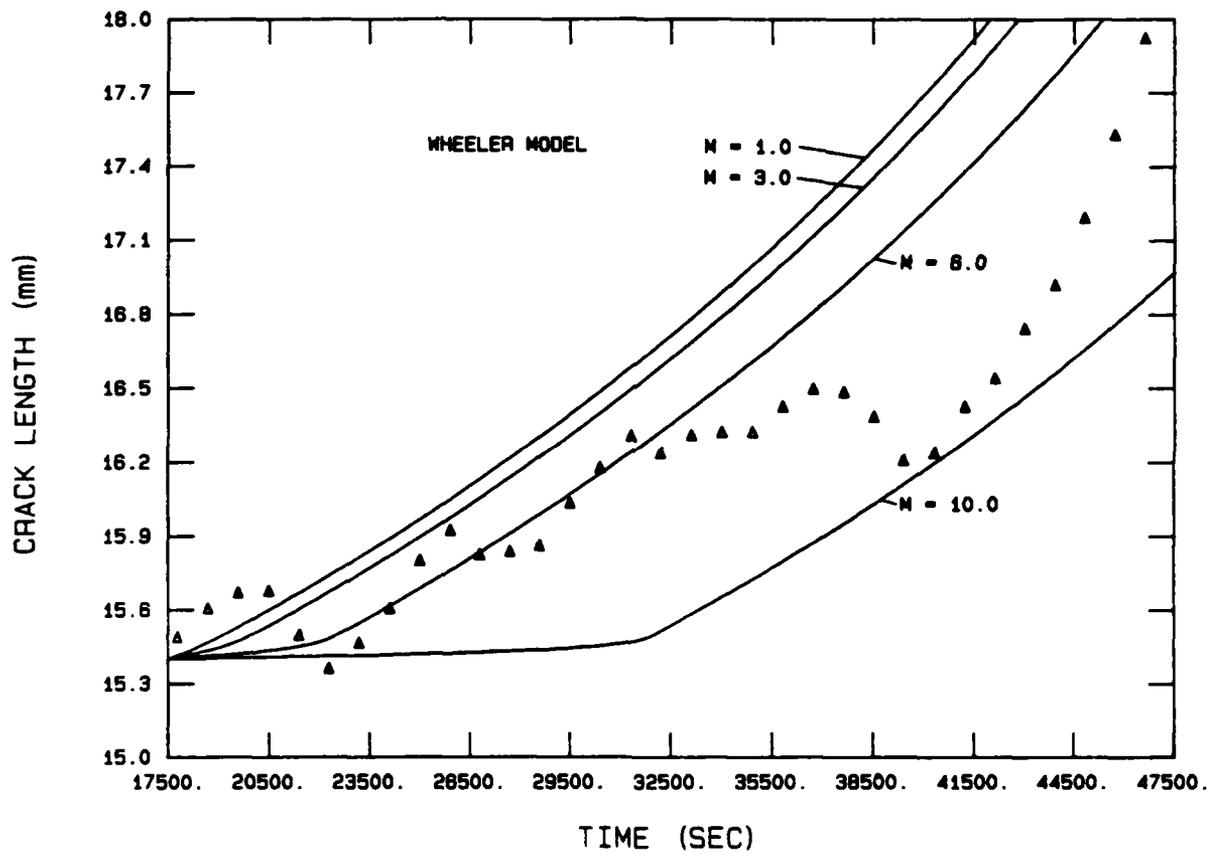
Figure 18 Best Fit Shaping Exponent (m)

for 20 Percent Overload at Low K.
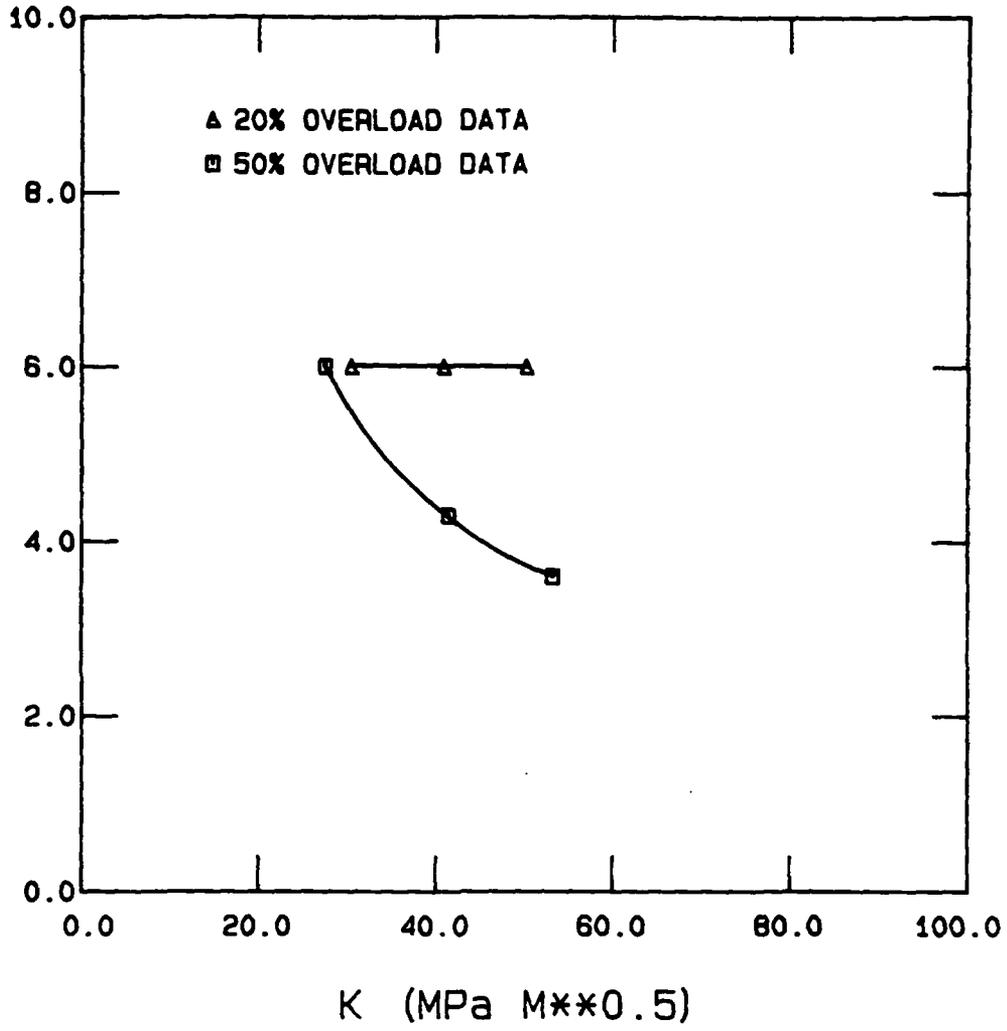
Figure 19 Best Fit Shaping Exponent (m) versus K.

V. Proof Test Experiments

Several proof tests were conducted to provide additional test data to verify the prediction capability of the retardation models. A description of the test apparatus and crack measurement procedures used during testing follows.

Test Apparatus

The experimental data for the proof tests was gathered using a semi-automated creep test system employing electric potential drop and optical readings to monitor crack length. The Air Force Wright Aeronautical Laboratories, Materials Laboratory, Wright-Patterson Air Force Base, Oh. provided the facilities and equipment to conduct the testing. The test system schematic is shown in figure 20. The test setup included the following components:

1. 12,000 lb Swedish creep test frame

2. Resistance heated furnace with power controlers

3. Tektronix 4051 microcomputer

4. Daytronic 9000 signal processor

5. Hewlett-Packard 3478A IEEE-488 digital voltmeter

6. Two Gaertner traveling microscopes

7. Current supply source

The 12,000 lb.-capacity Swedish creep frame was used to

load the specimens. The frame is constructed using a lever and fulcrum principle. The weights were suspended at the end of a 20 to 1 lever arm. The other end reacts to this mechanical advantage with a load line containing the specimen. Dynamic loading of the specimen is avoided by the use of a hydraulic ram which can support a fraction or all of the suspended weight. As pressure is added or removed from the ram, it removes or adds the load to the specimen in a smooth manner. Periodic overloads were applied by unloading the specimen, adding the calculated overload weight to the end of the lever arm, and reloading the specimen. Removing overloads was done using the same procedure except for removing the weights.

The compact tension specimens were mounted in the load line using load rods with Inconel 718 clevises and holding pins. An electric potential technique, to be discussed below, was used to measure crack length. To accommodate this system, the specimen was electrically isolated from the creep frame by using insulated sleeves and pins to connect the load transfer rods. A load cell to measure the applied load to the specimen was included in the load line. The load cell readings were used to ensure the hydraulic ram released the entire load to the specimen. The specimen's configuration and nominal dimensions are shown in figure 21. In addition, the individual test specimen's dimensions and the applied loads are listed in table 1.
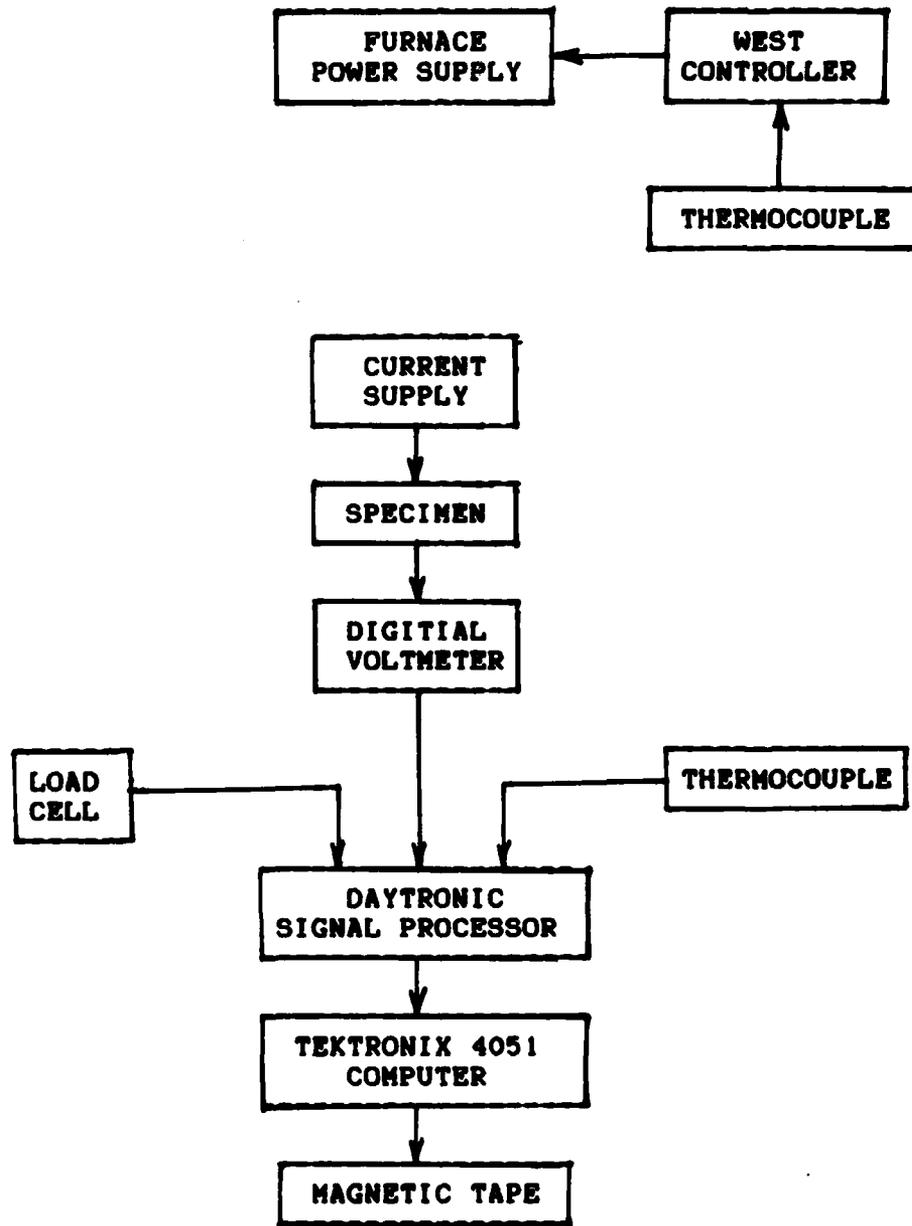
Figure 20 Test System Schematic.
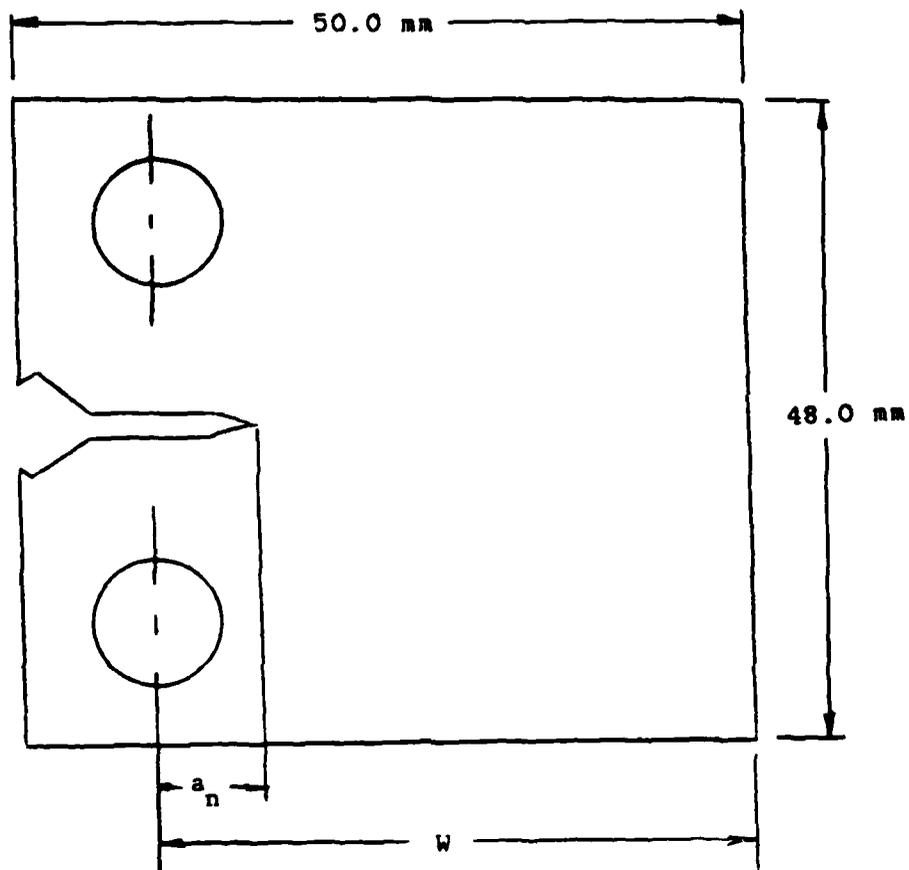
a_n = 7.0 mm
W = 40.0 mr
thickness (B) = 10.5 mm

Figure 21 Compact Tension Specimen.

Specimen	a_i (mm)	Width (mm)	Thickness (mm)	Load (KN)
84-502	10.381	39.969	10.008	12.188
84-503	10.665	40.008	10.033	12.023
84-504	10.267	40.018	10.008	12.677

Table 1   Specimen Dimensions and Applied Load.

The two piece resistance furnace was mounted on the creep frame as shown in figure 22. The oven was closed around the specimen and sealed with a flameproof wadding. The wadding material also served to insulate the leads for the electric potential system from the oven and frame. The furnace was constructed with four independently controlled power zones. The power to each zone is controlled by a time-proportioning West controller. The controller used a thermostat feedback loop to hold the oven's temperature constant. Two K-type, chromel-alumel, thermocouples were spot welded to the specimens and are shown in figure 23. One was used by the controller and the other served as a backup (connected to the Daytronic controller for periodic monitoring).

The Tektronix 4051 microcomputer was used as the data-acquisition system for the tests. The Tektronix was programed to take electric potential readings at predetermined time intervals. The time input signal sent to the Tektronix was provided by the Daytronic controller.
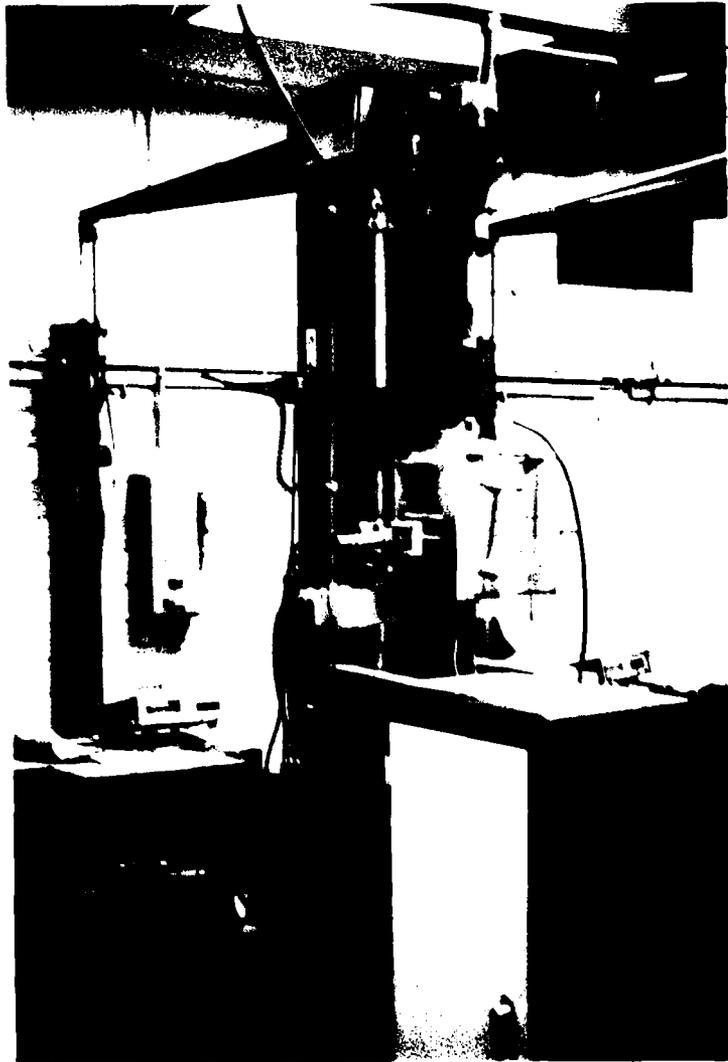
Figure 22 Creep Frame Photograph.
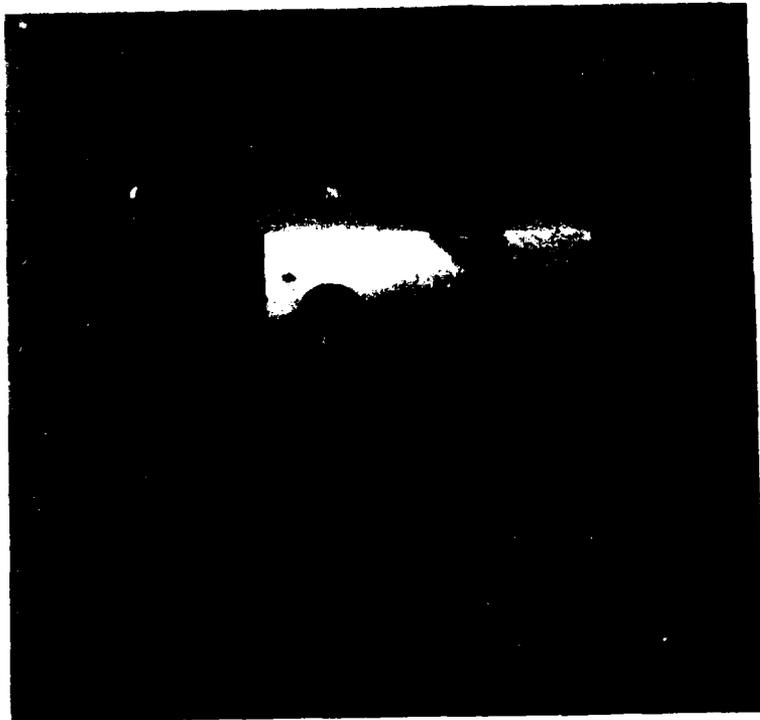
Figure 23 Instrumented Specimen Photograph.

Similarly, the Hewlett Packard 3478A IEEE-488 programable
voltmeter provided the voltage reading when requested by the
Tektronix's program. The two received signals were then
recorded on magnetic tape. This information was later
transferred to the host PDP 11/24 computer for use in the
data reduction procedures to obtain crack length from
voltage. In addition, optical crack length measurements
were obtained through the viewing ports on each side of the
resistance furnace using two Gaertner traveling
microscopes. The crack length measurement was displayed on
a digital readout to an accuracy of 0.0254 mm. However, due
to optical problems in determining exactly where the crack
ended, measurements were reproducible to an accuracy on the
order of 0.05 mm. The optical crack length measurements
were manually recorded periodically along with time and the
corresponding voltage readings from the electric potential
system in order to provide data for later correlation of
crack length to voltage.

Crack Measurement Procedure

The proof experiments utilized a D.C. Electric
Potential (EP) measurement system augmented with optical
readings to monitor crack length. The EP system is based on
the fact that when a current is passed through a conducting
body an electric field is generated. The field shape and
intensity depend upon such factors as applied current,

-55-

geometry shape and material properties. The EP system relies on relating changes in the EP field at the output leads to changes in geometry due to crack growth.

One of the instrumented specimens is shown in figure 23. The input and output leads were welded to the specimen at the locations shown in figure 24. The input leads, made of Inconel 718, were connected to a constant 10.0 amp current source. The output voltage was measured from nichrome wire leads spot welded on the specimen front surface. Nichrome wire was used for the output leads due to its superior oxidation resistance at elevated temperature. However, joining two dissimilar materials produces a thermocouple effect as the joint temperature is changed. This thermal voltage adds algebraically to the voltage generated due to the resistance in the specimen. It should be noted that the thermal voltage is present even when the applied input current is removed from the specimen. It was therefore possible to periodically measure the thermal voltage by shutting the current supply off. Plots of the output voltage versus thermal voltage (Vth) were generated for each test specimen and are shown in figures 25 through 27. The data were fitted with a linear equation using a least squares regression to yield thermal voltage as a function of output voltage reading. This thermal voltage was calculated at each data point and subtracted from the output voltage reading.

-56-
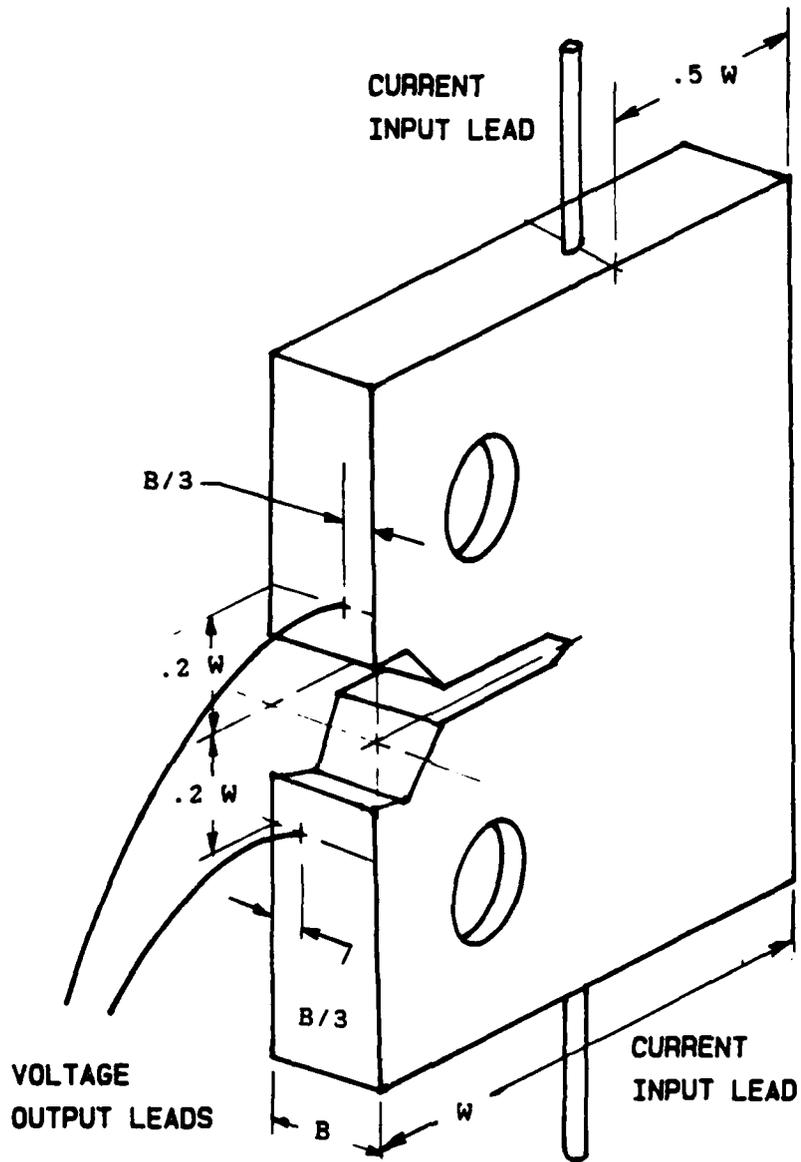
Figure 24 Location of Input and Output Leads on Specimen.

$$Vth = .2409311E\text{-}02 * Vout + .6982624E\text{-}04$$

Figure 25 Thermal versus Output Voltage

Specimen 84-502.

-58-
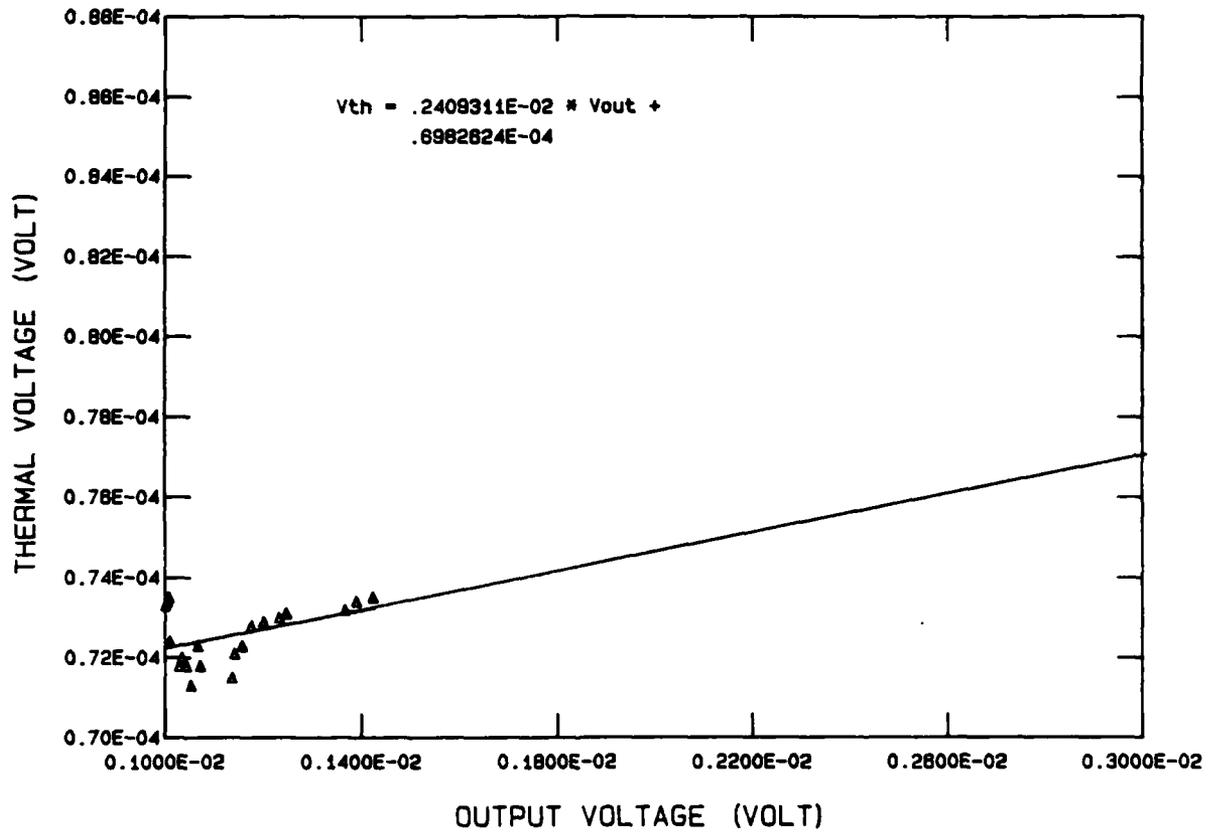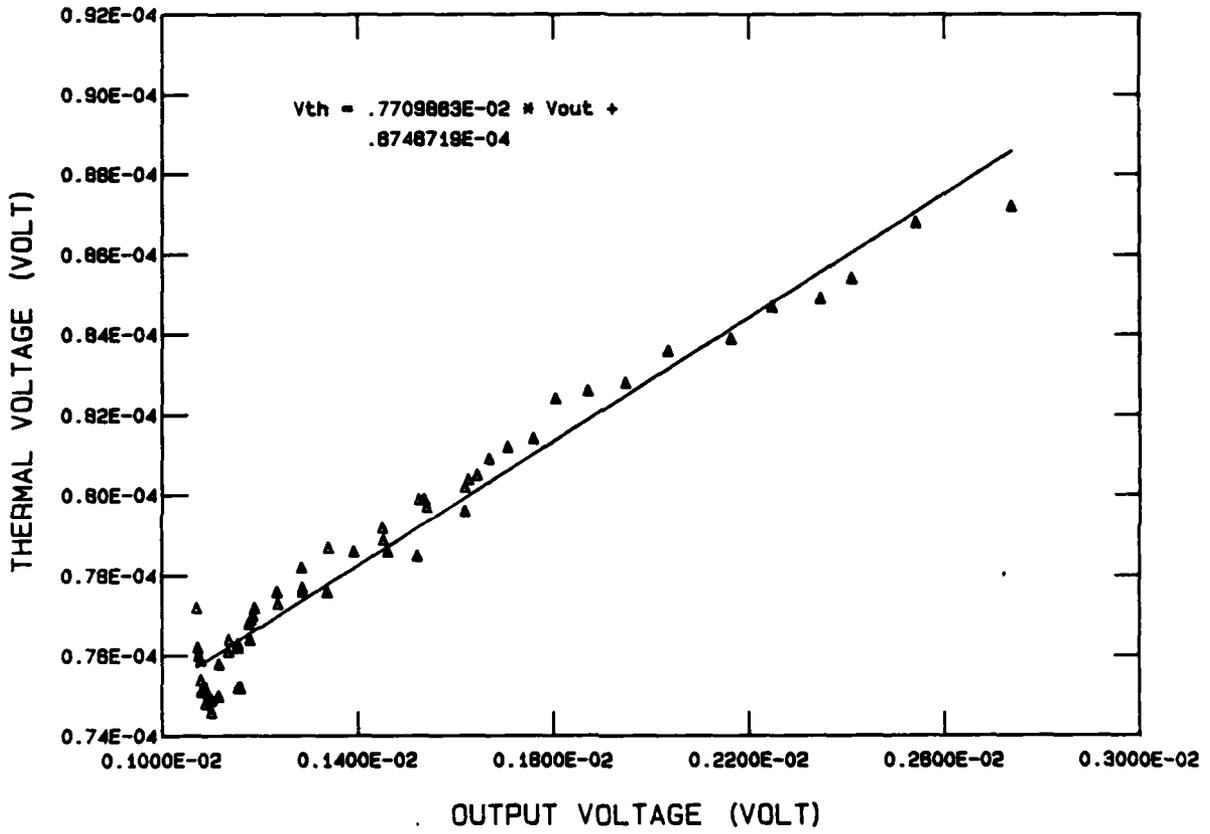
Figure 26 Thermal versus Output Voltage

Specimen 84-503.

-59-
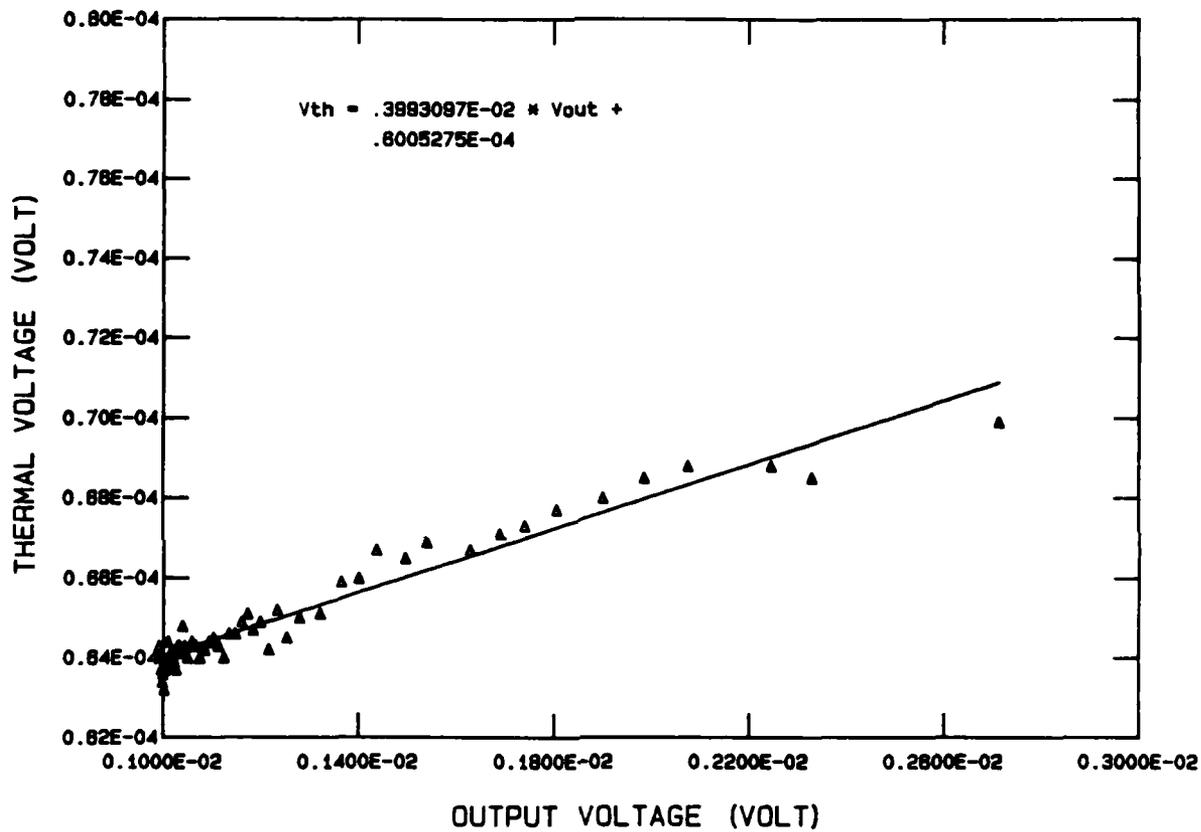
Figure 27 Thermal versus Output Voltage

Specimen 84-504.

-60-

Specimen 84-504-EE5 was tested to provide a calibration curve relating crack length to output voltage under conditions of sustained-load crack growth with no applied overloads. The initial voltage reading, minus thermal voltage, was called the initial reference voltage Vo. The initial crack size corresponding to Vo was measured using a method similar to the one recommended in ASTM E399 [16]. An optical microscope, equipped with a measurement table, was used to make a five point through-the-thickness measurement of the initial crack tip profile on the ruptured specimen. The weighted average, defined in figure 28, was used to calculate an average crack size and a crack tunneling correction factor to account for the increased crack depth at the center of the crack front. The tunneling correction factor was used to bias the surface optical measurements, made at locations a_1 and a_5 in figure 28, to obtain a average crack depth. Calibration specimen 84-504-EE5 had an average tunneling value of 0.445 mm. This value was added to each optical reading to get a corrected a_{opt} value.

A functional relationship between a_{opt} and voltage readings (minus thermal voltage, and normalized to the referenced voltage Vo) was developed using a least squares polynomial fitting program. H. H. Johnson [17] developed complex functions for calibrating crack length measurements to voltage. However, Johnson's results showed that the crack-starter geometry strongly influences the

-61-

$$a_{opt} = (a_1 + a_5)/2$$

$$a_{avg} = ((a_1)/2 + a_2 + a_3 + a_4 + (a_5)/2)/4$$

$$a_t = a_{avg} - a_{opt}$$
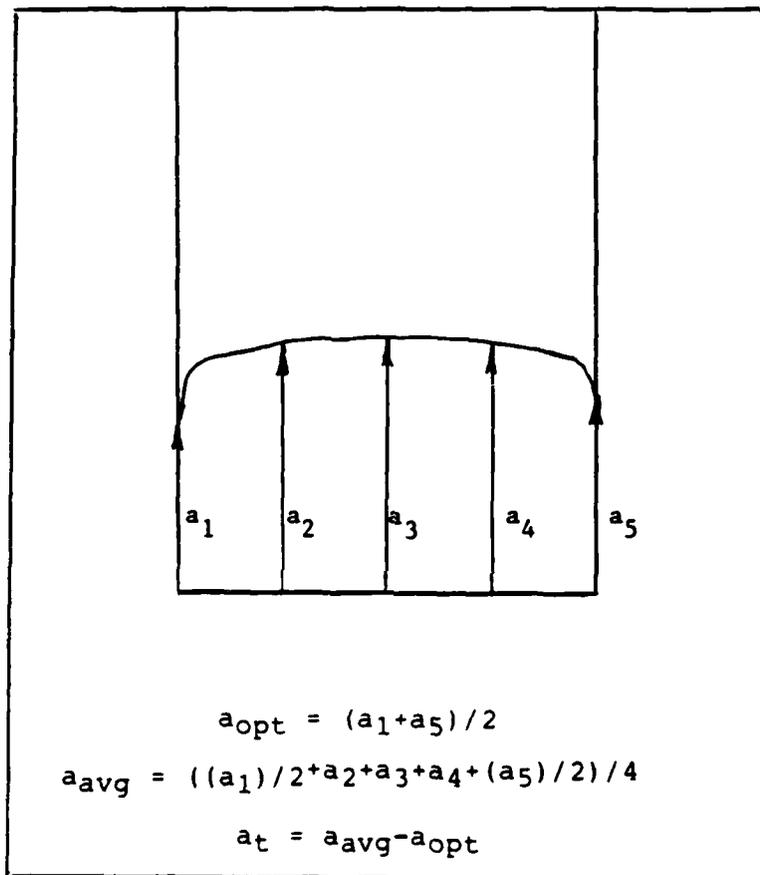
Figure 28 Average Crack Size and Tunneling Correction Term.

calibration between crack length and measured potential.
Therefore for simplicity, an eighth order polynomial was
used which was found to adequately fit the experimental data
and is shown in figure 29. The polynomial equation for crack
length as a function of normalized voltage is,

$$
\begin{aligned}
a = -332.1731 &+ 1503.861 \ (V/V_o)^1 - 2932.647 \ (V/V_o)^2 \\
&+ 3220.104 \ (V/V_o)^3 - 2176.264 \ (V/V_o)^4 \\
&+ 927.0209 \ (V/V_o)^5 - 243.1114 \ (V/V_o)^6 \\
&+ 35.90090 \ (V/V_o)^7 - 2.286718 \ (V/V_o)^8 \quad (inch)
\end{aligned}
\tag{28}
$$

The inverse of this function, normalized voltage as a
function of a_{opt}, was also generated and is shown in figure
30. The corresponding polynomial equation is,

$$
\begin{aligned}
V/V_o = -29.66971 &+ 364.3393 \ (a)^1 - 1859.148 \ (a)^2 \\
&+ 5319.197 \ (a)^3 - 9330.943 \ (a)^4 \\
&+ 10274.08 \ (a)^5 - 6926.514 \ (a)^6 \\
&+ 2612.860 \ (a)^7 - 422.2600 \ (a)^8
\end{aligned}
\tag{29}
$$

where (a) is measured in inches.

This equation was used to calculate the initial voltage
(Vo) for specimens with different initial crack sizes.
Knowing the initial crack size and corresponding voltage
reading (V), an effective Vo was calculated using equation
29. Substituting the ratio of $V/V_{o_{eff}}$ into equation 28
yields the correct initial flaw size for the specimen.

-63-
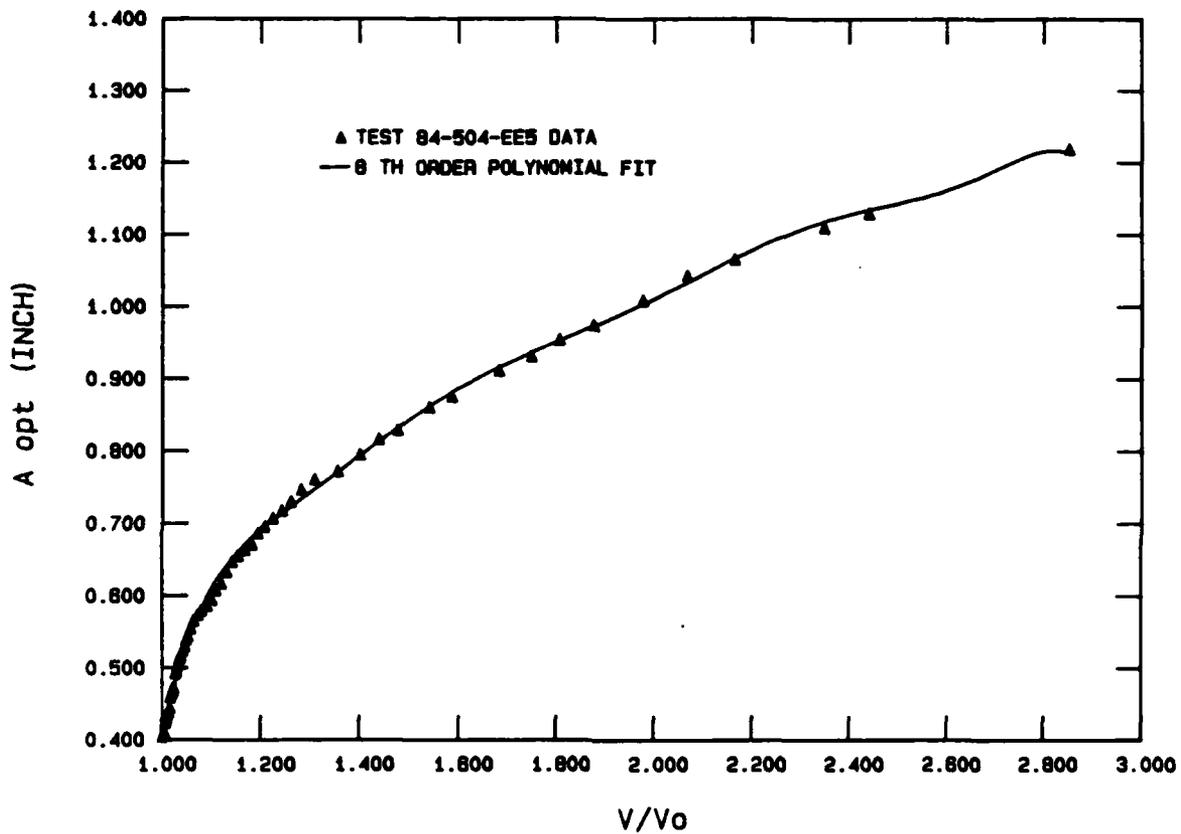
Figure 29 Calibration Curve A_{opt} versus V/V_o.
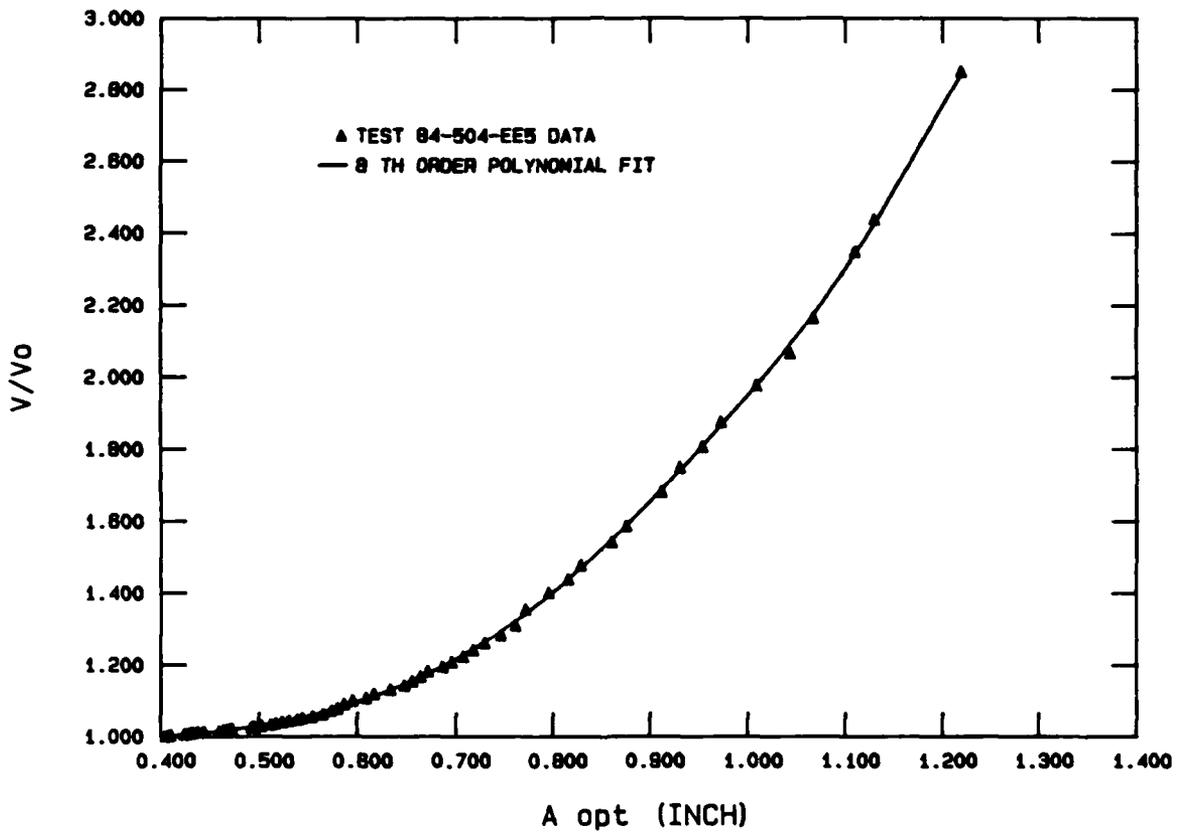
Figure 30 Calibration Curve V/V_o versus A_{opt}.
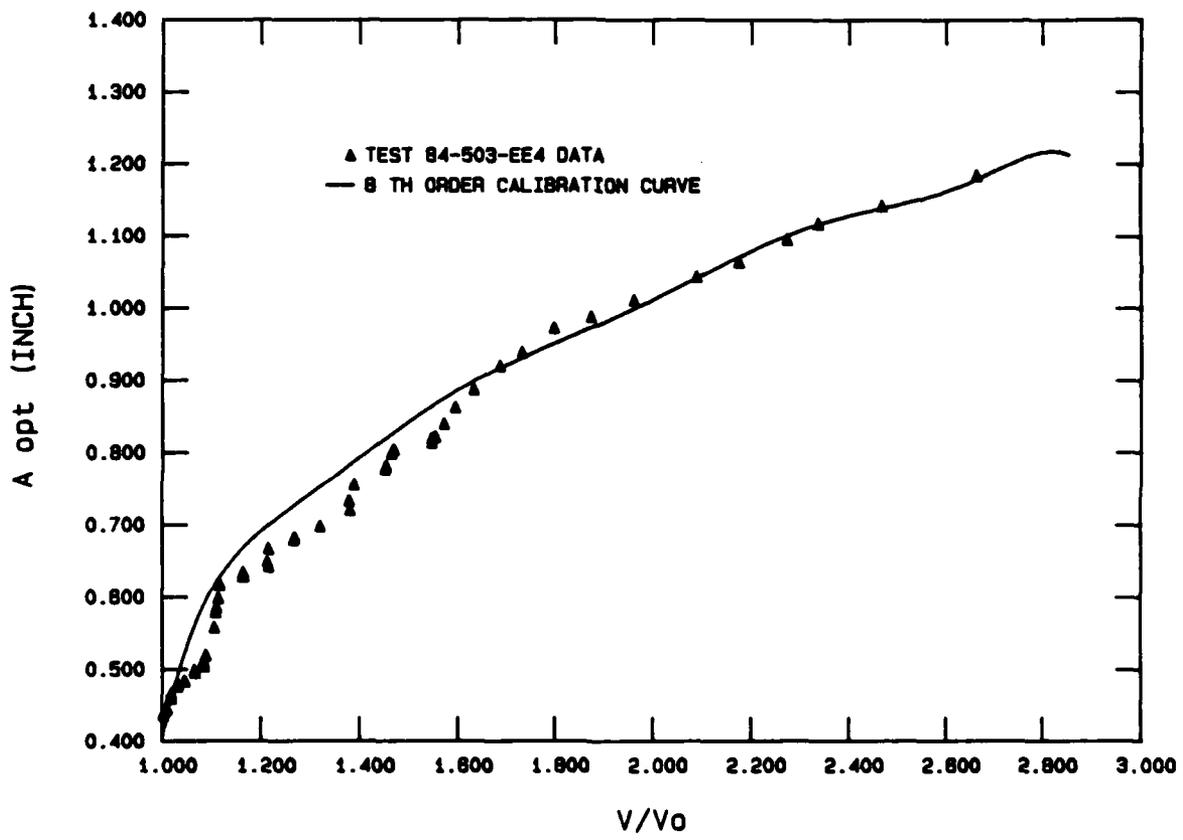
Figure 31 Overload Effect on Calibration Data.

In the tests involving overloads, it was found that each overload caused the a_{opt} versus voltage to shift from the calibration curve. This effect is shown in figure 31. After each overload, there is a period where voltage increases without an appreciable increase in the optical crack size. One explanation is that the crack changes shape with the edge breaking through during the overload cycle. This would account for the jump in optical measurement after each overload. As the crack starts to grow again, evidenced by increasing voltage readings, the surface growth may lag since the crack may tunnel to restore its preferred flaw shape. Although the exact mechanism that cause the deviations from the calibration curve are not fully understood, their effect may be negated by recalibrating the equation after each overload. The procedure is the same one used to calibrate the equation to different initial flaw sizes. Using equation 29 and the a_{opt} and voltage reading taken after each overload, a new Vo was calculated. This Vo was used to normalize voltage readings until the next overload.

The experimental voltage readings recorded by the Tektronix 4051 computer were reduced using the procedures described in this section. The resulting crack growth histories were compared with the analytical predictions and are discussed in the next section.

-67-

VI. Experimental Results and Discussion

Experiments were performed in the course of this study in order to obtain additional crack growth data for evaluating the prediction capability of the retardation models. An electric potential crack-measurement system was used for taking crack length measurements during sustained loading with periodic overloads. The applicability of this system to crack growth following overload was investigated.

Three specimens were used in the experimental work. One was used as a calibration specimen for the electric potential system. The other two specimens were used for the proof tests. In addition, a third proof test, conducted by Harms, was also used to evaluate the retardation models.

The crack growth prediction for the calibration specimen 84-504 is shown in figure 32. This specimen was subjected to sustained loading at 650 C with no overloads. As expected, both the Overload and Wheeler models predicted the same growth, since no retardation cycles were applied. The analytical prediction for time to failure was within 20% of the actual test data. This is well within the normal 2X scatter in crack growth data associated with variations in material properties.

Specimen 84-503 was tested with 20 % overloads applied each hour. The test data and retardation model predictions are shown in figure 33. The test was divided into two

-68-
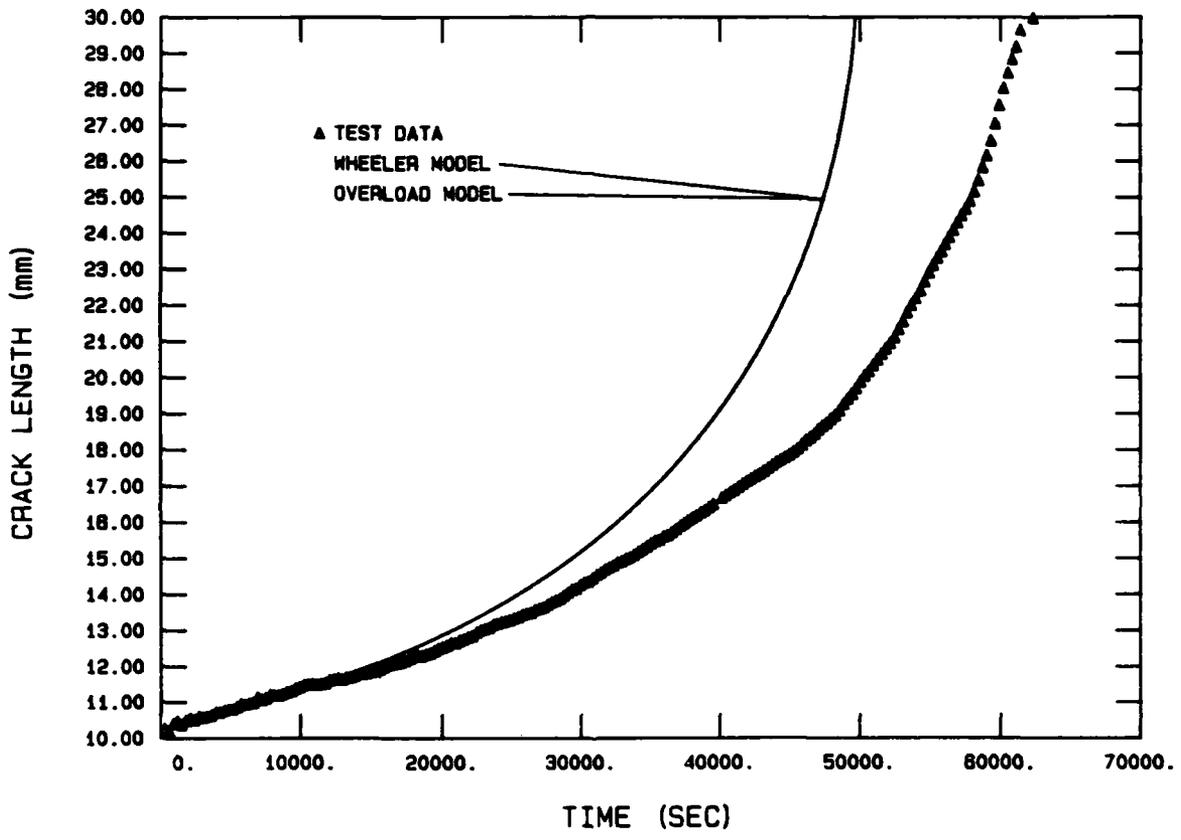
Figure 32 Crack Length versus Time Specimen 84-504.
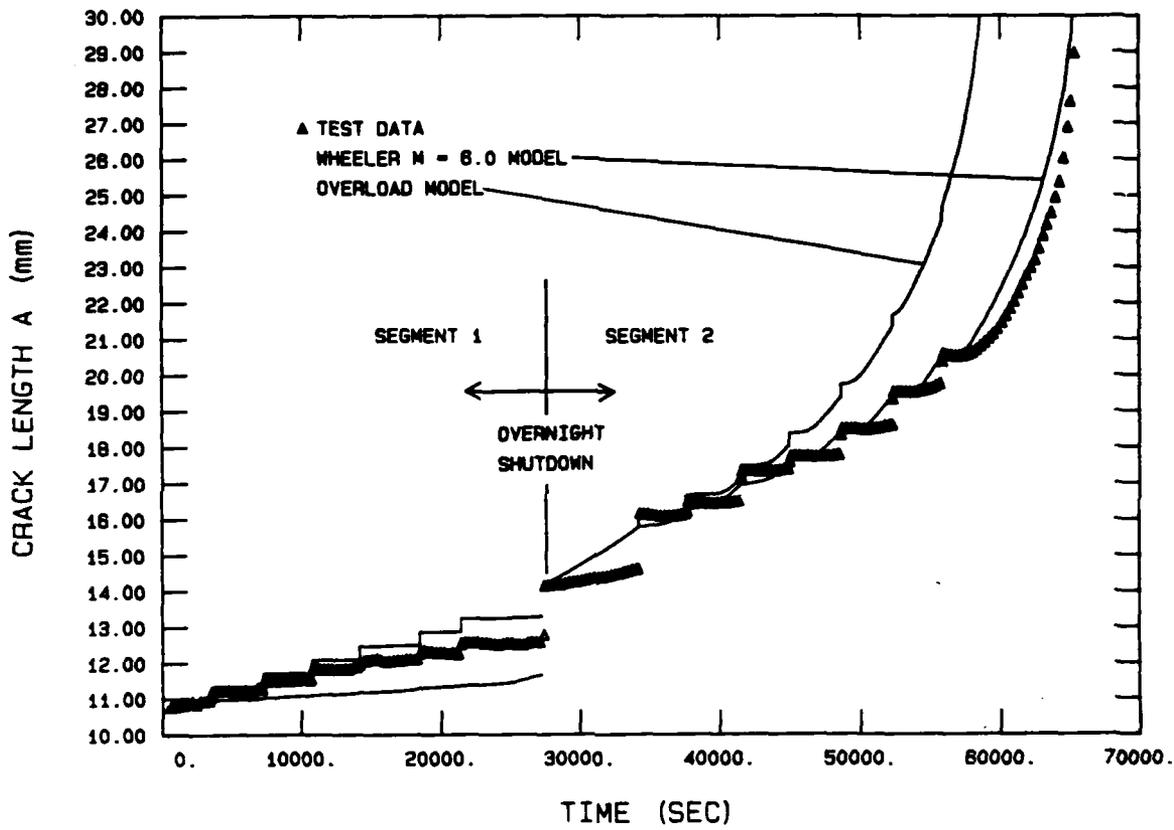
Figure 33 Crack Length versus Time Specimen 84-503.

segments, due to an overnight shutdown in the testing. In the first segment, the crack growth appears to be dominated by the crack jump due to the overload cycle. The Overload model overestimated this jump, while the calculated growth increment from the 0.01 Hz overload cycle before the Wheeler model is applied underestimated it. During the overnight shutdown, the load was removed from the specimen while the temperature was held constant at 650 C. Since the specimen experienced thermal soaking during this shutdown, reloading of the specimen was followed by at least an hour of sustained load growth to re-establish the previous growth rate. The overnight shutdown provided a heat tinted region visible on the fracture surface after the specimen fractured. Optical readings were taken of the crack tip profile to adjust the tunneling correction factor which was then added to the optical surface measurements. The Wheeler model overestimated the total time to failure for the second segment by 1 %. The Overload model underestimated the growth in the same segment by approximately 16 % .

Specimen 84-502 was tested with 20 % overload cycles applied at low 30 (MPa $m^{1/2}$), medium (40 MPa $m^{1/2}$) and high 50 (MPa $m^{1/2}$) stress intensity levels. The effects of the jumps on the crack length, due to overload cycles, was minimized by applying just three overload cycles during the test. The test data and retardation predictions are shown in figure 34. Both models predicted the total time of

-71-
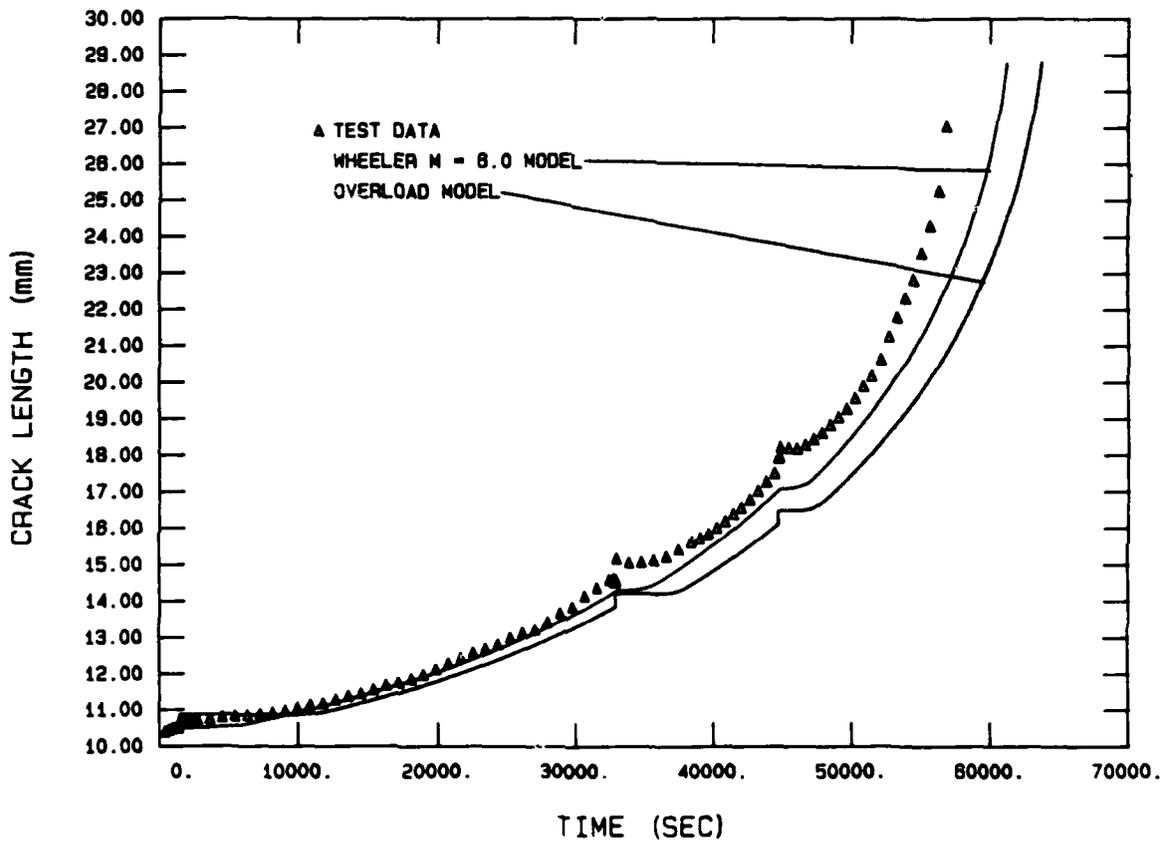
Figure 34 Crack Length versus Time Specimen 84-502.
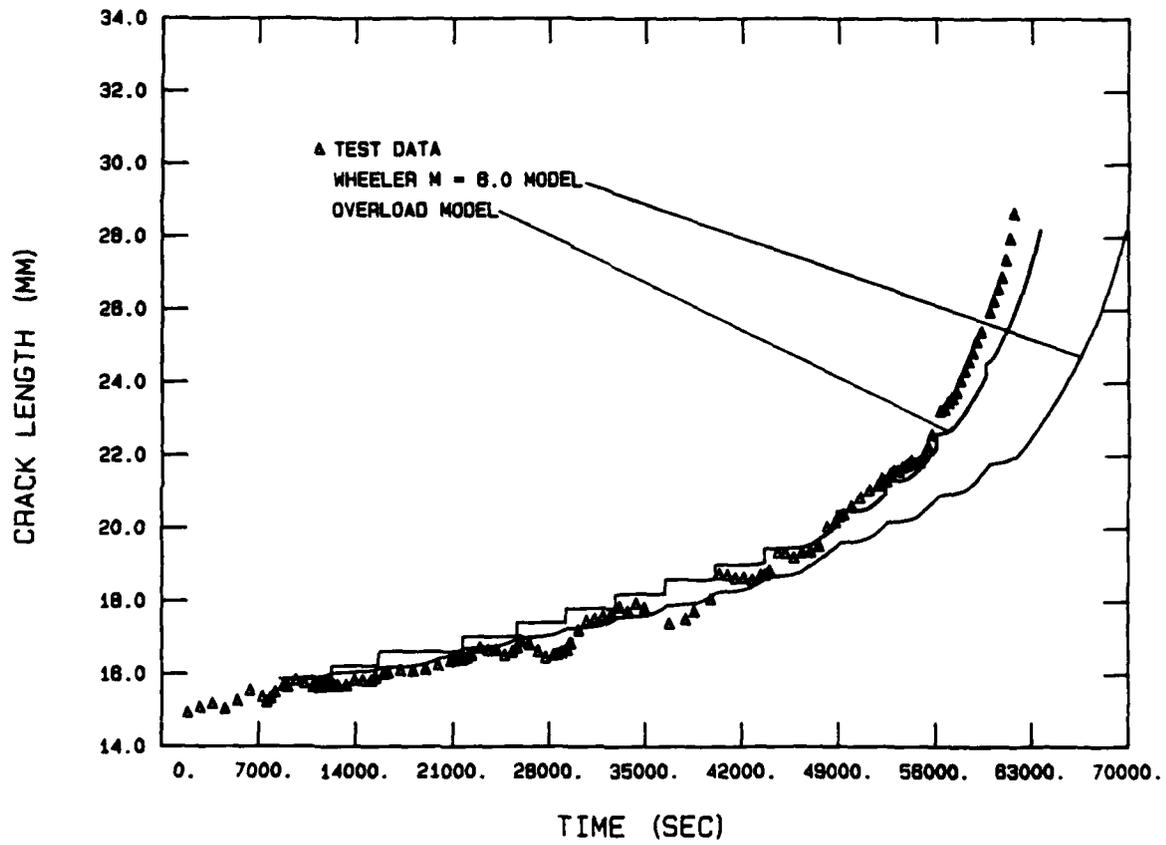
Figure 35 Crack Length versus Time Specimen 84-507.

growth within 10 % of the test data. It was also noted that the retardation predictions were similar in shape to the test data. In general, the models gave better predictions of total time-to-failure when the effect of the crack jumps was minimized.

In addition to the tests conducted as part of this study, the proof test conducted by Harms was analyzed. The test data and retardation predictions are shown in figure 35. The test spectrum consisted of 20 % overload applied each hour. The prediction of the Overload model was within 4 % of the total time-to-failure, but was dependent on the jump in crack length to account for crack growth at lower stress intensity levels. The Wheeler model again underestimated the growth due to the overload cycles and therefore predicted a growth time longer than the actual data.

Accurate prediction of the retardation caused by overload cycles requires empirically defining the α parameter for the Overload model and the shaping exponent m for the Wheeler model. These variables were defined for a specific overload ratio, temperature and test geometry. Both α and m were found to depend upon the stress intensity level at overload application. Using the predefined values of α and m from section IV, both models predicted the retardation effect of overload cycles, when the cycles were applied far enough apart so that no interations occured

between cycles. When the overload cycles were applied at
closer intervals, the crack growth was dominated by the
jumps in the crack length caused by the overload cycles. As
seen in the predictions in figures 33 and 35, the Overload
model did not accurately in predict the jump in crack length
when overload cycles were applied. Defining a function
relating the jump in crack length to the current stress
intensity level was difficult due to the large scatter in
crack length jump measurements. An attempt was made to
reduce the scatter in crack length jump measurements by
using an electric potential system to measure the crack
length. This attempt was unsuccessful mainly due to the
electric potential system requiring recalibration via an
optical crack measurement after each overload. Optical
crack measurements on a specimen under sustained loading are
difficult because the crack tip is not sharply defined.
Instead, the crack tip looks like a deformation zone, with
the exact tip location unknown. The accuracy of both models
for predicting crack growth when the overload cycles
interact could be improved by further refinement of the jump
function.

Overall both retardation programs predicted the total
time to failure within the normal 2X scatter associated with
variations in material properties. Using the CRACKS program
it is possible to analyze spectra of engine fatigue cycles
with the sustained-load growth between cycles included.

-75-

VII. Conclusions and Recommendations

Conclusions

During the course of this study several observations
were make on the applicability of using existing retardation
models, developed for airframe application, to predict
sustained load crack growth retardation. It was found that:

1.) The crack growth rate for sustained loading (da/dt)
and overload fatigue cycles (da/dn) can be represented by
one crack growth rate equation. This is accomplished by
modeling the sustained load time as equivalent fatigue
cycles. The equivalent cycle's period and R ratio are
adjusted to obtain the overload fatigue cycle's crack growth
rate. The equivalent sustained load fatigue cycles and
overload cycles were analyzed using the CRACKS crack growth
prediction program developed for airframe cyclic loading.
The CRACKS program is capable of predicting the total time
to failure within 20 % of experimental data.

2.) The Overload and Wheeler retardation models depend
on empirical parameters that are related to the stress
intensity level at overload application.

3.) The jump function dominates the crack growth in the
Overload model when the overload cycles are spaced close
enough to interact.

4.) The Willenborg retardation model modifies the
fatigue cycle's R ratio when accounting for retardation.

-76-

Thus, the retardation model was dependent on the R ratio chosen for the equivalent sustained load fatigue cycles and was deemed unacceptable for use.

5.) The electric potential crack measurement system is affected by overload cycles and requires recalibration after an overload cycle is applied.

This study verified that the Overload retardation model predicts sustained-load crack growth with periodic overload within normal test data scatter. In addition, procedures were developed to convert sustained loading into equivalent fatigue cycles and analyze crack growth using the CRACKS program. The CRACKS program, with the Wheeler retardation model selected to account for retardation effects, is capable of predicting the total time to failure within 20 % for tests consisting of sustained-load with periodic overloads.

The modified CRACKS program offers the unique capability to analyze sustained-loading and fatigue cycle loading together.. This capability can be readily applied to complex engine spectra, consisting of fatigue cycles with hold times, to predict the crack growth in engine components.

Recommendations

In this study it was found that the electric potential
system for measuring crack length was affected by periodic
overloads. Additional investigation is required to
understand how electric potential crack measurements made on
fatigue loaded specimens with periodic overloads compare
with sustained-loaded specimens with periodic overloads.

This study used Inconel 718 material exclusively for
experimental testing. Additional testing should be
performed to determine if the models apply to other
materials. Another nickel-base superalloy such as Rene 95
is recommended.

Finally, the retardation parameters for the Overload
and Wheeler models should be developed for a wider range of
overload ratios and used to analyze a more complex spectrum.

Bibliography

1. Hill, R. J., Reimann, W. H., and Ogg, J. S., "A
Retirement-For-Cause Study of an Engine Turbine Disk:,
AFWAL-TR-81-2094. Wright-Patterson AFB, OH 1981.

2. Harris, J. A., Sims, D. L., and Annis, C. G., "Concept
Definition: Retirement for Cause of F100 Rotor Components",
AFWAL-TR-80-4118. Wright-Patterson AFB. OH 1980.

3. Military Standard: Engine Structural Integrity Program,
MIL-STD-1783 (USAF) 30 November 1984.)

4. Engle, R.M., "CRACKS, A FORTRAN IV Digital Computer
Program for Crack Propagation Analysis," AFFDL-TR-70-107,
Air Force Flight Dynamics Laboratory, Wright-Patterson Air
Force Base, Oh, Oct. 1970

5. Wheeler, O.E., "Spectrum Loading and Crack Growth",
Journal of Basic Engineering, March 1972, pp 181-186.

6. Willenborg, J., Engle, R.M., and Wood, H.A., "A Crack
Growth Retardation Model Using an Effective Stress Concept",
AFFDL-TM-71-1-FBR, Wright-Patterson AFB, Oh, Jan 1971

7. Harms K. E., "Overload Effects on Sustained Load Crack
Growth at Elevated Temperatures", AFIT Thesis,
AFIT/GAE/AA/84D-8, Wright-Patterson AFB, Oh, 1984.

8. Weerasooriya, T., Nicholas, T., "Overload Effects in
Sustained Load Crack Growth in Inconel 718",
AFWAL-TR-85-4121. Wright-Patterson AFB, OH 1985.

9. Annual Book of ASTM Standards, Part 10;
Metals-Mechanical, Fracture, and Corrosion Testing; Erosion
and Wear; Effect of Temperature, E 647-83, page 719, 1983.

10. Utah, D. A. General Electric Company, "Crack Growth
Modeling in an Advanced Powder Metallurgy Alloy",
AFWAL-TR-80-4098, Wright-Patterson AFB, OH 1980.

11. Kreyszig, E., Advanced Engineering Mathematics, Fifth
Edition, New York, John Wiley and Sons, 1983, pp 788-789.

12. Larsen J. M. and Nicholas, T. "Load Sequence Crack
Growth Transients in a Superalloy at Elevated Temperature",
Fracture Mechanics: Fourteenth Symposium Volume II: Testing
and Applications, ASTM Society for Testing and Materials,
1983, pp II-536-II-552.

13. Haritos, G.K., Nicholas, T. and Painter, G.O., "Evaluation of Crack Growth Models for Elevated Temperature Fatigue," Eighteenth National Symposium on Fracture Mechanics, ASTM STP XXX, American Society for Testing and Materials, 1985, pp xx-yy

14. Engle, R. M. "Damage Accumulation Techniques in Damage Tolerance Analysis", ASTM STP 842, American Society for Testing and Materials, 1984, pp. 25-35.

15. Collins, J. A., Failure of Materials in Mechanical Design, New York, John Wiley and Sons, 1981, Ch. 8.

16. Annual Book of ASTM Standards, Part 10; Metals-Mechanical, Fracture, and Corrosion Testing; Erosion and Wear; Effect of Temperature, 1981.

17. Johnson, H. H., "Calibrating the Electric Potential Method for Studying Slow Crack Growth", Materials Research and Standards, Vol. 5, No. 9, 1965

Appendix A

Heat Treatment History of Test Specimens

Anneal at 968 C for 1 hour - air cool

Age harden at 718 C for 8 hours - furnace cool to 621 C

Age harden at 621 C for a total of an additional 10 hours

Appendix B.

OVERLOAD PROGRAM

```
C      PROGRAM OVERLD
       DOUBLE PRECISION V
       REAL*4 LOWER,KUP,KLOW,N,LOWW,KINIT,LOWERR,KLOWW,
      1KUPP,LOWR,LOAD,K
       INTEGER*4 PIECES,D,YNCR,NAME(7)
       DIMENSION TITLE(60),SBTITL(60),TIME(20),OL(20)
       COMMON/LYIA1/PI,Q,TOL,W,B,T,R,
      1QUE,PEA,BEE,DEE,DKSTAR,TOUGH,DKCRIT,
      2TRIG,LOWW,DAGER,NSKIP
C      DEFAULTS FOR DIAGNOSTIC OPTION
       DATA V/0/,PI/3.14159/,TOL/.0001/,R/.0/,ZERO/0.0/,T/.25/,TRIG/1./
C ***** OPEN FILES    2:INPUT  1:OUTPUT  3:PLOT  ***********************
       WRITE (5,40)
   40  FORMAT (' WHAT IS YOUR INPUT FILE? ',$)
       READ (5,90) NAME
       OPEN (UNIT=2,NAME=NAME,TYPE='OLD')
       WRITE (5,50)
   50  FORMAT (/,' WHAT SHOULD YOUR OUTPUT FILE BE CALLED? ',$)
       READ (5,90) NAME
       OPEN (UNIT=1,NAME=NAME,TYPE='NEW')
       WRITE (5,60)
   60  FORMAT (/,' WHAT SHOULD YOUR PLOT FILE BE CALLED? ',$)
       READ (5,90) NAME
       OPEN (UNIT=3,NAME=NAME,TYPE='NEW')
   90  FORMAT (11A2)
C ***** ALL FILES ATTACHED *****************************************
C
C ***** READ INPUT FILE *******************************************
       READ (2,1120) TITLE
       READ (2,1120,END=1590) SBTITL
       READ (2,*) TOUGH,LOWW,R,YIELD
       READ (2,*) BEE,DKSTAR,PEA,QUE,DEE
       READ (2,*) B,LOAD,W
       WRITE (5,105)
       WRITE (5,110)
  105  FORMAT (/,' DO YOU WANT TO BREAK OUT ON A CRACK LENGTH')
  110  FORMAT (' OR AT THE KIC STRESS INTENSITY [1=A, 0=KIC]-->',$)
       READ (5,*) AFIN
       IF (AFIN.EQ.1) GO TO 150
       AFIN=10000.
       GO TO 160
  150  WRITE (5,155)
  155  FORMAT (/,' INPUT YOUR FINAL CRACK LENGTH, INCHES-->',$)
       READ (5,*) AFIN
C ***** END OF INPUT *********************************************
C
C ***** PRINT OUTPUT FORMATS ON FILES ***************************
```

```
 160   WRITE (1,1130) TITLE
       WRITE (3,1130) TITLE
       WRITE (5,1130) TITLE
       WRITE (1,1130) SBTITL
       WRITE (3,1130) SBTITL
       WRITE (5,1130) SBTITL
       WRITE (1,740) TOUGH,R
       WRITE (1,850) YIELD
       WRITE (1,830) B,W
       WRITE (1,930) BEE,DKSTAR,PEA
       WRITE (1,940) DKSTAR,QUE
       WRITE (1,950) TOUGH,DEE
       WRITE (1,1270)
       WRITE (1,1280)
 740   FORMAT (' FRACTURE TOUGHNESS:',F6.2,/,' STRESS RATIO:',F5.3)
 830   FORMAT (' THICKNESS: ',F7.5,/,' WIDTH:',F7.5)
 850   FORMAT (' YIELD STRENGTH:',F6.2)
 930   FORMAT (' da/dN=EXP[',F6.2,']*[[DELTA-K/',F6.2,']**',F6.2,']*')
 940   FORMAT (' [[ln[DELTA-K/',F6.2,']]**',F6.2,']*')
 950   FORMAT (' [[ln[',F6.2,'/DELTA-K]]**',F6.2,']')
1120   FORMAT (60A2)
1130   FORMAT (1X,60A2)
1270   FORMAT (35X,' delta')
1280   FORMAT (1X,' N(x1000)',10X,'a',9X,'LOAD',2X,
      1'  K  ',1X,'  da/dN  ',1X,'PIECES')
C ***** END OF HEADERS *********************************************
C
C ***** READ IN SPECTRUM LOAD ************************************
       NA=1
  95   READ (2,*,END=100) TIME(NA),OL(NA)
       NA=NA+1
       GO TO 95
 100   NA=NA-1
C ****************************************************************
       TOUGH=(1.0-R)*TOUGH
       DKCRIT=TOUGH
       TOUGH=TOUGH*.95
1210   STAR=1.0
       KUP=0
       V=TIME(1)
       SIGMA=LOAD
C ***** START OF SPECTRUM INTEGRATION **************************
       DO 1570 NS=1,NA
       DAGER=OL(NS)
       NSKIP=0
       DOL=0
       U=LOWW
1290   LOWER=U
C ****************************************************************
C
1320   CALL SIMP(SUM,LOWER,UPPER,SIGMA,KLOW,KUP,PIECES,YIELD,STAR,PYTCH,
      1CALPHA,BETA,DOL)
```

- 83 -

```
C
C ***** Check if initial K is greater than K1c **********************
      IF (KLOW.LT.TOUGH)GO TO 1360
      WRITE (5,1330)
      WRITE (1,1330)
1330  FORMAT (10X,' *****SPECIMEN FAILED ON LOADING*****')
      GO TO 1490
C
1360  VN=V
      V=V+(SUM/1000)
C ***** CHECK IF INTEGRATED TIME IS MORE THAN THE NEXT OVERLOAD *
C
      IF (NS.EQ.NA) GO TO 1370
      IF (V.LT.TIME(NS+1)) GO TO 1370
      IF (NSKIP.EQ.1) GO TO 1370
      PYTIT=PYTCH*(TIME(NS+1)-VN)/(SUM/1000.)
      UPPER=LOWER+PYTIT
C ************ EXTRA PRINT TO CHECK LINEAR PYTCH INCREMENT *****
C      WRITE (1,1331) V,VN,NS,PYTIT,LOWER,UPPER,SUM
C1331  FORMAT (1X,'V= ',F15.6,' VN = ',F15.6,' NS= ',I2,/,' PYTIT= ',
C      1F15.10,/,' LOWER= ',F16.13,'UPPER= ',F16.13,' SUM = ',F9.3)
C ****************************************************************
      V=VN
      NSKIP=1
      GO TO 1320
C
1370  CALL FINDF2 (KLOW,F2)
      DADNL=1/F2
      CALL FINDF2 (KUP,F2)
      DADNU=1/F2
      IF (U.EQ.LOWW) WRITE (1,1421)VN,LOWER,SIGMA,KLOW,DADNL,PIECES
C
C *****  CONVERSION FROM ENGLISH TO METRIC TO WRITE PLOT FILE *******
C
      CBETA=BETA
      CBETA=CBETA/25.4
C      1/INCHES-->1/MILLIMETERS
C
      LOWERR=LOWER*25.4
C      INCHES-->MILLIMETERS
      DADNLL=DADNL*0.0254
C      INCHES PER CYCLE-->METERS PER CYCLE
C
      KLOWW=KLOW*1.0989
C      KSI ROOT INCHES-->MPA ROOT METERS
C
      UPPERR=UPPER*25.4
      DADNUU=DADNU*0.0254
      VV=V*1000
      KUPP=KUP*1.0989
      VVN=VN*1000.
C      IF (U.EQ.LOWW) WRITE(3,1375) CALPHA,CBETA
```

- 84 -

```
       IF (U.EQ.LOWW) WRITE (3,*) VVN,LOWERR,DADNLL,KLOWW
       IF (U.EQ.LOWW) WRITE (5,1420)VN,LOWER,SIGMA,KLOW,DADNL,PIECES
       WRITE (1,1420) ,V,UPPER,SIGMA,KUP,DADNU,PIECES
       WRITE (3,*)  ,VV,UPPERR,DADNUU,KUPP
       WRITE (5,1420)  ,V,UPPER,SIGMA,KUP,DADNU,PIECES
C1375  FORMAT(' ALPHA = ',F8.5,/,' BETA = ',F8.5)
1420   FORMAT (1X,F15.3,F9.6,F9.2,F7.2,E10.3,I7)
1421   FORMAT (1X,F15.4,F9.6,F9.2,F7.2,E10.3,I7)
1430   IF ((KUP.GT.TOUGH).OR.(UPPER.GE.AFIN)) GO TO 1490
       IF (NSKIP.EQ.1)GO TO 1560
       U=U+PYTCH
       CALL CPYTCH (KUP,PYTCH)
       GO TO 1290
1490   WRITE (1,1500)
1500   FORMAT (6X,' C-T SPECIMEN, SANS PLASTIC-ZONE CORRECTION')
       GO TO 1590
C
C ***** ENTER JUMP FUNCTION ***************************
C
1560   XJUMP=.015
       LOWW=U+XJUMP
C
       WRITE (1,1561) XJUMP,KUP
1561   FORMAT(1X,'XJUMP = ',F15.10,' KUP= ',F6.2)
1570   CONTINUE
C     ********************** END DO LOOP ***************
1590   CALL EXIT
       END
C     SUBROUTINES IN DESCENDING ORDER OF USE,FIRST,SIMPSON'S RULE APPROX
       SUBROUTINE SIMP(SUM,LOWER,UPPER,SIGMA,KLOW,KUP,PIECES,YIELD,STAR,
      1PYTCH,CALPHA,BETA,DOL)
       COMMON/LYIA1/PI,Q,TOL,W,B,T,R,
      1OUE,PEA,BEE,DEE,DKSTAR,TOUGH,DKCRIT,
      2TRIG,LOWW,DAGER,NSKIP
       REAL*4 K,N,LOWER,KLOW,KUP,LOWW
       INTEGER*4 PIECES
       PIECES=2
       X=LOWER/T
       CALL CT(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
       IF (X.EQ.LOWW/T)
      1CALL CAL(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
       KLOW=K
       IF (X.EQ.LOWW/T) CALL CPYTCH(K,PYTCH)
       IF (NSKIP.NE.1) UPPER=LOWER+PYTCH
       ESUM=F2
       DELTA=(UPPER-LOWER)/PIECES
       EVSUM=0
       X=(LOWER+DELTA)/T
       CALL CT(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
       ODSUM=F2
       X=UPPER/T
       CALL CT(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
```

- 85 -

```
              KUP=K
              ESUM=ESUM+F2
              SUM=(ESUM+4*ODSUM)*DELTA/3
      1600    PIECES=PIECES*2
              SUM1=SUM
              DELTA=(UPPER-LOWER)/PIECES
              EVSUM=EVSUM+ODSUM
              ODSUM=0
              L=IFIX(FLOAT(PIECES)/2)
              DO 1610 I=1,10000
              Z=LOWER+DELTA*(2*I-1)
              X=Z/T
              CALL CT(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
              ODSUM=ODSUM+F2
      1610    IF (I.EQ.L) GO TO 1620
      1620    SUM=(ESUM+4*ODSUM+2*EVSUM)*DELTA/3
              IF (ABS(SUM-SUM1).GT.ABS(TOL*SUM)) GO TO 1600
              RETURN
              END
      C       SUBROUTINE FOR CALCULATING CALPHA FOR THE CT RETARDATION
      C       MODEL (20% AND 50% OVERLOAD CASES)
              SUBROUTINE CAL(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
              COMMON/LYIA1/PI,Q,TOL,W,B,T,R,
             1QUE,PEA,BEE,DEE,DKSTAR,TOUGH,DKCRIT,
             2TRIG,LOWW,DAGER,NSKIP
              REAL*4 K,N,LOWW
              IF (DAGER.EQ.0.) CALPHA=0.
              IF (DAGER.EQ.0.) GO TO 1679
              SK=K/DKSTAR
              CAS=1-1/SK
              IF (DAGER.EQ.50) GO TO 1675
              BETA=(SQRT(2.0)*PI**2*YIELD**2)/(.44*K**2)
              CACAS=(-.730791E-01)*SK**3 + (.303086)*SK**2 -
             1 (.422108)*SK + 0.117517E01
              GO TO 1677
      1675    BETA=(SQRT(2.0)*PI**2*YIELD**2)/(1.25*K**2)
              CACAS=(-.121127E-01)*SK**2 + (.239231E-01)*SK + 0.987133
      1677    CALPHA=CACAS*CAS
      1679    DOL=1
              CALL CT(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
              RETURN
              END
      C       SUBROUTINE FOR COMPACT TENSION SPECIMENS
      C       IN THIS ROUTINE, "SIGMA" IS A LOAD, NOT A STRESS!!!
              SUBROUTINE CT(SIGMA,X,K,F2,YIELD,STAR,PYTCH,CALPHA,BETA,DOL)
              COMMON/LYIA1/PI,Q,TOL,W,B,T,R,
             1QUE,PEA,BEE,DEE,DKSTAR,TOUGH,DKCRIT,
             2TRIG,LOWW,DAGER,NSKIP
              REAL*4 K,M1,M2,M3,N,LOWW
              ALPHA=X*T/W
              M1=SIGMA/(B*(SQRT(W)))
              M2=(2+ALPHA)/((1-ALPHA)**1.5)
```

END

FILMED

DTIC
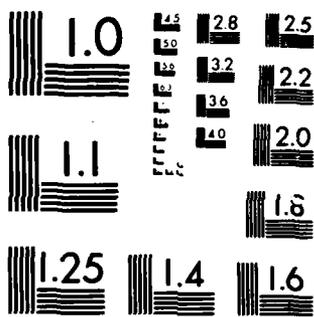
MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

```
             M3=0.886+(4.64*ALPHA)-(13.32*(ALPHA**2))+
            1(14.72*(ALPHA**3))-(5.6*(ALPHA**4))
             K=M1*M2*M3*(1.0-R)
             IF ((X.EQ.LOWW/T) .AND. (DOL.EQ.0)) GO TO 1760
             DELA=X*T-LOWW
             STAR=CALPHA*EXP(-BETA*DELA)
             EFFK=K*(1-STAR)
             K=EFFK
       1760  CALL FINDF2(K,F2)
             RETURN
             END
             SUBROUTINE FINDF2(K,F2)
       C     SUBROUTINE SELECTING F2 BASED MODIFIED SIGMODAL EQN ONLY
             REAL*4 K,N,LOWW
             INTEGER*4 CINH
             COMMON/LYIA1/PI,Q,TOL,W,B,T,R,
            1QUE,PEA,BEE,DEE,DKSTAR,TOUGH,DKCRIT,
            2TRIG,LOWW,DAGER,NSKIP
       C *************************************************
             CINH=2
       C ** USE MODIFIED SIGMODAL EQN ONLY CINH=2 *******
             IF (CINH.NE.2) GO TO 1840
             IF (K.GT.DKSTAR) GO TO 1830
             WRITE (5,1800) K
             WRITE (5,1810) DKSTAR
             WRITE (5,1820)
       1800  FORMAT (' YOUR INITIAL DELTA-K IS ',F5.2,',AND THIS IS SMALL-')
       1810  FORMAT (' ER THAN YOUR DELTA-K THRESHOLD OF ',F5.2,'. THIS')
       1820  FORMAT (' WILL GIVE THE MSE INDIGESTION. TRY AGAIN.')
             CALL EXIT
             STOP
       1830  F2=1.0/(EXP(BEE)*((K/DKSTAR)**PEA)*((ALOG10(
            1K/DKSTAR))**QUE)*((ALOG10(DKCRIT/K))**DEE))
       1840  IF (CINH.EQ.1) F2=1.0/(10.0**(S1*(SINH(S2*((ALOG10(K))+S3)))
            1+S4))
             IF (CINH.EQ.0) F2=(K**(-N))/C
             RETURN
             END
       C ***** SUBROUTINE TO CALCULATE PYTCH INCREMENT FOR SIMP *****
             SUBROUTINE CPYTCH(K,PYTCH)
             REAL*4 K
             COMMON/LYIA1/PI,Q,TOL,W,B,T,R,
            1QUE,PEA,BEE,DEE,DKSTAR,TOUGH,DKCRIT,
            2TRIG,LOWW,DAGER,NSKIP
             IF (K .LE. 22.77) PYTCH=.00001
             IF (22.77 .LT. K .AND. K .LE. 30.0)
            1PYTCH=(.002*K-.045)/53.5
             IF (30.0 .LT. K)
            1PYTCH=(.0004*K-.01)/7.0
             RETURN
             END
```

Sample Input

```
TYPE I502T.DAT
30 SEPT 85 SPECIMEN 84-502, EE3 RLH
SPECIMEN  84-502, EE3 AT 1200 DEG F
272.73,.4087,0.,120.
-8.69,21.00,-1.1,1.8,-1.8
.394,2.740,1.5736
0.0,0.0
1.426,20.0
32.888,20.0
44.707,20.0
```

Appendix C.

Modified CRACKS Program

```
      PROGRAM CRACKS4(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE1)000100
      COMMON/DATA/ EQN,NASA,J1PR,J2PR,J3PR,J4PR,J5PR,AZERO,AMAX,NZERO  000110
      INTEGER EQN                                                     000120
      REAL NZERO                                                      000130
      COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 000140
      INTEGER RETARD,PLSTRN                                           000150
      COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC, 000160
     /              KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX              000170
      REAL KSUBC,KSUBQ                                                000180
      COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10),000190
     /             NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS               000200
      COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                   000210
      COMMON/CORFAC/ ISURF,RATIO,PHI,THICK,IBETA(10),BETA(10),NPTS,   000220
     /               AOVERB(100),BTABLE(100),NPTS2,AOVRB2(100),       000230
     /               BTABL2(100),ASTART(10),ASTOP(10)                 000240
      COMMON/OUTPOT/KMAX,KMAXA,DELTAK,IFLT,DADNPR                     000250
      REAL KMAX,KMAXA                                                 000260
      WRITE(6,3000)                                                   000270
      IRSTRT = 0                                                      000280
      MOD1 = 0                                                        000290
      RTARD1 = 0                                                      000300
      ICASE = 0                                                       000310
    1 ICASE = ICASE + 1                                               000320
      ISPEC=0                                                         000330
    5 IFLT=0                                                          000340
      RETARD = 0                                                      000350
      JSTART=1                                                        000360
      OVLD = 0.0                                                      000370
      IF(RTARD1.EQ.0) GO TO 10                                        000380
      NORTRD = 0                                                      000390
      MODEL = MOD1                                                    000400
   10 ISPEC = ISPEC + 1                                               000410
      CALL INPUT(ICASE,ISPEC,IRSTRT)                                  000420
      IF(ISTOP.NE.0) GO TO 1                                          000430
      IF(IRSTRT.GE.4) GO TO 1150                                      000440
   20 A = AZERO                                                       000450
      CYC = NZERO                                                     000460
      ABLK = AZERO                                                    000470
      DO 1000  J1=JSTART,NBLKS                                        000480
      DO 900 J2=1,NSEGS                                               000490
      ISEG = ISEGS(J2)                                                000500
      IBLK = IBLKS(J2)                                                000510
      NLYR = NLYRS(ISEG)                                              000520
      DO 800 J3=1,IBLK                                                000530
      IFLT = IFLT + 1                                                 000540
      DO 700 J4=1,NLYR                                                000550
```

```
                  DN = CYCLES(J4,ISEG)                            000560
                  CYCSVE = CYC                                    000570
            500 CALL GRWCRK(CYC,A,DN)                             000580
                  IF(A.LT.AMAX) GO TO 550                         000590
                  WRITE(6,3600) A,AMAX,IFLT                       000600
                  GO TO 1100                                      000610
            550 IF (ISTOP.NE.2) GO TO 600                         000620
                  DN = CYCSVE + DN - CYC                          000630
                  ISTOP = 0                                       000640
                  ISURF = 0                                       000650
                  IF(DN.GT.0.0) GO TO 500                         000660
            600 IF(ISTOP.NE.0) GO TO 1100                         000670
                  IF(IFLT.EQ.1) CALL OUTPUT(CYC,A)                000680
                  IF(J4PR.EQ.0) GO TO 700                         000690
                  IF(MOD(IFLT,J4PR).EQ.0) CALL OUTPUT(CYC,A)      000700
            700 CONTINUE                                          000710
                  IF(J3PR.EQ.0) GO TO 800                         000720
                  IF(MOD(IFLT,J3PR).NE.0) GO TO 800               000730
                  WRITE(6,3050) IFLT,ISEG,A                       000740
            800 CONTINUE                                          000750
                  IF(J2PR.EQ.0) GO TO 900                         000760
                  IF(MOD(J2,J2PR).NE.0) GO TO 900                 000770
                  WRITE(6,3100) J2,J1,A                           000780
            900 CONTINUE                                          000790
                  IF(IRSTRT.EQ.3) CALL RESTRT(CYC,A,IRSTRT)       000800
                  IF(J1PR.EQ.0) GO TO 1000                        000810
                  IF(MOD(J1,J1PR).NE.0) GO TO 1000               000820
                  DELA = A - ABLK                                 000830
                  GROWTH = A - AZERO                              000840
                  ABLK = A                                        000850
                  WRITE(6,3200) J1,A,DELA,GROWTH                  000860
           1000 CONTINUE                                          000870
                  GROWTH = A - AZERO                              000880
                  WRITE(6,3500) CYC,A,GROWTH                      000890
           1100 IF(NORTRD.EQ.0) GO TO 1300                        000900
           1150 ICHECK = IRSTRT + 1                               000910
                  GO TO (1,5,10,1,1200,1250),ICHECK              000920
           1200 CALL RESTRT(CYC,A,IRSTRT)                         000930
                  AZERO = A                                       000940
                  NZERO = CYC                                     000950
                  JSTART = J1 - 1                                 000960
                  GO TO 20                                        000970
           1250 CALL RESTRT(CYC,A,IRSTRT)                         000980
                  AZERO = A                                       000990
                  NZERO = CYC                                     001000
                  JSTART = J1 + 1                                 001010
                  NBLKS = NBLKS + J1                              001020
                  GO TO 20                                        001030
           1300 ISPEC = ISPEC + 1                                 001040
                  WRITE(6,2000)                                   001050
                  WRITE(6,2900) ICASE,ISPEC                       001060
                  WRITE(6,3300)                                   001070
```

```
      RTARD1 = 1                                                        001080
      MOD1 = MODEL                                                      001090
      MODEL = 0                                                         001100
      ISTOP = 0                                                         001110
      RETARD = 0                                                        001120
      NORTRD = 1                                                        001130
      OVLD = 0.0                                                        001140
      IFLT = 0                                                          001150
      GO TO 20                                                          001160
 2000 FORMAT(1H1)                                                       001170
 2900 FORMAT(1H1,70(1H*)/26X,5HCASE ,I2,4HRUN ,I2/1X,70(1H*))           001180
 3000 FORMAT(1H1, 70(1H$)/1X, 70(1H$)//1X,29(1H*),12H CRACKS IV  29(1H*)001190
     1// 20X,9HVERSION 6,12X,9H09/21/79 /30X,13HR.M.ENGLE JR./18X,       001200
     2    36HAIR FORCE FLIGHT DYNAMICS LABORATORY//26X,10HAFFDL(FBE)/    001210
     3    26X,19HATTN(R.M.ENGLE JR.)/26X,17HW-PAFB,OHIO 45433//          001220
     4    26X,12H513-255-6104//1X, 70(1H$)/1X, 70(1H$) )                001230
 3050 FORMAT(15H END OF FLIGHT I5,9H  MISSION,I2,7X,14HCRACK LENGTH =,F  001240
     /10.5)                                                             001250
 3100 FORMAT(15H END OF SEGMENT,I6,9H OF BLOCK,I5,5X,14HCRACK LENGTH =,  001260
     /F10.5)                                                            001270
 3200 FORMAT(14H0 END OF BLOCK,I5,20X,14HCRACK LENGTH =,F10.5/          001280
     /5X,19HGROWTH THIS BLOCK =,F10.5,5X,14HTOTAL GROWTH =,F10.5//)     001290
 3300 FORMAT(///1X, 70(1H*)/24X,33HRERUN OF CASE WITH NO RETARDATION/   001300
     /       1X, 70(1H*))                                              001310
 3500 FORMAT(1H0, 70(1H*)/ 5X,14HTOTAL CYCLES =,F12.2,5X,20HFINAL CRACK 001320
     /LENGTH =,F10.5/22X,20HTOTAL CRACK GROWTH =,F10.5/1X, 70(1H*)/1H1) 001330
 3600 FORMAT(1H0, 70(1H*)/ 1X,20HCURRENT CRACK LENGTHF10.5/31H EXCEEDS A001340
     /LLOWABLE CRACK LENGTH,F10.5/9X17H IN FLIGHT NUMBER,I6/1X, 70(1H*))001350
      END                                                              001360
      SUBROUTINE OUTPUT(CYC,A)                                         001370
C                                                                      001380
      COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 001390
      INTEGER RETARD,PLSTRN                                            001400
      COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10),  001410
     /             NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS                 001420
      COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,    001430
     /             KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX                 001440
      COMMON/DATA/EQN,NASA,J1PR,J2PR,J3PR,J4PR,J5PR,AZERO,AMAX,NZERO
      REAL KSUBC,KSUBQ                                                 001450
      COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                    001460
      COMMON/OUTPOT/KMAX,KMAXA,DELTAK,IFLT,DADNPR                      001470
      REAL KMAX,KMAXA,KMXEFF                                           001480
      DATA NOP/0/
      IF(RETARD.EQ.0) GO TO 100                                        001490
C                                                                      001500
C     OBTAIN OUTPUT QUANTITIES FOR RETARDATION                         001510
C                                                                      001520
      GO TO (10,20,30,40,50), MODEL                                    001530
   10 CALL WHELER(CYC,A,DDNRET)                                        001540
      GO TO 90                                                         001550
   20 CALL WLNBRG(CYC,A,DDNRET)                                        001560
      GO TO 90                                                         001570
```

```
   30 DDNRET = DADNPR                                                001580
      GO TO 90                                                       001590
   40 CONTINUE                                                       001600
   50 CONTINUE                                                       001610
   90 KMXEFF = KMAX                                                  001620
      DLKEFF = DELTAK                                                001630
C                                                                    001640
C     DETERMINE UNRETARDED QUANTITIES CORRESPONDING TO CURRENT CRACK 001650
C     LENGTH                                                         001660
C                                                                    001670
  100 CONTINUE                                                       001680
      SIGMAX = SMAX(J4,ISEG)                                         001690
      SIGMIN = SMIN(J4,ISEG)                                         001700
      CALL RATE(CYC,A,DADN)                                          001710
      IF(MODEL.NE.0.AND.RETARD.NE.0) GO TO 200                       001720
      KMXEFF = KMAX                                                  001730
      DLKEFF = DELTAK                                                001740
      DDNRET = DADN                                                  001750
C                                                                    001760
C     PRINT HEADER AT TOP OF EACH PAGE                               001770
C                                                                    001780
  200 CONTINUE                                                       001790
      IF(J4.EQ.1) WRITE(6,2000)                                      001800
      IF(J4.EQ.1 .AND. NOP.EQ.0) WRITE(1,2000)
      IF(J4.EQ.1 .AND. NOP.EQ.0) WRITE(1,2001)NZERO,AZERO
 2001 FORMAT(16X,F10.1,1X,F9.6)
      NOP=NOP+1
      RETFAC = DDNRET/DADN                                           001810
      IF(DADN .EQ. 0.0) RETFAC = 0.0                                 001820
      WRITE(6,2100) IFLT,ISEG,J4,CYC,A,DLKEFF,KMXEFF,DDNRET,RETFAC   001830
      WRITE(1,2100) IFLT,ISEG,J4,CYC,A,DLKEFF,KMXEFF,DDNRET,RETFAC
      RETURN                                                         001840
 2000 FORMAT(2X*FLT   MSN  LYR    CYCLES*7X*A*4X                     001850
     /*DELTA K   K MAX*5X*DA/DN     RETARD* )
 2100 FORMAT(I5,2X,I3,2X,I3,1X,F10.1,1X,F9.6,1X,                     001870
     /F7.2,1X,F7.2,1X,E10.3,1X,F7.3)                                 001880
      END                                                            001890
      SUBROUTINE INPUT(ICASE,ISPEC,IRSTRT)                           001900
C                                                                    001910
C     READS LABELED SECTIONS OF DATA DECK IN ANY ORDER.             001920
C     PRINTS OUT PROBLEM AND SOLUTION DESCRIPTION.                   001930
C                                                                    001940
      COMMON/DATA/ EQN,NASA,J1PR,J2PR,J3PR,J4PR,J5PR,AZERO,AMAX,NZERO 001950
      INTEGER EQN                                                    001960
      REAL NZERO                                                     001970
      COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 001980
      INTEGER RETARD,PLSTRN                                          001990
      COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC, 002000
     /              KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX             002010
      REAL KSUBC,KSUBQ                                               002020
      COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10), 002030
     /             NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS              002040
```

```
      COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD              002050
      COMMON/CORFAC/ ISURF,RATIO,PHI,THICK,IBETA(10),BETA(10),NPTS,   002060
     /             AOVERB(100),BTABLE(100),NPTS2,AOVRB2(100),    002070
     /             BTABL2(100),ASTART(10),ASTOP(10)             002080
      DIMENSION CARD(18),TITLE(18),SEGTTL(18,50)                002090
      REAL LODLAB(3)                                            002100
      REAL LABEL(3)                                             002110
      REAL SPCTRM(3),ENDLDS(3),IEND(3),LTITLE(3)                002120
      REAL TAG(2)                                               002130
      REAL EQNS(3),MATL(3),LIMITS(3),ANAL(3),LOADS(3),END(3)    002140
      REAL DADN(3),WALK(3)                                      002150
      REAL EQUAT(3),DELK(3)                                     002160
      REAL FORMAN(3),PARIS(3),NASAL(3),BLANK(3),SIGMOID(3)
      REAL SURF(3),RET(3),BETAL(3),BETAE(3),SIGMAS(3),DELTAS(3) 002180
      REAL AMEANS(3),ENDSPC(3),TRSHLD(3),LRSTRT(3),KPRT(3)      002190
C     THE NEXT 3 CARDS ADDED FOR CLOSUR                         002200
      COMMON/CLOS/CF,CCOEF,CFEXP,B,BOL,NSAT                     002210
      REAL NSAT                                                 002220
      COMMON/CLOSIC/SC,SPEAK,PRVMX,APEAK,SC1,SC2,SC3,PRVMN      002230
      DATA SPCTRM  /4HSPEC,4HTRUM,1H /,ENDLDS  /3HEND,4HLOAD,1HS/   002240
     +    ,IEND   /3HEND,4HDATA,1H /,LTITLE   /4HTITL,1HE,1H /  002250
      DATA EQNS    /4HEQUA,4HTION,1H /,MATL   /4HMATE,4HRIAL,1H /   002260
     +    ,LIMITS  /4HLIMI,2HTS,1H /,ANAL    /4HANAL,4HYSIS,1H /    002270
     +    ,LOADS   /4HLOAD,1HS,1H /,END     /3HEND,2*1H /     002280
      DATA DADN   /4HDA/D,1HN,1H /,WALK    /4HWALK,2HER,1H /   002290
      DATA FORMAN   /4HFORM,2HAN,1H /,PARIS    /4HPARI,1HS,1H /    002300
     +    ,NASAL  /4HNASA,2*1H /,BLANK   /3*1H /,SIGMOID/4HSIGM,3HOID,1
     +H /
      DATA SURF   /4HSURF,3HACE,1H /,RET   /4HRETA,2HRD,1H /   002320
     +    ,BETAL  /4HBETA,2*1H /,BETAE   /3HEND,2*1H /         002330
     +    ,SIGMAS  /4HMAX-,3HMIN,1H /,DELTAS   /4HR-DE,4HLTA ,1H /  002340
      DATA AMEANS   /4HMEAN,4H-ALT,1H /,ENDSPC   /3HEND,4HSPEC,1HR/   002350
     +    ,TRSHLD   /4HTHRE,4HSHOL,1HD /,LRSTRT   /4HREST,3HART,1H /   002360
     +    ,KPRT   /4HPRIN,1HT,1H /                           002370
      ISTOP = 0                                                 002380
      IF(ISPEC.GT.1) GO TO 6                                    002390
C       INITIAL CONDITIONS FOR CLOSUR                           002400
        SC = 0.                                                 002410
        SC1 = 0.                                                002420
        SC2 = 0.                                                002430
        SC3 = 0.                                                002440
        SPEAK = 0.                                              002450
        PRVMX = 0.                                              002460
        PRVMN = 0.                                              002470
        APEAK = 0.                                              002480
C       END OF INITIAL CONDITIONS FOR CLOSUR                    002490
      IRSTRT = 0                                                002500
      NORTRD = 1                                                002510
      EQN = 1                                                   002520
      NASA=0                                                    002530
      NPTS=0                                                    002540
      NPTS2=0                                                   002550
```

```
      ISURF=0                                                    002560
      MODEL=0                                                    002570
      J1PR = 1                                                   002580
      J2PR = 0                                                   002590
      J3PR = 0                                                   002600
      J4PR = 0                                                   002610
      J5PR = 0                                                   002620
      RATIO = 1.0                                                002630
      RCUT = 1.                                                  002640
      KSUBC = 68000.                                             002650
      KSUBQ = KSUBC                                              002660
      C = 5.0E-13                                                002670
      SMALLN = 3.0                                               002680
      DELKTH = 0.                                                002690
      RMULT = 0.                                                 002700
      DO 5 J=1,10                                                002710
      BETA(J) = 0.0                                              002720
    5 IBETA(J) = 0.0                                             002730
    6 READ(5,1000) LABEL                                         002740
      IF(EOF(5)) 9999,7                                          002750
    7 WRITE(6,2900) ICASE,ISPEC                                  002760
      IF(IRSTRT.NE.0) WRITE(6,3400)                              002770
      GO TO 10                                                   002780
    1 READ(5,1000)LABEL                                          002790
      IF(EOF(5)) 9998,10                                         002800
   10 IF( LABEL(1) .EQ. LTITLE(1) .AND. LABEL(2) .EQ. LTITLE(2)  002810
     +                  .AND. LABEL(3) .EQ. LTITLE(3) ) GO TO 100  002820
      IF( LABEL(1) .EQ. KPRT(1) .AND. LABEL(2) .EQ. KPRT(2)      002830
     +                  .AND. LABEL(3) .EQ. KPRT(3) ) GO TO 150    002840
      IF( LABEL(1) .EQ. EQNS(1) .AND. LABEL(2) .EQ. EQNS(2)      002850
     +                  .AND. LABEL(3) .EQ. EQNS(3) ) GO TO 200    002860
      IF( LABEL(1) .EQ. MATL(1) .AND. LABEL(2) .EQ. MATL(2)      002870
     +                  .AND. LABEL(3) .EQ. MATL(3) ) GO TO 300    002880
      IF( LABEL(1) .EQ. TRSHLD(1) .AND. LABEL(2) .EQ. TRSHLD(2)  002890
     +                  .AND. LABEL(3) .EQ. TRSHLD(3) ) GO TO 350  002900
      IF( LABEL(1) .EQ. LIMITS(1) .AND. LABEL(2) .EQ. LIMITS(2)  002910
     +                  .AND. LABEL(3) .EQ. LIMITS(3) ) GO TO 400  002920
      IF( LABEL(1) .EQ. ANAL(1) .AND. LABEL(2) .EQ. ANAL(2)      002930
     +                  .AND. LABEL(3) .EQ. ANAL(3) ) GO TO 500    002940
      IF( LABEL(1) .EQ. LOADS(1) .AND. LABEL(2) .EQ. LOADS(2)    002950
     +                  .AND. LABEL(3) .EQ. LOADS(3) ) GO TO 600   002960
      IF( LABEL(1) .EQ. SPCTRM(1) .AND. LABEL(2) .EQ. SPCTRM(2)  002970
     +                  .AND. LABEL(3) .EQ. SPCTRM(3) ) GO TO 660  002980
      IF( LABEL(1) .EQ. LRSTRT(1) .AND. LABEL(2) .EQ. LRSTRT(2)  002990
     +                  .AND. LABEL(3) .EQ. LRSTRT(3) ) GO TO 680  003000
      IF( LABEL(1) .EQ. IEND(1) .AND. LABEL(2) .EQ. IEND(2)      003010
     +                  .AND. LABEL(3) .EQ. IEND(3) ) GO TO 700    003020
      WRITE(6,9020) LABEL                                        003030
      ISTOP = 1                                                  003040
      GO TO 1                                                    003050
  100 READ(5,1010) NTITLE                                        003060
      DO 110 I=1,NTITLE                                          003070
```

```
      READ(5,1000) CARD                                              003080
      WRITE(6,2005) CARD                                             003090
      WRITE(1,2005) CARD
  110 CONTINUE                                                       003100
      GO TO 1                                                        003110
  150 READ(5,1010) J1PR,J2PR,J3PR,J4PR,J5PR                          003120
      GO TO 1                                                        003130
  200 READ(5,1002) EQUAT,DELK                                        003140
      IF( EQUAT(1) .EQ. FORMAN(1) .AND. EQUAT(2) .EQ. FORMAN(2)      003150
     +                    .AND. EQUAT(3) .EQ. FORMAN(3) )EQN = 1     003160
      IF( EQUAT(1) .EQ. PARIS(1) .AND. EQUAT(2) .EQ. PARIS(2)        003170
     +                    .AND. EQUAT(3) .EQ. PARIS(3) )EQN = 2      003180
      IF( EQUAT(1) .EQ. DADN(1) .AND. EQUAT(2) .EQ. DADN(2)          003190
     +                    .AND. EQUAT(3) .EQ. DADN(3) )EQN = 3       003200
      IF( EQUAT(1) .EQ. WALK(1) .AND. EQUAT(2) .EQ. WALK(2)          003210
     +                    .AND. EQUAT(3) .EQ. WALK(3) )EQN = 4       003220
      IF( EQUAT(1) .EQ. SIGMOID(1) .AND. EQUAT(2) .EQ. SIGMOID(2)
     +                    .AND. EQUAT(3) .EQ. SIGMOID(3) )EQN = 5
      IF( DELK(1) .EQ. NASAL(1) .AND. DELK(2) .EQ. NASAL(2)          003230
     +                    .AND. DELK(3) .EQ. NASAL(3) )NASA = EQN    003240
      IF(NASA.NE.0) GO TO 260                                        003250
      GO TO (210,220,230,240,250),EQN                                003260
  210 WRITE(6,2010)                                                  003270
      GO TO 1                                                        003280
  220 WRITE(6,2020)                                                  003290
      GO TO 1                                                        003300
  230 WRITE(6,3030)                                                  003310
      GO TO 1                                                        003320
  240 CONTINUE                                                       003330
      WRITE(6,3050)                                                  003340
      GO TO 1                                                        003350
  250 CONTINUE                                                       003360
      WRITE(6,3070)
      GO TO 1                                                        003370
  260 GO TO (270,280,285,290,295), NASA                              003380
  270 WRITE(6,2030)                                                  003390
      GO TO 1                                                        003400
  280 WRITE(6,2040)                                                  003410
      GO TO 1                                                        003420
  285 WRITE(6,3040)                                                  003430
      GO TO 1                                                        003440
  290 CONTINUE                                                       003450
      WRITE(6,3060)                                                  003460
      GO TO 1                                                        003470
  295 CONTINUE                                                       003480
  300 READ(5,1000) MATID                                             003490
      WRITE(6,2050) MATID                                            003500
      CALL CNDIN(EQN)                                                003510
      GO TO 1                                                        003520
  350 READ(5,1020) DELKTH,RMULT                                      003530
      WRITE(6,2260) DELKTH,RMULT                                     003540
      GO TO 1                                                        003550
```

```
400 READ(5,1020) AZERO,AMAX,NZERO,RCUT                                  003560
    IF(RCUT.EQ.0.) RCUT =1.                                            003570
    IF(AMAX.EQ.0.) GO TO 410                                          003580
    WRITE(6,2070) AZERO,AMAX,NZERO                                    003590
    WRITE(6,2075) RCUT                                                003600
    GO TO 1                                                           003610
410 AMAX=1.0E+50                                                      003620
    WRITE(6,2080) AZERO,NZERO                                         003630
    WRITE(6,2075) RCUT                                                003640
    GO TO 1                                                           003650
500 READ(5,1070) LABEL,CON1,CON2,CON3,CON4,CON5,CON6                  003660
    IF( LABEL(1) .EQ. SURF(1) .AND. LABEL(2) .EQ. SURF(2)            003670
   +                     .AND. LABEL(3) .EQ. SURF(3) ) GO TO 510     003680
    IF( LABEL(1) .EQ. RET(1) .AND. LABEL(2) .EQ. RET(2)             003690
   +                     .AND. LABEL(3) .EQ. RET(3) ) GO TO 520      003700
    IF( LABEL(1) .EQ. BETAL(1) .AND. LABEL(2) .EQ. BETAL(2)         003710
   +                     .AND. LABEL(3) .EQ. BETAL(3) ) GO TO 530    003720
    IF( LABEL(1) .EQ. BETAE(1) .AND. LABEL(2) .EQ. BETAE(2)         003730
   +                     .AND. LABEL(3) .EQ. BETAE(3) ) GO TO 1      003740
    WRITE(6,9020) LABEL                                               003750
    ISTOP = 1                                                         003760
    GO TO 1                                                           003770
510 ISURF=1                                                          003780
    CZERO=CON1                                                        003790
    THICK=CON2                                                        003800
    RATIO=AZERO/(2.*CZERO)                                            003810
    SMALLK = SQRT((CZERO**2 - AZERO**2) / CZERO**2)                  003820
    CKSQD = 1.0 - SMALLK**2                                          003830
    CALL CELI2(PHI,SMALLK,1.0,CKSQD,IER)                             003840
    WRITE(6,2090) RATIO,THICK,PHI                                    003850
    IF(IER.EQ.0) GO TO 500                                           003860
    WRITE(6,9030)                                                    003870
    ISTOP = 1                                                         003880
    GO TO 500                                                        003890
520 MODEL=CON1+0.5                                                   003900
    PLSTRN=CON2+1.5                                                  003910
    NORTRD = CON4 + 0.5                                             003920
    OVLD = CON5                                                      003930
    ASUBP = CON6                                                     003940
    IF(NORTRD.NE.0) WRITE(6,2170)                                   003950
    GO TO (521,522,523,524,525),MODEL                              003960
521 SMALLM=CON3                                                     003970
    WRITE(6,2100) SMALLM                                            003980
    GO TO 526                                                       003990
522 WRITE(6,2105)                                                   004000
    OLMAX = CON3                                                    004010
    IF(OLMAX.NE.0.) WRITE(6,2106) OLMAX                            004020
    GO TO 526                                                       004030
523 CONTINUE                                                        004040
    WRITE(6,2107)                                                   004050
    READ(5,1020) CF,CCOEF,CFEXP,B,BOL,NSAT                         004060
    WRITE(6,2108) CCOEF,CF,CCOEF,CFEXP,B,BOL,NSAT                  004070
```

```
          RETARD = 1                                              004080
            GO TO 526                                             004090
524 CONTINUE                                                      004100
525 WRITE(6,3000)                                                 004110
    MODEL=2                                                       004120
526 GO TO (527,528),PLSTRN                                        004130
527 WRITE(6,2180)                                                 004140
    PLSTRN = PLSTRN - 1                                           004150
    GO TO 500                                                     004160
528 WRITE(6,2190)                                                 004170
    PLSTRN = PLSTRN - 1                                           004180
    GO TO 500                                                     004190
530 I = CON1 + 0.5                                                004200
    IBETA(I)=I                                                    004210
    GO TO(531,532,533,534,535,536,537,538,539),I                 004220
531 BETA(I)=CON2                                                  004230
    ASTART(I)=CON3                                                004240
    ASTOP(I)=CON4                                                 004250
    IF(ASTART(I).EQ.0.) ASTART(I) = AZERO                        004260
    IF(ASTOP(I).EQ.0.) ASTOP(I) = 1.E50                          004270
    WRITE(6,2110) BETA(1),ASTART(I),ASTOP(I)                     004280
    GO TO 500                                                     004290
532 BETA(I)=CON2                                                  004300
    ASTART(I)=CON3                                                004310
    ASTOP(I)=CON4                                                 004320
    IF(ASTART(I).EQ.0.) ASTART(I) = AZERO                        004330
    IF(ASTOP(I).EQ.0.) ASTOP(I) = 1.E50                          004340
    WRITE(6,2120) BETA(2),ASTART(I),ASTOP(I)                     004350
    GO TO 500                                                     004360
533 BETA(I)=CON3                                                  004370
    NPTS=CON2+0.5                                                 004380
    ASTART(I)=CON4                                                004390
    ASTOP(I)=CON5                                                 004400
    IF(ASTART(I).EQ.0.) ASTART(I) = AZERO                        004410
    IF(ASTOP(I).EQ.0.) ASTOP(I) = 1.E50                          004420
    IF(NPTS.LE.0.OR.NPTS.GT.100) GO TO 540                       004430
    READ(5,1000) CARD                                            004440
    READ(5,1040) (AOVERB(J),BTABLE(J),J=1,NPTS)                  004450
    WF...E(6,2130) BETA(I),ASTART(I),ASTOP(I),CARD,              004460
   1              (AOVERB(J),BTABLE(J),J=1,NPTS)                  004470
    GO TO 500                                                     004480
534 BETA(I)=CON3                                                  004490
    NPTS2=CON2+0.5                                                004500
    ASTART(I)=CON4                                                004510
    ASTOP(I)=CON5                                                 004520
    IF(ASTART(I).EQ.0.) ASTART(I) = AZERO                        004530
    IF(ASTOP(I).EQ.0.) ASTOP(I) = 1.E50                          004540
    IF(NPTS2.LE.0.OR.NPTS2.GT.100) GO TO 540                     004550
    READ(5,1000) CARD                                            004560
    READ(5,1040) (AOVRB2(J),BTABL2(J),J=1,NPTS2)                 004570
    WRITE(6,2140) BETA(I),ASTART(I),ASTOP(I),CARD,               004580
   1              (AOVRB2(J),BTABL2(J),J=1,NPTS2)                 004590
```

```
      GO TO 500                                           004600
535 CONTINUE                                              004610
    BETA(I) = CON2                                        004620
    ASTART(I) = CON3                                      004630
    ASTOP(I) = CON4                                       004640
    IF(ASTART(I).EQ.0.0) ASTART(I) = AZERO                004650
    IF(ASTOP(I).EQ.0.0) ASTOP(I) = 1.0E50                 004660
    WRITE(6,2200) BETA(I),ASTART(I),ASTOP(I)              004670
    GO TO 500                                             004680
536 CONTINUE                                              004690
    BETA(I) = CON2                                        004700
    ASTART(I) = CON3                                      004710
    ASTOP(I) = CON4                                       004720
    IF(ASTART(I).EQ.0.0) ASTART(I) = AZERO                004730
    IF(ASTOP(I).EQ.0.0) ASTOP(I) = 1.0E50                 004740
    WRITE(6,2205) BETA(I),ASTART(I),ASTOP(I)              004750
    GO TO 500                                             004760
537 CONTINUE                                              004770
        BETA(I) = CON2                                    004780
        THICK = CON3                                      004790
        ASTART(I) = CON4                                  004800
        ASTOP(I) = CON5                                   004810
      IF ( ASTART(I) .EQ. 0.0 ) ASTART(I) = AZERO         004820
      IF ( ASTOP(I) .EQ. 0.0 ) ASTOP(I) = 1.E50           004830
    WRITE(6,2206) BETA(I),THICK,ASTART(I),ASTOP(I)        004840
    GO TO 500                                             004850
538 CONTINUE                                              004860
        BETA(I) = CON2                                    004870
        THICK = CON3                                      004880
        ASTART(I) = CON4                                  004890
        ASTOP(I) = CON5                                   004900
      IF ( ASTART(I) .EQ. 0.0 ) ASTART(I) = AZERO         004910
      IF ( ASTOP(I) .EQ. 0.0 ) ASTOP(I) = 1.E50           004920
    WRITE(6,2207) BETA(I),THICK,ASTART(I),ASTOP(I)        004930
    GO TO 500                                             004940
539 CONTINUE                                              004950
        BETA(I) = CON2
        THICK = CON3

        ASTART(I) = CON4
        ASTOP(I) = CON5
      IF ( ASTART(I) .EQ. 0.0 ) ASTART(I) = AZERO
      IF (ASTOP(I) .EQ. 0.0 ) ASTOP(I) = 1.E50
    WRITE(6,2208) BETA(I),THICK,ASTART(I),ASTOP(I)
    GO TO 500                                             004960
540 WRITE(6,9000) IBETA(I)                                004970
    IBETA(I)=0                                            004980
    GO TO 500                                             004990
600 READ(5,1050) NBLKS,LPRT,(TITLE(I),I=1,18)             005000
    WRITE(6,2000)(TITLE(I),I=1,18),NBLKS                  005010
    IF(LPRT.LT.0) GO TO 1                                 005020
    ISEG = 0                                              005030
```

```
      605 ISEG = ISEG + 1                                                  005040
          READ(5,1002) LODLAB,(SEGTTL(I,ISEG),I=1,18)                      005050
          LYR=1                                                            005060
          IF( LODLAB(1) .EQ. SIGMAS(1) .AND. LODLAB(2) .EQ. SIGMAS(2)      005070
         +                    .AND. LODLAB(3) .EQ. SIGMAS(3) ) GO TO 610   005080
          IF( LODLAB(1) .EQ. DELTAS(1) .AND. LODLAB(2) .EQ. DELTAS(2)      005090
         +                    .AND. LODLAB(3) .EQ. DELTAS(3) ) GO TO 620   005100
          IF( LODLAB(1) .EQ. AMEANS(1) .AND. LODLAB(2) .EQ. AMEANS(2)      005110
         +                    .AND. LODLAB(3) .EQ. AMEANS(3) ) GO TO 630   005120
          IF( LODLAB(1) .EQ. ENDLDS(1) .AND. LODLAB(2) .EQ. ENDLDS(2)      005130
         +                    .AND. LODLAB(3) .EQ. ENDLDS(3) ) GO TO 662   005140
          WRITE(6,9020) LODLAB                                             005150
          ISTOP = 1                                                        005160
          GO TO 1                                                          005170
      C                                                                    005180
      C     LOAD SPECTRUM INPUT AS MAX AND MIN STRESSES                    005190
      C                                                                    005200
      610 READ(5,1030) TAG,SMAX(LYR,ISEG),SMIN(LYR,ISEG),CYCLES(LYR,ISEG)  005210
          IF( TAG(1) .EQ. END(1) .AND. TAG(2) .EQ. END(2) ) GO TO 650      005220
      C                                                                    005230
      C     PRESENT PROGRAM DOES NOT CONSIDER COMPRESSIVE LOADS            005240
      C                                                                    005250
      C         EXCEPT FOR CLOSURE MODEL WHICH DOES                        005260
      C                                                                    005270
             IF ( MODEL .EQ. 3 ) GO TO 618                                 005280
          IF(SMAX(LYR,ISEG).LT.0.) SMAX(LYR,ISEG)=0.                       005290
          IF(SMIN(LYR,ISEG).LT.0.) SMIN(LYR,ISEG) = 0.                     005300
      618 LYR=LYR+1                                                        005310
          GO TO 610                                                        005320
      620 IF(LPRT.EQ.0)                                                    005330
         1WRITE(6,3010) ISEG,(SEGTTL(I,ISEG),I=1,18)                       005340
          IF(LPRT.EQ.0) WRITE(6,2210)                                      005350
      625 READ(5,1030) TAG,DELSIG,R,CYCLES(LYR,ISEG)                       005360
          IF( TAG(1) .EQ. END(1) .AND. TAG(2) .EQ. END(2) ) GO TO 650      005370
          IF(LPRT.EQ.0)                                                    005380
         1WRITE(6,2220) LYR,TAG,DELSIG,R,CYCLES(LYR,ISEG)                  005390
      C                                                                    005400
      C     PRESENT PROGRAM DOES NOT CONSIDER COMPRESSIVE LOADS            005410
      C                                                                    005420
          IF(DELSIG.LT.0.) DELSIG=0.                                       005430
          IF(R.LT.0.)R=0.                                                  005440
          SMAX(LYR,ISEG)=DELSIG/(1.-R)                                     005450
          SMIN(LYR,ISEG)=SMAX(LYR,ISEG)-DELSIG                             005460
          LYR=LYR+1                                                        005470
          GO TO 625                                                        005480
      630 IF(LPRT.EQ.0)                                                    005490
         1WRITE(6,3010) ISEG,(SEGTTL(I,ISEG),I=1,18)                       005500
          IF(LPRT.EQ.0) WRITE(6,2230)                                      005510
      635 READ(5,1030) TAG,SMEAN,SALT,CYCLES(LYR,ISEG)                     005520
          IF( TAG(1) .EQ. END(1) .AND. END(2) .EQ. TAG(2)) GO TO 650       005530
          IF(LPRT.EQ.0)                                                    005540
         1WRITE(6,2240) LYR,TAG,SMEAN,SALT,CYCLES(LYR,ISEG)                005550
```

```
C                                                                      005560
C       PRESENT PROGRAM DOES NOT CONSIDER COMPRESSIVE LOADS            005570
C                                                                      005580
        SMAX(LYR,ISEG)=SMEAN+SALT                                      005590
        IF(SMAX(LYR,ISEG).LT.0.) SMAX(LYR,ISEG)=0.                     005600
        SMIN(LYR,ISEG)=SMEAN-SALT                                      005610
        IF(SMIN(LYR,ISEG).LT.0.) SMIN(LYR,ISEG)=0.                     005620
        LYR=LYR+1                                                      005630
        GO TO 635                                                      005640
  650 NLYRS(ISEG) = LYR-1                                              005650
        GO TO 605                                                      005660
C                                                                      005670
C       READ IN COMPOSITION OF SPECTRUM                               005680
C                                                                      005690
  660 READ(5,1010)NSEGS,IPRT                                           005700
        READ(5,1060) (IBLKS(I),ISEGS(I),I=1,NSEGS)                     005710
        IF(IPRT.EQ.0) WRITE(6,2250) NSEGS                             005720
        IFLTS = 0                                                      005730
        DO 661 I=1,NSEGS                                               005740
        IFLTS = IFLTS + IBLKS(I)                                       005750
        IF(IPRT.EQ.0) WRITE(6,2255) I,IBLKS(I),ISEGS(I),IFLTS         005760
  661 CONTINUE                                                         005770
        GO TO 1                                                        005780
  662 NNSEG = ISEG - 1                                                 005790
        IF(LPRT.NE.0) GO TO 1                                          005800
        DO 670 ISEG =1,NNSEG                                           005810
        WRITE(6,2270) ISEG,(SEGTTL(I,ISEG),I=1,18)                     005820
        LYRS = NLYRS(ISEG)                                             005830
        TOTCYC = 0.0                                                   005840
        DO 665 J2=1,LYRS                                               005850
        TOTCYC = TOTCYC + CYCLES(J2,ISEG)                             005860
        WRITE(6,2280) J2,SMAX(J2,ISEG),SMIN(J2,ISEG),CYCLES(J2,ISEG), 005870
       1               TOTCYC                                          005880
  665 CONTINUE                                                         005890
  670 CONTINUE                                                         005900
        GO TO 1                                                        005910
  680 READ(5,1060) IRSTRT                                              005920
        ICHECK = IRSTRT + 1                                            005930
        GO TO (1,685,685,690,695,695),ICHECK                          005940
  685 WRITE(6,2150)                                                    005950
        GO TO 1                                                        005960
  690 WRITE(6,2160)                                                    005970
        GO TO 1                                                        005980
  695 WRITE(6,2165)                                                    005990
        GO TO 1.                                                       006000
  700 WRITE(6,2015) (TITLE(I),I=1,18)                                  006010
        RETURN                                                         006020
 9998 IF(IRSTRT .NE. 0) WRITE(6,9010)                                 006030
 9999 STOP                                                             006040
C                                                                      006050
 1000 FORMAT(20A4)                                                     006060
 1002 FORMAT(2A4,A2,17A4,A2)                                           006070
```

- 100 -

```
1010 FORMAT(16I5)                                                    006080
1020 FORMAT(8E10.0)                                                  006090
1030 FORMAT(A4,A1,7E10.0)                                            006100
1040 FORMAT(2E10.0)                                                  006110
1050 FORMAT(2I5,17A4,A2)                                             006120
1060 FORMAT(2I5)                                                     006130
1070 FORMAT(2A4,A2,7E10.0)                                           006140
2000 FORMAT(1H0, 1X,17A4,A2// 5X,I6,19H BLOCKS IN SPECTRUM)          006150
2005 FORMAT( 2X,17A4,A2)                                             006160
2010 FORMAT(1H0, 1X,50HCRACK PROPAGATION ANALYSIS USING FORMAN'S EQUATI006170
    1ON    / 4X,44HDA/DN=C*(DELTA K)**N/((1-R)*KSUBC-DELTA K)        006180
    2      / 5X,51HWHERE K IS OF THE FORM .... K=SIGMA*SQRT(PI*A)*BETA) 006190
2015 FORMAT(///1X,29(1H*),12HEND OF INPUT,29(1H*)/////1H1,1X,70(1H$)/  006200
    /1X,26(1H*),18HCRACKS IV ANALYSIS,26(1H*)/1X,17A4,A2/            006210
    /1X,70(1H$)//)                                                   006220
2020 FORMAT(1H0, 1X,50HCRACK PROPAGATION ANALYSIS USING PARIS' EQUATION006230
    1      /16X,21HDA/DN=C*(DELTA K)**N / 5X,51HWHERE K IS OF THE FORM . 006240
    2.. K=SIGMA*SQRT(PI*A)*BETA )                                    006250
2030 FORMAT(1H0, 1X,50HCRACK PROPAGATION ANALYSIS USING FORMAN'S EQUATI006260
    1ON    / 4X,44HDA/DN=C*(DELTA K)**N/((1-R)*KSUBC-DELTA K)        006270
    2      / 5X,51HWHERE K IS OF THE FORM .... K=SIGMA*SQRT(A)*BETA   ) 006280
2040 FORMAT(1H0, 1X,50HCRACK PROPAGATION ANALYSIS USING PARIS' EQUATION006290
    1      /16X,21HDA/DN=C*(DELTA K)**N / 5X,51HWHERE K IS OF THE FORM . 006300
    2.. K=SIGMA*SQRT(A)*BETA    )                                    006310
2050 FORMAT(1H0, 1X,20A4)                                            006320
2070 FORMAT(1H0, 1X,27HINITIAL HALF CRACK LENGTH = E16.8/2X,35HMAXIMUM 006330
    1HALF CRACK LENGTH ALLOWED = E16.8/1H0, 1X,22HINITIAL CYCLE NUMBER 006340
    2  ,F11.2 )                                                      006350
2075 FORMAT(1H0, 1X,11HR CUTOFF = ,F6.3)                             006360
2080 FORMAT(1H0,. 1X,27HINITIAL HALF CRACK LENGTH = E16.8 /1H0, 1X,22HIN006370
    1ITIAL CYCLE NUMBER = ,F11.2 )                                   006380
2090 FORMAT(1H0, 1X,34HSURFACE FLAW ANALYSIS WITH A/2C OF F5.2 / 5X,21H006390
    1MATERIAL THICKNESS IS ,F8.5/ 5X,19HSHAPE FACTOR PHI = ,F8.5)    006400
2100 FORMAT(1H0, 1X,41HWHEELER'S RETARDATION MODEL WITH SMALLM = ,F6.3)006410
2105 FORMAT(1H0,1X,*WILLENBORG RETARDATION MODEL*)                   006420
2106 FORMAT(1H0,1X,*GALLAGHER-MODIFIED WILLENBORG RETARDATION MODEL */ 006430
    2X*WHERE...*// 5X,27HPHI = (1-THRESHOLD/K MAX)/(,F5.2,3H-1))      006440
C        FORMATS FOR CLOSUR INPUT/OUTPUT DATA                        006450
2107 FORMAT(1H0, 1X,18HCLOSURE MODEL WITH)                           006460
2108 FORMAT(1H0, 1X,16HCLOSURE FACTOR =,F6.4,3H+ (,F6.4,1H-,F6.4,     006470
    1     10H)*(1.-R)**,F6.4,/ 5X,33HEXPONENT FOR DECREASING CLOSURE =,006480
    /   F6.3,/                                                       006490
    2      5X,34HEFFECTIVENESS AFTER ONE OVERLOAD =,F6.4/             006500
    3      5X,36HNUMBER OF OVERLOADS FOR SATURATION =,F6.0)           006510
2110 FORMAT(1H0, 1X,40HCORRECTION FACTOR BETA(1) IS A CONSTANT / 5X,9H006520
    1BETA(1) = ,E16.8/ 5X,16HAPPLIED FROM A= ,E16.8,7H TO A = ,E16.8) 006530
2120 FORMAT(1H0, 1X,54HCORRECTION FACTOR BETA(2) IS FINITE WIDTH CORREC006540
    1TION    /5X28HBETA(2) = SQRT(SEC(PI*A/B))  /                     006550
    25X*WHERE THE EFFECTIVE PLATE WIDTH W = *E16.8/                   006560
    35X*APPLIED FROM A = *E16.8,* TO A = *E16.8)                     006570
2130 FORMAT(1H0, 1X,54HCORRECTION FACTOR BETA(3) IS A TABULAR FUNCTION 006580
    1OF A/L / 5X,10HWHERE L = ,E16.8 / 5X,16HAPPLIED FROM A = ,E16.8,7006590
```

```
                2H TO A = ,E16.8// 1X,18A4// 8X,3HA/L,16X,7HBETA(3)/( 5X,E15.8,5X,  006600
               3E15.8))                                                           006610
 2140 FORMAT(1H0, 5X,55HCORRECTION FACTOR BETA(4) IS A TABULAR FUNCTION 006620
               1OF A/L1  / 5X,10HWHERE L = E16.8/ 5X,16HAPPLIED FROM A = ,E16.8,7H006630
               2 TO A = ,E16.8// 1X,18A4// 8X,3HA/L,16X,7HBETA(4)/( 5X,E15.8,5X,E1006640
               35.8))                                                             006650
 2150 FORMAT(1H0, 1X,35HTHIS CASE WILL BE RESTARTED ON-LINE)                006660
 2160 FORMAT(1H0, 1X,36HTHIS CASE WILL BE RESTARTED OFF-LINE        /          006670
               1        48H  RESTART DATA WILL BE WRITTEN ON LOGICAL UNIT 7)      006680
 2165 FORMAT(1H0,1X,36HTHIS CASE WILL BE RESTARTED OFF-LINE        /           006690
               1        48H  RESTART DATA WILL BE READ FROM LOGICAL UNIT 7   )    006700
 2170 FORMAT(1H0, 1X,40HAUTOMATIC UNRETARDED SOLUTION SUPPRESSED)           006710
 2180 FORMAT(1H0, 1X,41HPLANE STRESS YIELD ZONE CONDITION ASSUMED )         006720
 2190 FORMAT(1H0, 1X,41HPLANE STRAIN YIELD ZONE CONDITION ASSUMED )         006730
 2200 FORMAT(1H0, 1X,71HCORRECTION FACTOR BETA(5) IS UNIAXIAL BOWIE SOLU006740
               1TION FOR A SINGLE CRACK/ 5X,35HFROM A CIRCULAR HOLE OF RADIUS R = 006750
               2E16.8 / 5X,16HAPPLIED FROM A = ,E16.8,8H TO A = ,E16.8)           006760
 2205 FORMAT(1H0, 1X,68HCORRECTION FACTOR BETA(6) IS UNIAXIAL BOWIE SOLU006770
               1TION FOR TWO CRACKS /5X,35HFROM A CIRCULAR HOLE OF RADIUS R = ,   006780
               2E16.8 / 5X,17HAPPLIED FROM A = ,E16.8,8H TO A = ,E16.8)           006790
 2206 FORMAT(1H0, 1X,58HCORRECTION FACTOR BETA(7) IS ASTM COMPACT TENSIO006800
               1N SPECIMEN/ 5X,15HWITH A WIDTH OF,E16.8/                          006810
               /      5X,16HAND THICKNESS OF,E16.5/                               006820
               2         5X,17HAPPLIED FROM A = ,E16.8,8H TO A = ,E16.8)          006830
 2207 FORMAT(1H0, 1X,61HCORRECTION FACTOR BETA(8) IS GRUMMAN COMPACT TEN006840
               1SION SPECIMEN/ 5X,15HWITH A WIDTH OF,E16.8/                       006850
               /      5X,16HAND THICKNESS OF,E16.5/                               006860
               2         5X,17HAPPLIED FROM A = ,E16.8,8H TO A = ,E16.8)          006870
 2208 FORMAT(1H0, 1X,71HCORRECTION FACTOR BETA(9) IS ASTM  E647-83 FOR A
               1COMPACT TENSION SPEIMEN/ 5X,15HWITH A WIDTH OF,E16.8/
               /      5X,16HAND THICKNESS OF,E16.5/
               2         5X,17HAPPLIED FROM A = ,E16.8,8H TO A = ,E16.8)
 2210 FORMAT(2X*LAYER*4X*LABEL*7X*DELTA*9X*R*4X*CYCLES PER*/             006880
               /23X*SIGMA*16X*LAYER*//)                                          006890
 2220 FORMAT(3X, I2,5X, 2A4,1X,E13.5,1X,F7.4,2X,F10.2)                     006900
 2230 FORMAT(2X*LAYER*4X*LABEL*8X*MEAN*6X*ALTERNATING*3X                   006910
               /*CYCLES PER*/23X*STRESS*8X*STRESS*7X*LAYER*//)                   006920
 2240 FORMAT(3X, I2,5X, 2A4,1X,E13.5,1X,E13.5,2X,F10.2)                    006930
 2250 FORMAT(1H1,1X, I6,52H SEGMENT SPECTRUM APPLIED IN THE FOLLOWING SEQ006940
               /UENCE //1X*SEGMENT*11X*FLIGHTS PER*6X*MISSION*11X*CUMULATIVE*/  006950
               /21X*MISSION*28X*FLIGHTS*//)                                     006960
 2255 FORMAT(3X, I2,17X, I4,12X, I2,17X, I5)                               006970
 2260 FORMAT(1H0, 1X,19HTHRESHOLD DELTA K =,E17.8,7H (1.0-(,F6.3,4H)*R))006980
 2270 FORMAT(1H1, 70(1H*)/ 5X,34HSTRESS SPECTRUM FOR MISSION NUMBER,I3/ 006990
               /2X, 17A4,A2/1X, 70(1H*)//2X*LAYER*5X*MAXIMUM*7X*MINIMUM*7X       007000
               /*CYCLES*6X*CUMULATIVE*/13X*STRESS*8X*STRESS*8X*PER*8X            007010
               /*CYCLES PER*/40X*LAYER*9X*MISSION*//)                           007020
 2280 FORMAT(3X, I2,4X, 4(F12.3,2X))                                       007030
 2900 FORMAT(1H1,70(1H*)/26X,5HCASE ,I2,5X,4HRUN ,I2/1X,70(1H*))           007040
 3000 FORMAT(1H0, 70(1H*)/ 5X,66HINACTIVE RETARDATION MODEL CHOSEN.EFFEC007050
               1TIVE STRESS MODEL ASSUMED.  /1X, 70(1H*))                        007060
 3010 FORMAT(1H1, 5X,33HINPUT SPECTRUM FOR MISSION NUMBER,I4/ 5X,18A4) 007070
```

```
3030 FORMAT(1H0, 1X,65HCRACK PROPAGATION ANALYSIS USING DIRECT INPUT OF007080
     1 DA/DN VS.DELTA K / 5X,51H WHERE K IS OF THE FORM ....SIGMA*SQRT(F007090
     2I*A)*BETA  )                                                   007100
3040 FORMAT(1H0, 1X,65HCRACK PROPAGATION ANALYSIS USING DIRECT INPUT OF007110
     1 DA/DN VS.DELTA K  / 5X,51HWHERE K IS OF THE FORM .... K=SIGMA*SQR007120
     2T(A)*BETA    )                                                 007130
3050 FORMAT(1H0, 1X,50HCRACK PROPAGATION ANALYSIS USING WALKER'S EQUATI007140
     10N/ 7X,35HDA/DN=C*(DELTA K/((1-R)**(1-M)))**N / 5X,52HWHERE K IS  007150
     20F THE FORM .... K=SIGMA*SQRT(PI*A)*BETA         )             007160
3060 FORMAT(1H0, 1X,50HCRACK PROPAGATION ANALYSIS USING WALKER'S EQUATI007170
     10N/ 7X,35HDA/DN=C*(DELTA K/((1-R)**(1-M)))**N / 5X,51HWHERE K IS O007180
     2F THE FORM .... K=SIGMA*SQRT(A)*BETA             )             007190
3070 FORMAT(1H0, 1X,50HCRACK PROPAGATION ANALYSIS USING SIGMOIDAL EQN.
     1 / 7X,60HDA/DN=EXP(B) (DELTA K/DELTA K*)**P (LN(DELTA K/DELTA K*)
     2)**Q/12X,26H (LN(DELTA KC/DELTA K))**D / 5X,51HWHERE K IS OF THE F...
     30RM .... K=SIGMA*SQRT(PI*A)*BETA            )
3400 FORMAT(1H0,70(1H*)/21X,40HRERUN OF CASE WITH THE FOLLOWING CHANGES007200
     / / 1X,70(1H*))                                                 007210
9000 FORMAT(1H0, 70(1H*)/ 5X,51HNUMBER OF POINTS IN TABULAR CORRECTION 007220
     1FACTOR BETA(I1,*) EXCEEDS 100.*/* CORRECTION FACTOR WILL BE IGNORE007230
     2D.*/1X, 70(1H*) )                                              007240
9010 FORMAT(1H0, 70(1H*)/ 5X,38HERROR IN DECK SETUP.E-O-F ENCOUNTERED. 007250
     1       /1X, 70(1H*))                                           007260
9020 FORMAT(1H0, 70(1H*)/ 5X,48HINCORRECT LABEL CARD ENCOUNTERED.LABEL 007270
     +READ WAS ,2A4,A2,1H*/ 1X,64HEXECUTION SUPPRESSED.PROGRAM WILL COMP007280
     2LETE INPUT DATA PROCESSING/1X, 70(1H*))                        007290
9030 FORMAT(1H0, 70(1H*)/ 1X,71HERROR IN CALCULATING PHI.PROGRAM REQUIR007300
     +ES (AZERO/2(CZERO))SQD .LE.  0.5/ 1X,65HEXECUTION SUPPRESSED. PROG007310
     +RAM WILL COMPLETE INPUT DATA PROCESSING/1X, 70(1H*))           007320
     END                                                            007330
     SUBROUTINE DELTA(A,DELTAK,KMAX,R)                               007340
     COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 007350
     INTEGER RETARD,PLSTRN                                           007360
     COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,  007370
     /              KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX             007380
     REAL KSUBC,KSUBQ                                                007390
     COMMON/CORFAC/ ISURF,RATIO,PHI,THICK,IBETA(10),BETA(10),NPTS,    007400
     /              AOVERB(100),BTABLE(100),NPTS2,AOVRB2(100),        007410
     /              BTABL2(100),ASTART(10),ASTOP(10)                 007420
     REAL KMAX,KMIN                                                  007430
     CALL K(SIGMIN,A,KMIN)                                           007440
     CALL K(SIGMAX,A,KMAX)                                           007450
     R = 0.0                                                         007460
     IF( KMAX .NE. 0.0 ) R = KMIN/KMAX                               007470
     DELTAK=KMAX-KMIN                                                007480
     RETURN                                                          007490
     END                                                            007500
     REAL FUNCTION  TRP2(T,X,Y,M)                                    007510
C                                                                    007520
     DIMENSION T(100),Z(4),D(6)                                      007530
C                                                                    007540
     L1=0                                                            007550
```

```fortran
      X1=X                                                              007560
      Y1=Y                                                              007570
      I = T(1) / 1000. + 1.                                             007580
      J = AMOD(T(1),1000.) + 1.                                         007590
      L=J*M                                                             007600
      I1=J*3+1                                                          007610
      I2=I*J                                                            007620
      M1=M                                                              007630
      DO 10 K=I1,I2,L                                                   007640
      IF(X1-T(K)) 20,20,10                                              007650
   10 CONTINUE                                                          007660
      K=I2+1-J                                                          007670
   20 DO 30 L=4,J,M1                                                    007680
      IF (Y1-T(L)) 40,40,30                                             007690
   30 CONTINUE                                                          007700
      L=J                                                               007710
   40 L1=L1+1                                                           007720
      DO 50 MN=1,3                                                      007730
      N=L+MN-3                                                          007740
      N1=K+(J*(L1-3))+N-1                                               007750
      D(MN)=T(N)                                                        007760
   50 D(MN+3)=T(N1)                                                     007770
   60 Z(L1)=D(4)+(Y1-D(1))*((D(5)-D(4))/(D(2)-D(1))+(                   007780
     1Y1-D(2))/(D(3)-D(1))*((D(6)-D(5))/(D(3)-D(2))                     007790
     2-(D(5)-D(4))/(D(2)-D(1))))                                        007800
      IF (L1-3)40,70,90                                                 007810
   70 DO 80 MN=1,3                                                      007820
      D(MN+3)=Z(MN)                                                     007830
      N1=K+(J*(MN-3))                                                   007840
   80 D(MN)=T(N1)                                                       007850
      L1=4                                                              007860
      Y1=X                                                              007870
      GO TO 60                                                          007880
   90  TRP2=Z(4)                                                        007890
      RETURN                                                            007900
      END                                                              007910
      SUBROUTINE RESTRT(CYC,A,IRSTRT)                                   007920
      COMMON/DATA/I1(7),R1(3)                                           007930
      COMMON/RDATA  /I2(3),R2(5)                                        007940
      COMMON/MDATA   /I3(18),R3(209)                                    007950
      COMMON/LDATA/R4( 600 ),I4( 112)                                   007960
      COMMON/STEPS  /I5(8)                                              007970
      COMMON/CORFAC /I6,R5(3),I7(10),R6(10),I8,R7(200),I9,R8(220)       007980
      COMMON/MKCRVE/R9(100)                                             007990
      COMMON/OUTPOT/R10(3),I10(2)                                       008000
      REWIND 7                                                          008010
      INOUT = IRSTRT - 2                                                008020
      GO TO (100,200,200),INOUT                                         008030
    C                                                                   008040
    C     WRITE RESTART TAPE                                            008050
    C                                                                   008060
  100 WRITE(7) I1,R1,I2,R2,I3,R3                                        008070
```

- 104 -

```
              WRITE(7) R4,I4,I10                                          008080
              WRITE(7) I5,I6,R5,I7,R6,I8,R7,I9,R8,R9,R10,CYC,A            008090
              RETURN                                                      008100
C                                                                         008110
C       READ RESTART TAPE                                                 008120
C                                                                         008130
  200 READ (7) I1,R1,I2,R2,I3,R3                                          008140
      READ (7) R4,I4,I10                                                  008150
      READ (7) I5,I6,R5,I7,R6,I8,R7,I9,R8,R9,R10,CYC,A                    008160
      WRITE(6,2000) CYC,A                                                 008170
      RETURN                                                              008180
 2000 FORMAT(1H0, 70(1H*)/ 5X,43HRESTART TAPE READ. THIS RUN BEGINS AT C008190
     1YCLE , E16.8/5X,24H WITH A CRACK LENGTH OF ,F9.5/1X, 70(1H*))       008200
      END                                                                 008210
      SUBROUTINE CNDIN(IEQN)                                              008220
      COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,     008230
     /              KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX                  008240
      COMMON/PARIS/C1,SN1,DKCOM,C2,SN2                                    008250
      COMMON /DIRECT/ NDADN                                               008260
      COMMON /WALKER/ CWALK,EXPM,EXPN                                     008270
      COMMON /SIGMOID/DKSTAR,TOUGH,BEE,PEA,QUE,DEE
      REAL KSUBC,KSUBQ                                                    008280
      GO TO (100,300,200,400,500),IEQN                                    008290
  100 READ(5,1200) C,SMALLN,KSUBC                                         008300
      WRITE(6,2700) C,SMALLN,KSUBC                                        008310
      GO TO 600                                                           008320
  200 READ(5,1100) PTS                                                    008330
      NDADN = PTS + 0.5                                                   008340
      READ(5,1100) (CARRAY(I),SNARAY(I),I= 1,NDADN)                       008350
      WRITE(6,2000)                                                       008360
      WRITE(6,2100)(CARRAY(I),SNARAY(I),I = 1,NDADN)                      008370
      DO 250 J=1,NDADN                                                    008380
  250 SNARAY(J) = ALOG10(SNARAY(J))                                       008390
      GO TO 600                                                           008400
  300 READ(5,1200) C1,SN1,DKCOM,C2,SN2                                    008410
      IF(DKCOM.GT.0.) GO TO 350                                           008420
      C2 = C1                                                             008430
      SN2 = SN1                                                           008440
      WRITE(6,2550) C1,SN1                                                008450
      GO TO 600                                                           008460
  350 WRITE(6,2600) C1,SN1,DKCOM,C2,SN2,DKCOM                             008470
      GO TO 600                                                           008480
  400 READ(5,1200) CWALK,EXPM,EXPN                                        008490
      WRITE(6,2500) CWALK,EXPM,EXPN                                       008500
      GO TO 600                                                           008510
  500 READ(5,1200)DKSTAR,TOUGH,BEE,PEA,QUE,DEE
      WRITE(6,2520)BEE,DKSTAR,PEA,DKSTAR,QUE,TOUGH,DEE
 2520 FORMAT(1X,10HDA/DN=EXP(,F12.7,12H)*((DELTA-K/,F12.7,3H)**,F6.2,1H)
     1,1H* / 1X,13H((LN(DELTA-K/,F12.7,4H))**,F6.2,2H)*
     2/,1X,5H((LN(,F12.7,12H/DELTA-K))**,F6.2,1H))
  600 READ(5,1100) KSUBQ,SIGMAY                                           008530
      WRITE(6,2400) KSUBQ,SIGMAY                                          008540
```

```
      1100 FORMAT(2E10.0)                                                 008550
      1200 FORMAT(8E10.0)                                                 008560
      2000 FORMAT(1H0, 1X,32HDIRECT INPUT OF DA/DN VS DELTA K//34X,7HDELTA K 008570
           1  ,14X,5HDA/DN )                                              008580
      2100 FORMAT( 5X,2E20.8)                                             008590
      2400 FORMAT(1H0, 1X,7HKSUBQ = ,E15.8,5X,15HYIELD STRESS =  ,E15.8 )  008600
      2500 FORMAT(1H0, 1X,3HC =,E16.8,5X,3HM =,F6.4,5X,3HN =,F6.3)         008610
      2550 FORMAT(1H0,1X,30HLINEAR PARIS FIT AS FOLLOWS         /          008620
           /      5X,E12.4,11H(DELTA K)** ,F5.3)                          008630
      2600 FORMAT(1H0, 1X,29HBILINEAR PARIS FIT AS FOLLOWS/ 5X,E12.4,11H(DELT008640
           1A K)**,F5.3,22H FOR DELTA K LESS THAN,E16.8/5X,E12.4,11H(DELTA K)*008650
           2*  ,F5.3,22H FOR DELTA K MORE THAN ,E16.8  )                  008660
      2700 FORMAT(1H0, 1X,3HC =,E16.8,5X,8HSMALLN =,F6.3,5X,7HKSUBC =,E16.8) 008670
           RETURN                                                         008680
           END                                                            008690
           BLOCK DATA                                                     008700
           COMMON/MKCRVE/ MK(100)                                         008710
           REAL MK                                                        008720
      C             INITIALIZATION VALUES FOR FIRST 84 ELEMENTS OF MK     008730
           DATA    MK/                                                    008740
           1     11006.,0.05,0.10,0.20,0.30,0.40,0.50,                    008750
           2          0.0,1.00,1.00,1.00,1.00,1.00,1.00,                  008760
           3          0.1,1.01,1.01,1.01,1.01,1.01,1.00,                  008770
           4          0.2,1.03,1.03,1.02,1.02,1.01,1.00,                  008780
           5          0.3,1.06,1.06,1.04,1.03,1.02,1.00,                  008790
           6          0.4,1.12,1.12,1.08,1.05,1.02,1.00,                  008800
           7          0.5,1.22,1.18,1.14,1.08,1.03,1.00,                  008810
           8          0.6,1.34,1.30,1.22,1.13,1.06,1.01,                  008820
           9          0.7,1.48,1.42,1.31,1.20,1.08,1.02,                  008830
           A          0.8,1.64,1.57,1.41,1.26,1.13,1.04,                  008840
           B          0.9,1.77,1.68,1.50,1.32,1.18,1.08,                  008850
           C          1.0,1.84,1.75,1.59,1.38,1.22,1.10/                  008860
           END                                                            008870
           SUBROUTINE CELI2(RES,AK,A,B,IER)                               008880
      C                                                                   008890
      C     ..............................................................008900
      C                                                                   008910
      C     SUBROUTINE CELI2                                              008920
      C                                                                   008930
      C     PURPOSE                                                       008940
      C        COMPUTES THE GENERALIZED COMPLETE ELLIPTIC INTEGRAL OF     008950
      C        SECOND KIND.                                               008960
      C                                                                   008970
      C     USAGE                                                         008980
      C        CALL CELI2(RES,AK,A,B,IER)                                 008990
      C                                                                   009000
      C     DESCRIPTION OF PARAMETERS                                     009010
      C        RES    - RESULT VALUE                                      009020
      C        AK     - MODULUS (INPUT)                                   009030
      C        A      - CONSTANT TERM IN NUMERATOR                        009040
      C        B      - FACTOR OF QUADRATIC TERM IN NUMERATOR             009050
      C        IER    - RESULTANT ERROR CODE WHERE                        009060
```

```
C                         IER=0    NO ERROR                               009070
C                         IER=1    AK NOT IN RANGE -1 TO +1               009080
C                                                                         009090
C        REMARKS                                                          009100
C           FOR AK = +1,-1 THE RESULT VALUE IS SET TO 1.E75 IF B IS       009110
C           POSITIVE, TO -1.E75 IF B IS NEGATIVE.                         009120
C           SPECIAL CASES ARE                                             009130
C           K(K) OBTAINED WITH A = 1, B = 1                               009140
C           E(K) OBTAINED WITH A = 1, B = CK*CK WHERE CK IS               009150
C           COMPLEMENTARY MODULUS.                                        009160
C           B(K) OBTAINED WITH A = 1, B = 0                               009170
C           D(K) OBTAINED WITH A = 0, B = 1                               009180
C           WHERE K, E, B, D DEFINE SPECIAL CASES OF THE GENERALIZED      009190
C           COMPLETE ELLIPTIC INTEGRAL OF SECOND KIND IN THE USUAL        009200
C           NOTATION, AND THE ARGUMENT K OF THESE FUNCTIONS MEANS         009210
C           THE MODULUS.                                                  009220
C                                                                         009230
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                    009240
C           NONE                                                          009250
C                                                                         009260
C        METHOD                                                           009270
C           DEFINITION                                                    009280
C           RES=INTEGRAL((A+B*T*T)/(SQRT((1+T*T)*(1+(CK*T)**2))*(1+T*T))009290
C           SUMMED OVER T FROM 0 TO INFINITY).                           009300
C           EVALUATION                                                    009310
C           LANDENS TRANSFORMATION IS USED FOR CALCULATION.               009320
C           REFERENCE                                                     009330
C           R.BULIRSCH,  NUMERICAL CALCULATION OF ELLIPTIC INTEGRALS      009340
C           AND ELLIPTIC FUNCTIONS, HANDBOOK SERIES SPECIAL FUNCTIONS,    009350
C           NUMERISCHE MATHEMATIK VOL. 7, 1965, PP. 78-90.                009360
C                                                                         009370
C        ...........................................................009380
C                                                                         009390
      IER=0                                                               009400
C                                                                         009410
C     TEST RANGE                                                          009420
C                                                                         009430
      CK=AK*AK                                                            009440
      IF(CK-1.) 20,20,10                                                  009450
   10 IER=1                                                               009460
      RETURN                                                              009470
C                                                                         009480
C     COMPUTE COMPLEMENTARY MODULUS                                       009490
C                                                                         009500
   20 GEO=SQRT(1.0-CK)                                                    009510
      IF(GEO) 70,30,70                                                    009520
C                                                                         009530
C     SET RESULT VALUE = OVERFLOW                                         009540
C                                                                         009550
   30 IF(B) 40,60,50                                                      009560
   40 RES=-1.E38                                                          009570
      RETURN                                                              009580
```

```
     50 RES=1.E38                                                        009590
        RETURN                                                           009600
     60 RES=A                                                            009610
        RETURN                                                           009620
C                                                                        009630
C       COMPUTE INTEGRAL                                                 009640
C                                                                        009650
     70 ARI=1.                                                           009660
        AA=A                                                             009670
        AN=A+B                                                           009680
        W=B                                                              009690
     80 W=W+AA*GEO                                                       009700
        W=W+W                                                            009710
        AA=AN                                                            009720
        AARI=ARI                                                         009730
        ARI=GEO+ARI                                                      009740
        AN=W/ARI+AN                                                      009750
C                                                                        009760
C       TEST OF ACCURACY                                                 009770
C                                                                        009780
        IF(AARI-GEO-1.E-4*AARI) 100,100,90                              009790
     90 GEO=SQRT(GEO*AARI)                                               009800
        GEO=GEO+GEO                                                      009810
        GO TO 80                                                         009820
    100 RES=.78539816*AN/ARI                                            009830
        RETURN                                                           009840
        END                                                              009850
        SUBROUTINE GRWCRK(CYC,A,DN)                                      009860
        COMMON/PDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 009870
        INTEGER RETARD,PLSTRN                                            009880
        COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10), 009890
       /              NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS                009900
        COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                    009910
        EXTERNAL RATE,WHELER,WLNBRG                                      009920
        SIGMAX = SMAX(J4,ISEG)                                           009930
        SIGMIN = SMIN(J4,ISEG)                                           009940
        IF(MODEL.GT.0) GO TO 100                                         009950
        CALL RK1DES(CYC,A,DN,RATE)                                       009960
        RETURN                                                           009970
    100 GO TO (110,120,130,140,150),MODEL                               009980
    110 CALL YLDZNE(CYC,A,DN,WHELER)                                     009990
        GO TO 200                                                        010000
    120 CALL YLDZNE(CYC,A,DN,WLNBRG)                                     010010
        GO TO 200                                                        010020
    130 CONTINUE                                                         010030
        CALL CLOSUR(CYC,A,DN)                                            010040
        GO TO 200                                                        010050
    140 CONTINUE                                                         010060
    150 CONTINUE                                                         010070
    200 RETURN                                                           010080
        END                                                              010090
```

```fortran
      SUBROUTINE TRANS(A,ATRANS,CYCTR)                                010100
      COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10),  010110
     /             NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS                010120
      COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                   010130
      COMMON/CORFAC/ ISURF,RATIO,PHI,THICK,IBETA(10),BETA(10),NPTS,    010140
     /               AOVERB(100),BTABLE(100),NPTS2,AOVRB2(100),       010150
     /               BTABL2(100),ASTART(10),ASTOP(10)                 010160
      ATRANS = THICK                                                  010170
      IF(A.LT.ATRANS)RETURN                                           010180
  100 AEFF = ATRANS/(2.*RATIO)                                        010190
      A = AEFF                                                        010200
      ISTOP = 2                                                       010210
      WRITE(6,1000) A,CYCTR                                           010220
 1000 FORMAT(1H0, 70(1H*)/ 5X,55HTRANSITION TO A THRU CRACK OF EFFECTIVE010230
     1 LENGTH, AEFF = ,F9.5,4H AT ,F12.2,7H CYCLES/1X, 70(1H*))       010240
      RETURN                                                          010250
      END                                                             010260
      SUBROUTINE RATE (CYCLE,A,DADN)                                  010270
      COMMON/DATA/ EQN,NASA,J1PR,J2PR,J3PR,J4PR,J5PR,AZERO,AMAX,NZERO 010280
      INTEGER EQN                                                     010290
      REAL NZERO                                                      010300
      COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,  010310
     /              KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX              010320
      REAL KSUBC,KSUBQ                                                010330
      COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10),  010340
     /             NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS                010350
      COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                   010360
      COMMON/CORFAC/ ISURF,RATIO,PHI,THICK,IBETA(10),BETA(10),NPTS,    010370
     /               AOVERB(100),BTABLE(100),NPTS2,AOVRB2(100),       010380
     /               BTABL2(100),ASTART(10),ASTOP(10)                 010390
      COMMON/OUTPOT/ KMAX,KMAXA,DELTAK,IFLT,DADNPR                    010400
      COMMON/PARIS/C1,SN1,DKCOM,C2,SN2                                010410
      COMMON/DIRECT/ NDADN                                            010420
      COMMON /WALKER/ CWALK,EXPM,EXPN                                 010430
      COMMON /SIGMOID/DKSTAR,TOUGH,BEE,PEA,QUE,DEE
      REAL KMAX                                                       010440
      REAL KMAXA                                                      010450
      CALL DELTA(A,DELTAK,KMAX,R)                                     010460
      IF(ISTOP .NE. 0)GO TO 575                                       010470
      IF( R. GE. RCUT) R = RCUT                                       010480
      CALL K(SMAX(J4,ISEG),A,KMAXA)                                   010490
      IF(DELTAK.LE.0.0) GO TO 300                                     010500
      IF(ISURF.EQ.0) GO TO 50                                         010510
      CALL TRANS(A,ATRANS,CYCLE)                                      010520
      IF(A.LT.ATRANS) GO TO 50                                        010530
      RETURN                                                          010540
   50 THRSLD = DELKTH*(1.0-RMULT*R)                                   010550
      IF (DELTAK.LE.THRSLD) GO TO 300                                 010560
      GO TO (100,200,230,240,250),EQN                                010570
  100 DENOM = (1.0 - R) * KSUBC - DELTAK                             010580
      IF(KMAXA.GE.KSUBQ) GO TO 400                                   010590
      IF(DENOM.LE.0.0) GO TO 525                                      010600
```

- 109 -

```
      DADN=(C*(DELTAK)**SMALLN)/DENOM                                  010610
      RETURN                                                           010620
  200 DELKC = KSUBQ - KMAXA                                            010630
      IF(DELKC.LE.0.0) GO TO 400                                       010640
      C = C1                                                           010650
      SMALLN = SN1                                                     010660
      IF(DELTAK.GE.DKCOM) C = C2                                       010670
      IF(DELTAK.GE.DKCOM) SMALLN = SN2                                 010680
      DADN=C*(DELTAK)**SMALLN                                          010690
      RETURN                                                           010700
  230 DADN = TBLKUP(CARRAY,SNARAY,NDADN,100,DELTAK)                    010710
      DADN = 10.**DADN                                                 010720
      IF(KMAXA.GE.KSUBQ) GO TO 400                                     010730
      RETURN                                                           010740
  240 IF(KMAXA.GE.KSUBQ) GO TO 400                                     010750
      DADN=CWALK*(DELTAK/((1.-R)**(1.-EXPM)))**EXPN                    010760
      RETURN                                                           010770
  250 IF (KMAXA .GE. KSUBQ) GO TO 400
      TOUG=TOUGH*(1.-.4)
      IF (DELTAK .GE. TOUG) GO TO 260
      XKSTAR=DKSTAR*(1.-.4)
      IF (DELTAK .LE. XKSTAR) GO TO 300
      DADN=EXP(BEE)*((DELTAK/XKSTAR)**PEA)*((ALOG(DELTAK/XKSTAR))**QUE)
     1*((ALOG(TOUG/DELTAK))**DEE)
      RETURN
  260 ISTOP=1
      WRITE(6,270)
  270 FORMAT(1H0, 70(1H*)/ 5X,50HDELTA-K EXCEEDS THE TOUGHNESS PROB. IS
     1 TERMINATED/1X,70(1H*)/1H0, 1X,26HLAST CALCULATED VALUES ARE///
     2)
      GO TO 575
  300 DADN=0.0                                                         010780
      RETURN                                                           010790
  400 ISTOP = 1                                                        010800
      WRITE(6,500)                                                     010810
  500 FORMAT(1H0, 70(1H*)/ 5X,46HKMAX APPLIED EXCEEDS KSUBQ. PROBLEM TER010820
     1MINATED/1X, 70(1H*)/1H0, 1X,26HLAST CALCULATED VALUES ARE///)    010830
      GO TO 575                                                        010840
  525 WRITE(6,550)                                                     010850
  550 FORMAT(1H0, 70(1H*)/ 5X,49HDELTA K EXCEEDS (1-R)KSUBC. PROBLEM IS 010860
     1TERMINATED/1X,70(1H*)/1H0, 1X,26HLAST CALCULATED VALUES ARE///)  010870
      ISTOP = 1                                                        010880
  575 CONTINUE                                                         010890
      WRITE(6,600) J1,J2,ISEG,IFLT,J4,CYCLE,A,KMAXA,KMAX,DELTAK,DADN   010900
  600 FORMAT( 5X,18HBLOCK IN SPECTRUM ,I4/                             010910
     /        5X,18HSEGMENT NUMBER    ,I4/                             010920
     /        5X,18HMISSION NUMBER    ,I4/                             010930
     /        5X,18HFLIGHT NUMBER     ,I6/                             010940
     /        5X,18HLAYER IN MISSION  ,I4/                             010950
     5        5X,18HACCUMULATED CYCLES,E16.8/                          010960
     6        5X,18HCRACK LENGTH      ,E16.8/                          010970
     7        5X,18HKMAX APPLIED      ,E16.8/                          010980
```

```
      8          5X,18HKMAX EFFECTIVE     ,E16.8/                    010990
      9          5X,18HDELTA K            ,E16.8/                    011000
      /          5X,18HDA/DN              ,E16.8)                    011010
       RETURN                                                        011020
       END                                                          011030
       SUBROUTINE K(SIGMA,A,KM)                                     011040
       COMMON/CORFAC/DUMMY(14),BETA(10),DUMMY1(422)                 011050
       COMMON/DATA/ EQN,NASA,J1PR,J2PR,J3PR,J4PR,J5PR,AZERO,AMAX,NZERO  011060
       INTEGER EQN                                                  011070
       REAL NZERO                                                   011080
       REAL KM                                                      011090
       DATA PI/3.14159265/                                          011100
       CALL BETAS (SIGMA,A,BETAT,Q)                                 011110
       IF(NASA.NE.0) GO TO 100                                      011120
       KM=SIGMA*SQRT(PI*A)*BETAT                                    011130
       RETURN                                                       011140
  100 KM=SIGMA*SQRT(A)*BETAT                                        011150
       RETURN                                                       011160
       END                                                          011170
       SUBROUTINE BETAS(SIGMA,A,BETAT,Q)                            011180
       COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,  011190
      /              KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX           011200
       REAL KSUBC,KSUBQ                                             011210
       COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                011220
       COMMON/CORFAC/ ISURF,RATIO,PHI,THICK,IBETA(10),BETA(10),NPTS,  011230
      /              AOVERB(100),BTABLE(100),NPTS2,AOVRB2(100),    011240
      /              BTABL2(100),ASTART(10),ASTOP(10)               011250
       COMMON/MKCRVE/ MK(100)                                       011260
       REAL MK,MSUBK,M1                                             011270
       BETAT=1.0                                                    011280
       MSUBK=1.0                                                    011290
       Q=1.0                                                        011300
       PI = 3.14159265                                              011310
    5 DO 100 I=1,10                                                 011320
       J=IBETA(I)                                                   011330
       IF(J.EQ.0) GO TO 100                                         011340
       IF(A.LT.ASTART(J). OR .A.GT.ASTOP(J)) GO TO 100              011350
       GO TO(10,20,30,40,50,60,70,80,90),J                         011360
C                                                                   011370
C      CONSTANT MULTIPLIER                                          011380
C                                                                   011390
   10 BETAT = BETAT * BETA(J)                                       011400
       GO TO 100                                                    011410
C                                                                   011420
C      FINITE WIDTH SECANT CORRECTION                              011430
C                                                                   011440
   20 HOLE = BETA(5)+BETA(6)                                        011450
       SMALLC=A                                                     011460
       IF(ISURF.NE.0) SMALLC=A/(2.*RATIO)                          011470
       SMALLK = (SMALLC + HOLE)/BETA(J)                            011480
       IF(SMALLK.GT.0.5) GO TO 200                                  011490
       BETAT = BETAT*SQRT(1./COS(PI*SMALLK))                       011500
```

- 111 -

```fortran
         GO TO 100                                                    011510
C                                                                     011520
C        TABULAR CORRECTION FACTOR                                    011530
C                                                                     011540
   30 SMALLK=A/BETA(J)                                                011550
      BETAT=BETAT*TBLKUP(AOVERB,BTABLE,NPTS,100,SMALLK)               011560
      GO TO 100                                                       011570
C                                                                     011580
C        SECOND TABULAR CORRECTION FACTOR                             011590
C                                                                     011600
   40 SMALLK=A/BETA(J)                                                011610
      BETAT=BETAT*TBLKUP(AOVRB2,BTABL2,NPTS2,100,SMALLK)              011620
      GO TO 100                                                       011630
C                                                                     011640
C        BOWIE SOLUTION FOR SINGLE CRACK FROM CIRCULAR HOLE           011650
C                                                                     011660
   50 BETAT = BETAT * (0.6762062 + (0.8733015/(0.3245442+A/BETA(J)))) 011670
      GO TO 100                                                       011680
C                                                                     011690
C        BOWIE SOLUTION FOR DOUBLE CRACK FROM CIRCULAR HOLE           011700
C                                                                     011710
   60 BETAT = BETAT * (0.9438510 + (0.6805078/(0.2771965+A/BETA(J)))) 011720
      GO TO 100                                                       011730
C                                                                     011740
C        SOLUTION FOR ASTM COMPACT TENSION SPECIMEN USING J. C. NEWMAN 011750
C        EQUATION (12) FROM 'STRESS ANALYSIS OF THE COMPACT SPECIMEN  011760
C        INCLUDING THE EFFECTS OF PIN LOADING'                        011770
   70     AW = A/BETA(I)                                              011780
          BETAT = BETAT*(4.55-40.32*AW+414.7*AW**2.-1698.*AW**3.      011790
     1    +3781.*AW**4.-4287.*AW**5.+2017.*AW**6.)/SQRT(BETA(I)*A*PI)/ 011800
     2      THICK                                                     011810
      GO TO 100                                                       011820
C        SOLUTION FOR GRUMMAN COMPACT TENSION SPECIMEN                011830
C        H/W = .95          D/W = .25                                 011840
   80     AW = A/BETA(I)                                              011850
          POLY = .1229+16.4098*AW-37.395*AW**2.+54.7667*AW**3.        011860
      IF ( AW .LE. 0.5 ) GO TO 81                                     011870
          POLY = 114.054-830.132*AW+2327.177*AW**2.-2890.811*AW**3.   011880
     1      +1382.306*AW**4.                                          011890
   81     BETAT = BETAT*POLY/SQRT(PI*A)/THICK                         011900
      GO TO 100                                                       011910
C        SOLUTION FOR COMPACT TENSION SPECIMEN FROM ASTM E 747-83
C        CONSTANT-LOAD-AMPLITUDE FATIGUE CRACK GROWTH RATES ABOVE
C        10-08 M/CYCLE
   90     AW=A/BETA(I)
          BETAT=BETAT*(2.+AW)*(.886+4.64*AW-13.32*AW**2.+14.72*AW**3.
     1    -5.6*AW**4.)/((THICK*(1-AW)**1.5)*SQRT(BETA(I)*PI*A))
  100 CONTINUE                                                        011930
      IF(ISURF.EQ.0) RETURN                                           011940
C                                                                     011950
C        SURFACE FLAW CORRECTION                                      011960
C        FROM NEWMAN - NASA TN D-8244                                 011970
```

```
C                                                                   011980
      Q=PHI**2.0-0.212*(SIGMA/SIGMAY)**2.0                           011990
      AOVERT=A/THICK                                                 012000
      AOVERC=2.*RATIO                                                012010
      P=2.+8.*AOVERC**3.                                             012020
      M1=1.13-0.1*AOVERC                                             012030
      MSUBK=(SQRT(Q/AOVERC)-M1)*AOVERT**P                            012040
      BETAT=BETAT*(M1+MSUBK)/SQRT(Q)
      RETURN                                                         012060
  200 WRITE(6,1000)                                                  012070
 1000 FORMAT(1H0, 70(1H*)/ 5X,52HCRACK LENGTH EXCEEDS PLATE WIDTH. TERMIO12080
     1NATE PROBLEM.   / 70(1H*))                                     012090
      ISTOP = 1                                                      012100
      RETURN                                                         012110·
      END                                                            012120
      SUBROUTINE YLDZNE(CYC,A,DN,FR)                                 012130
C                                                                   012140
C     THIS ROUTINE CONTROLS APPLICATION OF THE WHEELER MODEL AND THE 012150
C     EFFECTIVE STRESS(WILLENBORG)MODEL BASED ON THE PROGRESS THRU A 012160
C     YIELD ZONE(ASUBP)DUE TO AN OVERLOAD.                           012170
C                                                                   012180
      COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 012190
      INTEGER RETARD,PLSTRN                                          012200
      COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10),  012210
     /             NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS              012220
      COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,  012230
     /             KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX             012240
      REAL KSUBC,KSUBQ                                               012250
      COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                  012260
      REAL KMAX,KOVLD                                                012270
      EXTERNAL FR,RATE,DNDA                                          012280
      CYCF = CYC + DN                                                012290
      IF(RETARD.EQ.1) GO TO 100                                      012300
C                                                                   012310
C     IS THIS LAYER SUBJECT TO RETARDATION &                        012320
C                                                                   012330
      IF(SMAX(J4,ISEG).GE.OVLD) GO TO 800                           012340
C                                                                   012350
C     RETARDATION IS APPLIED.                                       012360
C                                                                   012370
      RETARD = 1                                                    012380
C                                                                   012390
C     DETERMINE EXTENT OF ELASTIC-PLASTIC INTERFACE                 012400
C                                                                   012410
      CALL K(OVLD,A,KOVLD)                                           012420
      RSUBY = RY(KOVLD,PLSTRN)                                       012430
      ASUBP = A + RSUBY                                              012440
C                                                                   012450
C     WILL FIRST CYCLE OF THIS LAYER CAUSE ASUBP TO BE EXCEEDED &   012460
C                                                                   012470
  100 CALL SIGEFF(A,ASUBP,SIGMAY)                                   012480
      IF(RETARD.EQ.0) GO TO 800                                     012490
```

```
        CALL K(SIGMAX,A,KMAX)                                   012500
        RSUBY1 = RY(KMAX,PLSTRN)                                 012510
        IF(A+RSUBY1.LT.ASUBP) GO TO 200                         012520
        RETARD = 0                                              012530
        GO TO 800                                               012540
C                                                               012550
C       THIS LAYER IS SUBJECT TO RETARDATION.                   012560
C       ASSUME THAT RETARDATION APPLIES OVER THE ENTIRE LAYER.  012570
C                                                               012580
  200 CYC1 = CYC                                                012590
        AA = A                                                  012600
        CALL RK1DES(CYC1,AA,DN,FR)                              012610
        IF(ISTOP.NE.0) RETURN                                   012620
        IF(AA.GT.ASUBP) GO TO 400                               012630-
        CALL K(SIGMAX,AA,KMAX)                                  012640
        RSUBY = RY(KMAX,PLSTRN)                                 012650
C                                                               012660
C       CHECK  ASSUMPTION.                                      012670
C                                                               012680
        IF(AA+RSUBY.GE.ASUBP) GO TO 400                         012690
        CYC = CYC1                                              012700
        A = AA                                                  012710
        RETURN                                                  012720
C                                                               012730
C       ENTIRE LAYER IS NOT RETARDED                            012740
C       CALCULATE DELTA A AND ITS ASSOCIATED YIELD ZONE(RSUBY) SUCH THAT  012750
C       (A+DA+RSUBY = ASUBP) TO A GIVEN TOLERANCE(TOL)          012760
C                                                               012770
  400 TOL = 1.0E-5                                              012780
        RED = 1.0                                               012790
        SUBT = 0.1                                              012800
        CYC1 = CYC                                              012810
        AA = A                                                  012820
  500 DA = (ASUBP-(AA+RSUBY1)) * RED                            012830
        CALL SIGEFF(AA+DA,ASUBP,SIGMAY)                         012840
        CALL K(SIGMAX,AA+DA,KMAX)                               012850
        RSUBY = RY(KMAX,PLSTRN)                                 012860
        IF(AA+DA+RSUBY.LT.ASUBP) GO TO 700                      012870
        RED = RED - SUBT                                        012880
        GO TO 500                                               012890
  600 RED = RED + SUBT                                          012900
        SUBT = SUBT/10.                                         012910
        GO TO 500                                               012920
  700 IF(SUBT.GT.TOL) GO TO 600                                 012930
C                                                               012940
C       KNOWING DN, DETERMINE NUMBER OF CYCLES IN THIS LAYER REQUIRED  012950
C       TO PRODUCE DA.                                          012960
C                                                               012970
        CALL RUNKUT(AA,CYC1,DA,DNDA)                            012980
        IF(ISTOP.NE.0) RETURN                                   012990
        CYC = CYC1                                              013000
        A = AA                                                  013010
```

- 114 -

```
C                                                             013020
C      REMAINDER OF LAYER IS NOT SUBJECT TO RETARDATION.       013030
C      USE UNRETARDED DADN FUNCTION(RATE)                      013040
C                                                             013050
       RETARD = 0                                              013060
       DN = CYCF - CYC                                         013070
       IF(DN.LE.0.0) RETURN                                    013080
C                                                             013090
C      UNRETARDED DADN FUNCTION                                013100
C                                                             013110
   800 SIGMAX = SMAX(J4,ISEG)                                  013120
       SIGMIN = SMIN(J4,ISEG)                                  013130
       OVLD = SIGMAX                                           013140
       CALL RK1DES(CYC,A,DN,RATE)                              013150
       RETURN                                                  013160
       END                                                     013170
       REAL FUNCTION RY(K,PLSTRN)                              013180
       COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,  013190
      /          KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX          013200
       REAL KSUBC,KSUBQ                                        013210
       REAL K                                                  013220
       INTEGER PLSTRN                                          013230
       DATA PI,ROOT2 /3.1415926,2.828428/                      013240
       IF(K.LE.0.) GO TO 999                                   013250
       RY = ((K/SIGMAY)**2.)/(2.*PI)                           013260
       IF(PLSTRN.NE.0) RY = RY/ROOT2                           013270
       RETURN                                                  013280
   999 RY= 0.0                                                 013290
       RETURN                                                  013300
       END                                                     013310
       SUBROUTINE DNDA(A,CYC,DDNINV)                           013320
C                                                             013330
C      THIS ROUTINE CALCULATES THE NUMBER OF CYCLES IN A LAYER OVER   013340
C      WHICH RETARDATION IS APPLIED                            013350
C                                                             013360
       COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 013370
       INTEGER RETARD,PLSTRN                                   013380
       GO TO (10,20),MODEL                                     013390
    10 CALL WHELER(CYC,A,DADN)                                 013400
       GO TO 30                                                013410
    20 CALL WLNBRG(CYC,A,DADN)                                 013420
    30 DDNINV = 1.0/DADN                                       013430
       RETURN                                                  013440
       END                                                     013450
       SUBROUTINE WLNBRG(CYC,A,DADN)                           013460
C                                                             013470
C      THIS ROUTINE BRINGS THE EFFECTIVE STRESSES GENERATED BY THE    013480
C      WILLENBORG MODEL INTO THE UNRETARDED GROWTH RATE EQUATIONS.    013490
C                                                             013500
       COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 013510
       INTEGER RETARD,PLSTRN                                   013520
       COMMON/MDATA/ MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC,  013530
```

```
      /                KSUBQ,SIGMAY,DELKTH,RMULT,RCUT,OLMAX          013540
       REAL KSUBC,KSUBQ                                              013550
       CALL SIGEFF(A,ASUBP,SIGMAY)                                   013560
       CALL RATE(CYC,A,DADN)                                         013570
       RETURN                                                        013580
       END                                                           013590
       SUBROUTINE SIGEFF(A,ASUBP,SIGMAY)                             013600
C                                                                    013610
C      THIS ROUTINE COMPUTES THE EFFECTIVE STRESSES FOR USE IN THE   013620
C      WILLENBORG MODEL. THE EFFECTIVE STRESSES ARE STORED IN LOCATIONS 013630
C      'SIGMAX' AND 'SIGMIN' IN COMMON BLOCK /RDATA/.                013640
C                                                                    013650
C                                                                    013660
       COMMON/DATA/IDUM1,NASA,IDUM2(5),DUM3(3)                       013670
       COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,DUMMY,SMALLM 013680
       INTEGER RETARD,PLSTRN                                         013690
       COMMON/LDATA/SMAX( 20,10),SMIN( 20,10),CYCLES( 20,10),NLYRS(10), 013700
      /                NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS           013710
       COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                 013720
       COMMON/MDATA/MATID(18),C,SMALLN,CARRAY(100),SNARAY(100),KSUBC, 013730
      1               KSUBQ,DUMMYY,DELKTH,RMULT,RCUT,OLMAX           013740
       REAL KSUBC,KSUBQ,KMAX                                         013750
C                                                                    013760
C      PUT APPLIED STRESSES IN SIGMAX AND SIGMIN                     013770
C                                                                    013780
       SIGMAX = SMAX(J4,ISEG)                                        013790
       SIGMIN = SMIN(J4,ISEG)                                        013800
       IF(MODEL.EQ.2) GO TO 100                                      013810
       RETURN                                                        013820
  100 CALL BETAS(SIGMAX,A,BETAT,QMAX)                                013830
       IF(A.GT.ASUBP) GO TO 200                                      013840
       SIGREF =  (SIGMAY*SQRT(2.0*(ASUBP-A)/A))/BETAT                013850
       IF(NASA.NE.0) SIGREF=SIGREF*SQRT(3.1415926)                   013860
       IF(PLSTRN.NE.0) SIGREF = SIGREF * SQRT(2.828428)              013870
       SIGRED = SIGREF - SIGMAX                                      013880
       CALL DELTA(A,DELTAK,KMAX,R)                                   013890
       THRSLD = DELKTH*(1.-RMULT*R)                                  013900
       PHI = (1.-THRSLD/KMAX)/(OLMAX-1.)                             013910
       IF(OLMAX.EQ.0.) PHI = 1.                                      013920
       SIGRED = PHI * SIGRED                                         013930
       IF(SIGRED.LE.0.0) GO TO 200                                   013940
       SIGMAX = SIGMAX - SIGRED                                      013950
       IF(SIGMAX.LT.0.0) SIGMAX = 0.0                                013960
       SIGMIN = SIGMIN - SIGRED                                      013970
       IF(SIGMIN.LT.0.0) SIGMIN = 0.0                                013980
       RETURN                                                        013990
  200 RETARD = 0                                                     014000
       RETURN                                                        014010
       END                                                           014020
       SUBROUTINE WHELER(CYC,A,DADN)                                 014030
C                                                                    014040
C      THIS ROUTINE APPLIES THE WHEELER CORRECTION TO THE UNRETARDED 014050
```

```
C     GROWTH RATE                                                    014060
C                                                                    014070
      COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,ASUBP,SMALLM 014080
      INTEGER RETARD,PLSTRN                                          014090
      CSUBP = 1.0                                                    014100
C                                                                    014110
C     DETERMINE EXTENT OF CURRENT YIELD ZONE                         014120
C                                                                    014130
      CALL K(SIGMAX,A,KMAX)                                          014140
      RSUBY = RY(KMAX,PLSTRN)                                        014150
      IF(A + RSUBY .GE. ASUBP) GO TO 20                              014160
C                                                                    014170
C     CALCULATE WHEELER'S RETARDATION PARAMETER                      014180
C                                                                    014190
      CSUBP = (RSUBY/(ASUBP-A))** SMALLM                             014200
   20 CALL RATE(CYC,A,DADN)                                          014210
      DADN = CSUBP * DADN                                            014220
      RETURN                                                         014230
      END                                                            014240
      SUBROUTINE RK1DES(CYC,A,DCYC,F)                                014250
C                                                                    014260
      EXTERNAL F                                                     014270
      COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORTRD                  014280
      IF(DCYC .GE. 1.) GO TO 300
      CYCF=CYC+DCYC                                                  014300
      AO=A                                                           014310
   50 H=0.005*AO                                                     014320
      A1=AO+H                                                        014330
      AGROW=AO+2.0*H                                                 014340
      CALL F(CYC,A1,DADN)                                            014350
      IF(DADN.LE.0.) GO TO 200                                       014360
      IF(ISTOP.NE.0) GO TO 250                                       014370
      DCYCR=2.*H/DADN                                                014380
      IF(DCYCR.GT.DCYC) GO TO 100                                    014390
      CYC=CYC+DCYCR                                                  014400
      DCYC=DCYC-DCYCR                                                014410
      AO=AGROW                                                       014420
      GO TO 50                                                       014430
  100 DA=(CYCF-CYC)*DADN                                             014440
      AO=AO+DA                                                       014450
  200 CYC=CYCF                                                       014460
      A=AO                                                           014470
  250 RETURN                                                         014480
  300 CALL RUNKUT(CYC,A,DCYC,F)                                      014490
      RETURN                                                         014500
      END                                                            014510
      FUNCTION TBLKUP(X,Y,N,NMAX,ARG)                                014520
C                                                                    014530
C                                                                    014540
      DIMENSION X(NMAX),Y(NMAX)                                      014550
      DO 10 I=1,N                                                    014560
      IF(X(I)-ARG) 10,20,20                                          014570
```

```
      10 CONTINUE                                                        014580
         I=N                                                             014590
      20 IF(I-1)30,30,40                                                 014600
      30 I=2                                                             014610
      40 SLOPE=(Y(I)-Y(I-1))/(X(I)-X(I-1))                               014620
         TBLKUP=SLOPE*(ARG-X(I-1))+Y(I-1)                                014630
         RETURN                                                          014640
         END                                                             014650
         SUBROUTINE CLOSUR(CYC,A1,DN)                                    014660
   C                                                                     014670
   C        EQUIVALENCE BETWEEN CRACKS 2  AND  CLOSUR                    014680
   C        CRACKS 2              CLOSUR                                 014690
   C        ASUBP                 AP                                    014700
   C        SMAX                  S                                     014710
   C        CYCLES                CYCLS                                 014720
   C        ISEG                  I                                     014730
   C        J4                    K                                     014740
   C        SMALLN                AN        (NOT USED AS SMALLN ANYWAY) 014750
   C        KSUBQ                 AKC                                   014760
   C        DELKTH                KTH                                   014770
   C        A                     A1                                    014780
   C        AMAX                  AF                                    014790
   C                                                                     014800
   C                                                                     014810
         COMMON/DATA/ EQN,NASA,J1PR,J2PR,J3PR,J4PR,J5PR,AZERO,AF   ,NZERO 014820
         INTEGER EQN                                                     014830
         REAL NZERO                                                      014840
         COMMON/RDATA/ MODEL,RETARD,PLSTRN,OVLD,SIGMAX,SIGMIN,AP,SMALLM  014850
         INTEGER RETARD,PLSTRN                                           014860
         COMMON/MDATA/ MATID(18),C,AN,CARRAY(100),SNARAY(100),KSUBC,     014870
        /              AKC,SIGMAY,   KTH,RMULT,RCUT,OLMAX               014880
         REAL KSUBC                                                      014890
         COMMON/LDATA/   S( 20,10),SMIN( 20,10),CYCLS( 20,10),NLYRS(10), 014900
        /              NBLKS,IBLKS( 50 ),ISEGS( 50 ),NSEGS              014910
         COMMON/STEPS/   K,J1,J2,J3,I,J5,ISTOP,NORTRD                   014920
         COMMON/CORFAC/ ISURF,RATIO,PHI,THICK,IBETA(10),BETA(10),NPTS,   014930
        /              AOVERB(100),BTABLE(100),NPTS2,AOVRB2(100),       014940
        /              BTABL2(100),ASTART(10),ASTOP(10)                 014950
         COMMON/PARIS/C1,SN1,DKCOM,C2,SN2                               014960
         COMMON /DIRECT/ NDADN                                          014970
         COMMON/OUTPOT/XK,XKA,XKEFFN,IFLT,DADNPR                        014980
         COMMON/CLOS/CF,CCOEF,CFEXP,B,BOL,NSAT                         014990
         COMMON/CLOSIC/SC,SPEAK,PRVMX,APEAK,SC1,SC2,SC3,PRVMN           015000
         REAL      ISUM,KTHSG,KCOEF,KEXP,NOL,NSAT,NPR,NPREV,KTH         015010
         DIMENSION Q(6),QQ(2)                                           015020
         DATA PI/3.14159265/                                           015030
   C                                                                     015040
   C                                                                     015050
   C                                                                     015060
   C             CLOSURE FUNCTION                                        015070
         CLOSE(G1,G2) = G1*(CCOEF + (CF - CCOEF)*(1. + G2)**CFEXP)       015080
```

```
C                                                                       015090
C             FUNCTION FOR SC  LT  OR  EQ  SC INITIAL                    015100
      DOWN(Z1,Z2,Z3) = Z1 - (Z1-SC3)*(Z2/Z3)**B                         015110
C                                                                       015120
C             FUNCTIONS FOR SC  GT  SC INITIAL                          015130
      SCONE(SC3) = SC3*BOL                                              015140
C                                                                       015150
C             FUNCTION FOR INCREASING CLOSURE STRESS                    015160
      NPREV(SC,SC3,SC11) =1.+(NSAT-1.)*(SC-SC11)/(SC3-SC11)             015170
C                                                                       015180
C                                                                       015190
C                                                                       015200
C INITIALIZE PARAMETERS                                                 015210
      IGROW = 0                                                         015220
      MODE = 0                                                          015230
      ITEM = 1                                                          015240
      IGROW = 1                                                         015250
      CYSUM = CYCLS(I,K)                                                015260
      NOL = 0.                                                          015270
      ASTRT = A1                                                        015280
      KLU = 1                                                           015290
      R = 0.                                                            015300
    IF ( S(I,K) .NE. 0. )  R = SMIN(I,K)/S(I,K)                         015310
      SMNGR = SMIN(I,K)                                                 015320
    IF ( SPEAK .NE. 0. ) GO TO 30                                       015330
   SINITL = S(I,K)                                                      015340
   IF ( SINITL .LE. 0.) SINITL = 0.05*SIGMAY                           015350
   R = SMIN(I,K)/SINITL                                                 015360
   CALL BETAS(S(I,K),A1,ALP,QE)                                        015370
      XK = SINITL*SQRT(PI*A1)*ALP                                      015380
      G1 = SINITL                                                       015390
      G2 = R                                                            015400
    IF ( G2 .LT. -1.) G2 =-1.                                          015410
      SC = CLOSE(G1,G2)                                                 015420
      SC1 = SC                                                          015430
      SC2 = SC                                                          015440
      SC3 = SC                                                          015450
      SPEAK = SINITL                                                    015460
      PRVMX = SINITL                                                    015470
      PRVMN = SMIN(I,K)                                                 015480
      APEAK = A1                                                        015490
      AP = A1 + RY(XK,PLSTRN)                                          015500
      OMGA2 = AP                                                        015510
   30 CONTINUE                                                          015520
C                                                                       015530
C                                                                       015540
C START ANALYSIS                                                        015550
  100     ISUM =0.                                                      015560
C                                                                       015570
C                                                                       015580
      G1 = S(I,K)                                                       015590
      G2 = R                                                            015600
```

- 119 -

```
        IF ( G2 + 1. ) 104,105,105                              015610
  104     G2 = -1.                                               015620
  105     SC3 = CLOSE(G1,G2)                                     015630
        IF(SMIN(I,K) - PRVMN) 5003,60,60                         015640
C           MINIMUM STRESS ADJUSTMENT                            015650
 5003 CONTINUE                                                   015660
        PRVMN = SMIN(I,K)                                        015670
        G1 = SPEAK                                               015680
        G2 = SMIN(I,K)/SPEAK                                     015690
        IF (G2 + 1.) 5005,5006,5006                              015700
 5005     G2 = -1.                                               015710
 5006     SC1 = CLOSE(G1,G2)                                     015720
        G1 = PRVMX                                               015730
        G2 = SMIN(I,K)/PRVMX                                     015740
        IF (G2 + 1.) 57,58,58                                    015750
   57     G2 = -1.                                               015760
   58     SC3 = CLOSE(G1,G2)                                     015770
        G3 =  ASTRT - APEAK                                      015780
        G4 = AP - APEAK                                          015790
        IF(G3/G4 .LT. 0.) G3=0.                                  015800
        SC2T = DOWN(SC1,G3,G4)                                   015810
        IF(SC2T - SC2) 59,60,60                                  015820
   59     SC2 = SC2T                                             015830
   60     SC = SC2                                               015840
        IF(S(I,K)-SC2) 450,450,5009                              015850
 5009     SMNGR = SC2                                            015860
        IF (SMIN(I,K)-SC2) 5010,5011,5011                        015870
 5010     SMNGR = SC2                                            015880
 5011 CALL BETAS(S(I,K),A1,ALP,QE)                               015890
        XKEFF=(S(I,K)-SMNGR)*SQRT(PI*A1)*ALP                     015900
        CKTH = KTH*(1.-RMULT*R)                                  015910
        IF (R .LT. 0. ) CKTH = KTH                               015920
        IF(XKEFF*(1.-R)/(1.-CF) .LE. CKTH) GO TO 450             015930
 5012     G1 = S(I,K)                                            015940
        G2 = R                                                   015950
        IF (G2 + 1.) 93,94,94                                    015960
   93     G2 = -1.                                               015970
   94     SC3 = CLOSE(G1,G2)                                     015980
        IF ( S(I,K) - SC3 ) 95,96,96                             015990
   95     SC3 = S(I,K)                                           016000
   96 CONTINUE                                                   016010
        IF(S(I,K) - SPEAK) 90,5013,5013                          016020
 5013     KLU = 3                                                016030
C                                                                016040
C            INITIALIZATION FOR INTEGRATION ROUTINE              016050
C                                                                016060
C     MODE = 1    IF  CYCLS(I,K) .LT. 20 AND SC3 .LE. SC         016070
C     MODE = 2    IF  CYCLS(I,K) .LT. 20 AND SC3 .GT. SC         016080
C     MODE = 3    IF  CYCLS(I,K) .GE. 20, SC3.GT.SC AND N.LT.NSAT 016090
C     MODE = 4    IF  CYCLS(I,K) .GE. 20 AND SC3 .LT. SC         016100
C     MODE = 5    IF  CYCLS(I,K) .GF. 20 AND SC3 = SC            016110
C     INTEGRATION PERFORMED FOR MODE = 4 AND 5                   016120
```

```
 90      MODE = 5                                            016130
        IF ( CYCLS(I,K) - 20. ) 201,203,203                 016140
201      MODE = 1                                            016150
        IF ( SC - SC3 ) 202,205,205                         016160
202      MODE = 2                                            016170
        GO TO 205                                            016180
203     IF ( SC .EQ. SC3 ) GO TO 205                        016190
         MODE = 3                                            016200
        IF ( SC3 - SC ) 204,205,205                         016210
204      MODE = 4                                            016220
205 CONTINUE                                                 016230
C                                                            016240
C                                                            016250
        DCYC = CYCLS(I,K)/30000.                             016260
        IF ( DCYC .LT. 1.0 )  DCYC = 1.0                     016270
        NNCYC = CYCLS(I,K)/DCYC                              016280
C                                                            016290
C       START CYCLE - BY - CYCLE ANALYSIS                    016300
C                                                            016310
    DO 310 J = 1,NNCYC                                       016320
        IF ( MODE - 3 ) 210,210,6000                         016330
210     IF (SC3 - SC) 92,92,111                              016340
 92     GO TO (101,160,101,101),KLU                          016350
101     IF (OMGA1) 5014,5015,5015                            016360
5014     OMGA1 = 0.                                          016370
5015    IF (OMGA1/OMGA2 - 1.) 5017,5017,5016                 016380
5016     OMGA1 = OMGA2                                       016390
5017     IF (OMGA1/OMGA2 .LT. 0.) OMGA1 =0.                  016400
        SC = DOWN(SC2,OMGA1,OMGA2)                           016410
111     IF(SMIN(I,K) - SC) 5023,160,160                     016420
5023     SMNGR = SC                                          016430
160 CONTINUE                                                 015440
        GO TO 4050                                           016450
4051 CONTINUE                                                016460
        ISUM = ISUM + DCYC                                   016470
        CYC = CYC + DCYC                                     016480
        CYSUM = CYSUM - DCYC                                 016490
        DN = CYSUM                                           016500
        GO TO (138,501,501,501),ITEM                         016510
138 CONTINUE                                                 016520
        IF ( MODE - 2 ) 143,144,144                          016530
143      OMGA1 = A1 - ASTRT                                  016540
        IF (A1 - AP) 300,141,141                             016550
141      KLU = 2                                             016560
         SC = SC3                                            016570
         AP = A1                                             016580
        GO TO 300                                            016590
144      SC11 = SCONE(SC3)                                   016600
         NPR = NPREV(SC,SC3,SC11)                            016610
        IF ( NPR ) 145,146,146                               016620
145      NPR = 0.                                            016630
146      NOL = NPR + DCYC                                    016640
```

- 121 -

```
                  SC = SC11 + (SC3-SC11)*(NOL-1.)/(NSAT-1.)           016650
            IF ( SC3 - SC ) 147,147,300                                016660
    147     SC = SC3                                                   016670
            IF ( MODE - 2 ) 300,300,148                                016680
    148     MODE = 5                                                   016690
    300 CONTINUE                                                       016700
    310 CONTINUE                                                       016710
    311 CONTINUE                                                       016720
            PRVMX = S(I,K)                                             016730
            SC2 = SC                                                   016740
          GO TO (501,400,400,400),KLU                                  016750
    400 CALL BETAS(S(I,K),A1,ALP,QE)                                   016760
            XK = S(I,K)*SQRT(PI*A1)*ALP                                016770
            AP = A1 + RY(XK,PLSTRN)                                    016780
            APEAK = A1                                                 016790
            SPEAK = S(I,K)                                             016800
            PRVMN = SMIN(I,K)                                          016810
            IF (SC3 - SC) 420,420,430                                  016820
    420     SC1 = SC3                                                  016830
            GO TO 501                                                  016840
    430     SC1 = SC                                                   016850
            GO TO 501                                                  016860
    450     ISUM = ISUM + CYCLS(I,K)                                   016870
            CYC = CYC + CYCLS(I,K)                                     016880
            DN = 0.                                                    016890
    501 CONTINUE                                                       016900
    C                                                                  016910
          GO TO(500,600,620,503),ITEM                                  016920
    503     ISTOP = 2                                                  016930
            A1 = ATRANS/(2.*RATIO)                                     016940
          WRITE(6,1) A1,CYC                                            016950
      1 FORMAT(1H0, 70(1H*)/ 5X,55HTRANSITION TO A THRU CRACK OF EFFECTIVE016960
      1 LENGTH, AEFF =  ,F9.5,4H AT ,F12.2,7H CYCLES  /,1X, 70(1H*))   016970
            GO TO 50                                                   016980
    C                                                                  016990
    C                                                                  017000
    C     END OF CYCLE BY CYCLE ROUTINE                                017010
    C                                                                  017020
    500 CONTINUE                                                       017030
          GO TO 50                                                     017040
    C        A1 EXCEEDED AF  (AMAX)                                    017050
    600     ISTOP = 1                                                  017060
            GO TO 50                                                   017070
    620     ISTOP = 1                                                  017080
          WRITE(6,7)                                                   017090
      7 FORMAT(1H0, 70(1H*)/ 5X,46HKMAX APPLIED EXCEEDS KSUBQ. PROBLEM TER017100
      1MINATED/1X, 70(1H*)/1H0, 1X,26HLAST CALCULATED VALUES ARE///)   017110
            DELTAK = XK*(1.-R)                                         017120
          WRITE(6,8)    J1,J2,I,IFLT,K,CYC,A1,XK,XKEFF,DELTAK,DADN     017130
      8 FORMAT( 5X,18HBLOCK IN SPECTRUM ,I4/                           017140
      /         5X,18HSEGMENT NUMBER    ,I4/                           017150
      /         5X,18HMISSION NUMBER    ,I4/                           017160
```

- 122 -

```
        /       5X,18HFLIGHT NUMBER     ,I4/                     017170
        /       5X,18HLAYER IN MISSION  ,I4/                     017180
        5       5X,18HACCUMULATED CYCLES,E16.8/                  017190
        6       5X,18HCRACK LENGTH       ,E16.8/                 017200
        7       5X,18HKMAX APPLIED      ,E16.8/                  017210
        8       5X,18HKMAX EFFECTIVE    ,E16.8/                  017220
        9       5X,18HDELTA K           ,E16.8/                  017230
        /       5X,18HDA/DN             ,E16.8)                  017240
C                                                               017250
C                                                               017260
   50 CONTINUE                                                   017270
        XKA = XK                                                 017280
      RETURN                                                     017290
C                                                               017300
C                                                               017310
C          GROWTH CALCULATIONS                                  017320
 4050 CONTINUE                                                   017330
C          CHECK FOR XK (KMAXA) .GE. AKC (KSUBQ)                017340
      CALL BETAS(S(I,K),A1,ALP,QE)                               017350
        XK = S(I,K) *SQRT(PI*A1)*ALP                            017360
      IF(XK - AKC) 4062,4060,4060                                017370
 4060    ITEM = 3                                               017380
         DADN = 0.                                               017390
       GO TO (502,502,502,4062,4062),MODE                       017400
 4062    XKEFF = (S(I,K) - SMNGR)*SQRT(PI*A1)*ALP               017410
         XKEFFN = XKEFF/(1.-CF)                                  017420
      IF ( FQN .EQ. 1 ) GO TO 9000                               017430
      IF ( EQN .EQ. 4 ) GO TO 9010                               017440
      IF ( EQN - 2 ) 8000,8000,8010                              017450
 8000 CONTINUE                                                   017460
C          BI-LINEAR PARIS EQUATION                             017470
         C = C1                                                  017480
         AN = SN1                                                017490
      IF ( XKEFFN .GE. DKCOM ) C = C2                            017500
      IF ( XKEFFN .GE. DKCOM ) AN = SN2                          017510
         DADN = C*XKEFFN**AN                                     017520
         GO TO 3020                                              017530
C                                                               017540
 8010 CONTINUE                                                   017550
C          TABULAR RATE VALUES                                  017560
         DADN = TBLKUP(CARRAY,SNARAY,NDADN,100,XKEFFN)          017570
         DADN = 10.**DADN                                        017580
 8020 CONTINUE                                                   017590
      IF(RETARD.NE.0)DADNPR = DADN                               017600
         A1 = A1 + DADN*DCYC                                     017610
      IF (A1 - AF)4064,4063,4063                                 017620
 4063    ITEM = 2                                                017630
       GO TO 502                                                 017640
 4064 IF ( ISURF .EQ. 0 ) GO TO 502                              017650
         ATRANS = THICK - (((XK/SIGMAY)**2.)/(2.*PI))           017660
      IF ( A1 .LT. ATRANS) GO TO 502                             017670
         ITEM = 4                                                017680
```

- 123 -

```
       502   GO TO (4051,6060),IGROW                              017690
C                                                                 017700
C                                                                 017710
C            END OF CRACK GROWTH CALCULATIONS                     017720
C                                                                 017730
C                                                                 017740
C                                                                 017750
C            INTEGRATION ROUTINE                                  017760
 6000 CONTINUE                                                    017770
       IF ( MODE - 4 ) 6010,6010,6020                             017780
 6010    Q(1) = A1                                                017790
         Q(2) = AP                                                017800
         Z2 = A1 - ASTRT                                          017810
         Z3 = AP - ASTRT                                          017820
         IF ( Z2/Z3 .LT. 0.) Z2 =0.                              017830
         QQ(1) = DOWN(SC2,Z2,Z3)                                 017840
         QQ(2) = SC3                                              017850
       IF ( ISURF .EQ. 0 )  GO TO 6030                           017860
       IF ( Q(2) .LT. THICK )  GO TO 6030                        017870
         Q(2) = THICK                                             017880
         Z2 = THICK - ASTRT                                      017890
         IF (Z2/Z3 .LT. 0.) Z2 =0.                               017900
         QQ(2) = DOWN(SC2,Z2,Z3)                                 017910
         ISTOP = 2                                                017920
       GO TO 6030                                                 017930
 6020    Q(1) = A1                                                017940
         Q(2) = 1.10*A1                                           017950
         QQ(1) = SC3                                              017960
         QQ(2) = SC3                                              017970
       IF ( ISURF .EQ. 0 )  GO TO 6030                           017980
       IF ( Q(2) .LT. THICK )  GO TO 6030                        017990
         Q(2) = THICK                                             018000
         ISTOP = 2                                                018010
 6030 CONTINUE                                                    018020
         SC = QQ(1)                                               018030
         DO 6100 KK = 1,2                                         018040
         A1 = Q(KK)                                               018050
         SMNGR = SMIN(I,K)                                        018060
       IF ( SMIN(I,K) - QQ(KK) ) 6045,6050,6050                  018070
 6045    SMNGR = QQ(KK)                                           018080
 6050    IGROW = 2                                                018090
         GO TO 4050                                               018100
 6060    GO TO (6070,6065,6065,6065),ITEM                        018110
 6065    GO TO (501,6068),KK                                     018120
 6068    ITEM = 1                                                 018130
         ISTOP = 0                                                018140
 6070    KKK = KK + 2                                             018150
         KKKK = KK + 4                                            018160
         Q(KKK) = XKEFF                                           018170
         Q(KKKK) = DADN                                           018180
 6100 CONTINUE                                                    018190
         Q2 = (Q(3)-Q(4))/(Q(1)-Q(2))                            018200
```

```
          Q1 = Q(3) - Q2*Q(1)                                      018210
          Q4 = (ALOG(Q(5)/Q(6)))/(ALOG(Q(3)/Q(4)))                018220
          Q3 = Q(5)/(Q(3)**Q4)                                     018230
          Q5 = 1. - Q4                                             018240
          DELTN = (Q(4)**Q5 - Q(3)**Q5)/(Q2*Q3*Q5)                018250
       IF ( DELTN - CYSUM ) 6200,6150,6250                        018260
 6150 CONTINUE                                                     018270
 6200     A1 = Q(2)                                                018280
       IF ( AP - A1 ) 6210,6210,6220                              018290
 6210     AP = A1                                                  018300
          KLU = 2                                                  018310
          MODE = 5                                                018320
 6220     ISUM = ISUM + DELTN                                      018330
          CYC = CYC + DELTN                                        018340
          CYSUM = CYSUM - DELTN                                    018350
          DN = CYSUM                                               018360
       IF ( CYSUM - .01 ) 6260,6240,6240                          018370
 6240  IF ( MODE - 4 ) 6010,6010,6020                             018380
 6250     Q(4) = (Q(3)**Q5+CYSUM*Q2*Q3*Q5)**(1./Q5)              018390
          Q(2) = (Q(4) -Q1)/Q2                                    018400
          DELTN = CYSUM                                            018410
          ISTOP = 0                                                018420
       GO TO 6150                                                  018430
 6260     SC = SC3                                                 018440
 6270     Z2 = A1 - ASTRT                                          018450
          Z3 = AP - ASTRT                                          018460
       IF (Z3 .LE. 0. ) Z3 = Z2                                   018470
        IF (Z2/Z3 .LT. 0.) Z2 =0.                                 018480
          SC = DOWN(SC2,Z2,Z3)                                     018490
 6280  GO TO 311                                                   018500
C                                                                  018510
C                                                                  018520
C          END OF INTEGRATION ROUTINE                              018530
C                                                                  018540
C                                                                  018550
C          DIAGNOSTICS FOR IMPROPER EQN                            018560
 9000 WRITE(6,9005)                                                018570
 9005 FORMAT(1H0, 1X,44HCLOSURE MODEL CAN NOT ACCEPT FORMAN EQUATION /  018580
     1    40X,20HEXECUTION SUPPRESSED )                            018590
       GO TO 9020                                                  018600
 9010 WRITE(6,9015)                                                018610
 9015 FORMAT(1H0, 1X,44HCLOSURE MODEL CAN NOT ACCEPT WALKER EQUATION /  018620
     1    40X,20HEXECUTION SUPPRESSED )                            018630
 9020     ISTOP = 1                                                018640
          XKA = XK                                                 018650
       RETURN                                                      018660
C                                                                  018670
C                                                                  018680
       END                                                         018690
       SUBROUTINE RUNKUT(X,Y,DX,F)                                 018700
C                                                                  018710
       EXTERNAL F                                                  018720
```

```
        COMMON/STEPS/ISEG,J1,J2,J3,J4,J5,ISTOP,NORETRD              018730
 10 XO=X                                                            018740
    X=X+DX                                                          018750
    H=DX                                                            018760
 20 IF(ABS(H).GT.ABS(X-XO)) H=X-XO                                  018770
 30 YO=Y                                                            018780
    HT=H                                                            018790
    XT=XO                                                           018800
    RMAXP=1.E37                                                     018810
 40 YT=YO                                                           018820
    ASSIGN 50 TO K                                                  018830
    GO TO 100                                                       018840
 50 CONTINUE                                                        018850
 60 YP=Y                                                            018860
 70 HT=0.5*H                                                        018870
    ASSIGN 80 TO K                                                  018880
    GO TO 100                                                       018890
 80 CONTINUE                                                        018900
 90 YT=Y                                                            018910
    XT=XO+HT                                                        018920
    ASSIGN 150 TO K                                                 018930
100 CALL F(XT,YT,PO)                                                018940
    IF(ISTOP.EQ.0) GO TO 110                                        018950
    X = XT                                                          018960
    Y = YT                                                          018970
    RETURN                                                          018980
110 Y=YT+0.5*HT*PO                                                  018990
    CALL F(XT+0.5*HT,Y,P1)                                          019000
    IF(ISTOP.EQ.0) GO TO 120                                        019010
    X = XT+0.5*HT                                                   019020
    RETURN                                                          019030
120 Y=YT+0.5*HT*P1                                                  019040
    CALL F(XT+0.5*HT,Y,P2)                                          019050
    IF(ISTOP.EQ.0) GO TO 130                                        019060
    X = XT+0.5*HT                                                   019070
    RETURN                                                          019080
130 Y=YT+HT*P2                                                      019090
    CALL F(XT+HT,Y,P3)                                              019100
    IF(ISTOP.EQ.0) GO TO 140                                        019110
    X = XT+HT                                                       019120
    RETURN                                                          019130
140 Y=YT+HT*(PO+2.*(P1+P2)+P3)/6.                                   019140
    GO TO K,(50,80,150)                                             019150
150 RMAX=0.                                                         019160
160 RMAX=AMAX1(RMAX,0.07*ABS((Y-YP)/Y))                             019170
    IF((RMAX.GT.1.E-07).AND.(RMAX.LT.RMAXP)) GO TO 170              019180
    XO=XO+H                                                         019190
    IF(XO.EQ.X) RETURN                                              019200
    IF((RMAX.LT.1.E-08). OR.(RMAX.GT.RMAXP)) H=H+H                  019210
    GO TO 20                                                        019220
170 H=HT                                                            019230
    XT=XO                                                           019240
```

```
       180  YP=YT                                                        019250
       190  YT=YO                                                        019260
            RMAXP=RMAX                                                   019270
            GO TO 70                                                     019280
            END                                                          019290


                            Sample Input


       TITLE
          1
       LOW-MED-HIGH SPECIMEN 84-502, EE3, WHEELER M=6.0, 1 CYCLE=20 SEC
       EQUATION
       SIGMOID
       MATERIAL
       INCO 718 MSE FIT DA/DT TIMES 20
       21.0       272.73      -5.6942677-1.1      1.8         -1.8
       300.0      120.0
       THRESHOLD
       21.0       1.0
       LIMITS
       .4087      1.1330      0.         0.0
       ANALYSIS
       RETARD     1.0         0.0        6.0         1.0
       BETA       9.0         1.5736     .394
       END
       LOADS
          1     0 PROOF TEST SPECTRUM
       MAX-MIN    SUSTAINED LOAD EQUIVALENT OF 1 CYCLE = 20 SEC R=.4
                  2.740       1.096      1.
       END
       MAX-MIN    OVERLOAD OF 20 PERCENT EQUIVALENT CYCLE
                  3.288       0.0        1.
       END
       END LOADS
       SPECTRUM
          7     0
         72     1
          1     2
       1573     1
          1     2
        590     1
          1     2
       1000     1
       RESTART
          1
       PRINT
          1     0     0     1
       END DATA
```

Vita

Robert L. Hastie Jr. was born on 18 November 1957 in Harrison, Pennsylvania. He graduated from Kiski Area high school in Vandergrift, Pennsylvania in 1975 and attended Grove City College. In, May 1979 he graduated Cum Laude and received a Bachelor of Science degree in Mechanical Engineering. Upon graduation, he received a commission in the USAF through the ROTC program. He was called to act' e duty in August 1979 and served as a Structural Strength Engineer in the Deputy for Engineering at the Aeronautical Systems Division (ASD) at Wright-Patterson AFB, Ohio. In November 1980 he was transfered to the Deputy for Propulsion within the ASD and was a Propulsion Durability Engineer for the F107, F100, TF-34, and F100 engines. In June 1984, Captain Hastie entered the School of Engineering, Air Force Institute of Technology, to earn a Master of Science degree in Astronautical Engineering.

Permanent address: RD 1 Box 295

Vandergrift, Pa   15690

AD-A114-08

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION	1b. RESTRICTIVE MARKINGS
UNCLASSIFIED	

2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
	APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	

4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFIT/GA/AA/85D-6	

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
School of Engineering	AFIT/ENY	

6c. ADDRESS (City, State and ZIP Code)	7b. ADDRESS (City, State and ZIP Code)
Air Force Institute of Technology	

8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State and ZIP Code)	10. SOURCE OF FUNDING NOS.			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
11. TITLE (Include Security Classification) See Box 19	61102F	2302	P1	01

12. PERSONAL AUTHOR(S)
Robert L. Hastie Jr., B.S., Captain, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
MS THESIS	FROM _____ TO _____	1985 December	139

16. SUPPLEMENTARY NOTATION

17.	COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	Sustained-load, Crack Growth Rate, Crack Growth Retardation Elevated Temperature, Nickel-base Superalloy, CRACKS, Crack Growth Prediction Model, Wheeler, Willenborg
20	11		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

TITLE: PREDICTING THE EFFECTS OF OVERLOADS ON SUSTAINED-LOAD CRACK GROWTH IN A HIGH-TEMPERATURE SUPERALLOY

THESIS CHAIRMAN: George K. Haritos, Major, USAF

ABSTRACT: See Reverse

Approved for public release: IAW AFR 190-1/.

~~Lynn~~ E. V. ~~OLAVER~~
Dean for Research and Professional Development
Air Force Institute of Technology (RTG)
Wright-Patterson AFB OH 45433
16 JAN fL

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
George K. Haritos, Associate Professor	(513)-255-2998	AFIT/ENY

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE.

Abstract

thesis

^ This ~~study~~ investigates methods of modeling the effects
of overloads on high-temperature sustained-load crack
growth. In addition to a model previously developed for
this specific problem, a computer program developed for
low-temperature, high-frequency cyclic load applications was
evaluated. Sustained-load hold times were converted to
equivalent fatigue cycles to analyze a load spectrum,
consisting of sustained-load with periodic overloads. The
CRACKS crack growth program was used with the Wheeler and
Willenborg models used to account for crack growth
retardation due to overloads.

Predictions were compared with experimental test data
generated on specimens of Inconel 718 at 650 C with periodic
overloads of either 20 or 50 percent. Crack measurements
were made using a electric potential system. The
application of e electric potential system to crack growth
measurement following overloads was extensively evaluated.
It was concluded that the system had to be recalibrated
after each overload due to a sudden advancement in crack
length. ⟵

The retardation models were found to require empirical
parameters that depend upon the stress intensity level for
each overload application. Using relationships developed
for these parameters, the CRACKS program using the Wheeler
model was found to be capable of predicting the
time-to-failure for sustained-loading with periodic
overloads within 20 percent of test data. The Willenborg
model was found to be inapplicable to this problem because
it depends solely on stress ratio which has no physical
meaning for sustained-loading. The Wheeler model, on the
other hand, could generally be applied to sustained-load
crack growth using equivalent fatigue cycles. In
conjunction with the CRACKS computer program, this could
provide a powerful new method for evaluating crack growth
under general engine mission spectra including the effects
of overloads.

END

FILMED

3-86

DTIC