Modeling Cycle and Time Dependent Creep/Relaxation Effects on Fatigue Lives of Notched Members

by

DeRome Osmond Dunn

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Engineering Science and Mechanics

APPROVED:

N. E. Dowling, Chairman

R. W. Hendricks

J. N. Reddy

K. L. Reifsnider

C. W. Smith

September 1991

Blacksburg, Virginia

Modeling Cycle and Time Dependent Creep/Relaxation Effects on Fatigue Lives of Notched Members

by

DeRome Osmond Dunn

Committee Chairman: Norman E. Dowling

Engineering Science and Mechanics

(ABSTRACT)

Mechanical cyclic variations in mean stress and strain amplitude is a well-known occurrence for metals even at room temperature. Many fatigue analysis procedures ignore these variations. Fatigue analysis which included both time and cycle dependent mechanical material behavior for metals at room temperature had not been previously studied except for the case of creep. An investigation studying transient mechanical effects on Ti-6Al-4V titanium and 7475-T651 Al alloys was done to determine how great an effect transients at room temperature would have on fatigue life under cyclic conditions. The mechanical material response was modeled using viscoplasticity constitutive laws and Neuber's rule eliminating the need for finite element modeling of uniaxially loaded notched members. However, the Neuber's modeling may be used with any material constitutive law. The procedures for fatigue damage used cycle counting to compute strain amplitude and mean stress. Since a large amount of fatigue data is

reported as strain-life curves, the fatigue analysis was developed using this fatigue data although it did not include transients. If favorable results are obtained, development of modeling and testing to include transients in strain-life fatigue data could be avoided, and the existing fatigue data base utilized.

Experimental work was undertaken and nonlinear optimization techniques used to compute model constants for the two alloys. However, small amounts of rate dependence was found for cyclic strain control testing. The viscoplasticity models became stiff when rate dependence was low causing numerical problems, and model constants for the viscoplastic constitutive law could not be determined since convergence was not achieved. Also, only small amounts of transient static stress relaxation was observed for extended hold periods.

Finally, experimental verification was done for the local surface stresses in a notched member under load using advanced x-ray stress equipment. Measurements during brief pauses were made over a cycle. From the x-ray results, an anomalous surface behavior was observed. The surface yielded before the bulk material with the lower surface yielding seeming to be time dependent in nature. Since rolled plates of the alloys were used, texture was measured and studied in the form of pole figures, and extreme texture was found for both alloys. However, successful x-ray measurements were made for the alloys studied even though assuming linear d-spacing versus $sin^2\psi$. Finally, x ray measurements for a cycled notched member, exhibited relaxation of mean stress and not relaxation of residual stress.

ACKNOWLEDGMENT

The work described was performed by the Engineering Science and Mechanics Department of Virginia Polytechnic Institute and State University for the Naval Air Development Center under contract number N62269-85-C-0256. The principal investigator was Prof. Norman E. Dowling; the program manager for NADC was L. W. Gause, and the project engineer was R. E. Vining.

TABLE OF CONTENTS

TABLE OF CONTENTS CONTINUED

LIST OF FIGURES

LIST OF FIGURES CONTINUED

LIST OF FIGURES CONTINUED

LIST OF FIGURES CONTINUED

LIST OF TABLES

1.0 INTRODUCTION

Studies have not been done to consider both mechanical time and cycle dependent effects on fatigue life of notched members for metals at room temperature except in the case of time dependent creep when hold periods occurred during cycling. At room temperature for metals, most prevalent fatigue analysis ignores these variations' effect on fatigue life. In this study, an attempt was made to account for mean stress variation along with strain amplitude variation by using a material mechanical model including both cyclic ratcheting and time dependent material behavior at room temperature. These variations were included in the fatigue damage computations to determine their effect on fatigue life during uniaxial cyclic loading of notched members for metals at room temperature. Creep damage was not included in the life determination.

Mechanical constitutive laws just to model time and cycle dependent transients for metals at room temperature have not been developed. However, this study seeks to apply the viscoplastic constitutive laws previously developed to model mechanical transients for metals at elevated temperatures to metals at room temperature.

Walker's unified viscoplastic constitutive model was used to model the mechanical material response for uniaxial loading of metals at room temperature. Numerical difficulties were later encountered because the two materials chosen, Ti-6Al-4V titanium and 7475-T651 aluminum alloys, exhibited very little cyclic rate dependence during cyclic variation leading to an inability to determine material constants for these materials. The viscoplastic models do not reduce to a rate independent form, and instead become numerically stiff which means variations both very small and large in values are computed simultaneously leading to the nonconvergence of iterative routines. Also, static relaxation tests were performed to further evaluate the rate dependence of the two metals chosen at room temperature. The relaxation tests indicated the presence of fairly small amounts of time dependent mechanical response at room temperature for the two alloys chosen.

The significance and magnitude of the influence of mechanical transients on fatigue was of interest allowing comparisons with fatigue analysis which based their predictions on stable cyclic mechanical behavior of metals at room temperature. Also of interest, was the ways in which transient material response may affect fatigue life. The model would allow the study of transient mechanical variations in mean stress and strain amplitude since mean stress and strain amplitude variations are included. Finally, how transient mean stress and strain amplitude variations occur in a notch are described

when the notch effects are supplied by Neuber's rule where the use of Neuber's rule with stable cyclic analysis was extended for use with mechanical transient material response. Using Neuber's rule allowed the use of a one-dimensional mechanical stress-strain analysis for the case of uniaxially loaded notched members avoiding the use of two or three-dimensional finite element computations.

Procedures and methods for cycle counting and cyclic fatigue damage determination have not been previously developed when mechanical transients were present. Procedures are outlined for prediction of fatigue life once local notch mechanical cycle and time dependent behavior was modeled. Considerations are presented for using the local strain approach with Neuber's rule under these conditions. Also presented is rainflow cycle counting of the resulting local stress-strain history which was done to determine strain amplitude and mean stress. Only the local peak-valley values for the stress-strain history was used to reduce the amount of numerical data needed during the damage calculations. Also the fatigue damage calculations were devised to use strain-life curves even though these curves are compiled ignoring the presence of time and cycle dependent transients to draw upon the large amount of fatigue data available in this form and to avoid the use of more elaborate fatigue testing.

Finally, x-ray stress analysis using advanced equipment was undertaken to monitor local stress variations at the notch root during

cycling. The measurements were made during brief pauses during cycling. These types of measurements during loading previously could not be done with prior x-ray stress instrumentation. The x-ray results were gathered to validate the model developed. However, an unexpected lower than bulk surface yield point was observed which seemed to be time dependent in nature. Also, pole figures were compiled to quantify the texture of the rolled plates used. Extreme texture was noted, but the linear d-spacing versus $sin^2\psi$ relationship was still found to apply for the alloys studied in the presence of the extreme texture. Finally, x-ray measurements of the cyclic variation of residual stress was done to determine if relaxation of mean stress or relaxation of residual stress occurs as a result of cycling a notched member. No definitive answer to this question was possible with prior x-ray stress instrumentation.

2.0 LITERATURE REVIEW

2.1 CYCLE AND TIME DEPENDENT TRANSIENTS AT ROOM TEMPERATURE

Cycle and time dependent transient responses are seen in the stress-strain response of metals. Time dependent relaxation of stress is a well-known phenomenon in metals in the strained state. This phenomenon has been observed even at room temperature. The time dependent transient of low temperature creep has been previously seen in wrought Ti-6Al-4V: This behavior was described in Reference [1] at less than 250° Fahrenheit and occurs even below the elastic limit. The transient of cycle dependent relaxation was observed for SAE 4340 steel and was discussed in Reference [2]. These and other transients are to be described in the following discussion.

One widely observed transient is that of cyclic hardening and/or softening which causes variations in stress/strain amplitude. The strain amplitude changes during strain control cyclic testing, whereas the stress amplitude changes during stress control cyclic testing. Both stress and strain amplitude changes occur during Neuber control cyclic testing. These effects were previously documented in References [3, 4].

Cyclic hardening is illustrated in Figure 1. The plots were derived from data for a 7475-T651 aluminum alloy obtained from measurements in this study. Plotted are the stress-time, strain-time, and stress-strain responses. Fully reversed strain control testing was done, and the strain-time response was constant as expected. However, for the stress-time response, the stress slightly increased with time. Also, the stress amplitude of the hysteresis loops are seen to vary during the test.

Cyclic softening is illustrated in Figure 2. The plots were derived from data for Ti-6Al-4V titanium alloy. The Ti-6Al-4V data were also obtained in this study. As before, plotted are the stress-time, strain-time, and stress-strain responses for this alloy. Again, fully reversed strain control testing was done. As expected, the strain-time response was fairly constant. However, for the stress-time response, the stress decreased with time. Also, the stress amplitude of the hysteresis loops is seen to vary during the test. Next, the path traced from zero to the first reversal point, the monotonic tension curve, is clearly visible. The monotonic tension curve's path also clearly indicates that the stress amplitude decreased from its initial value.

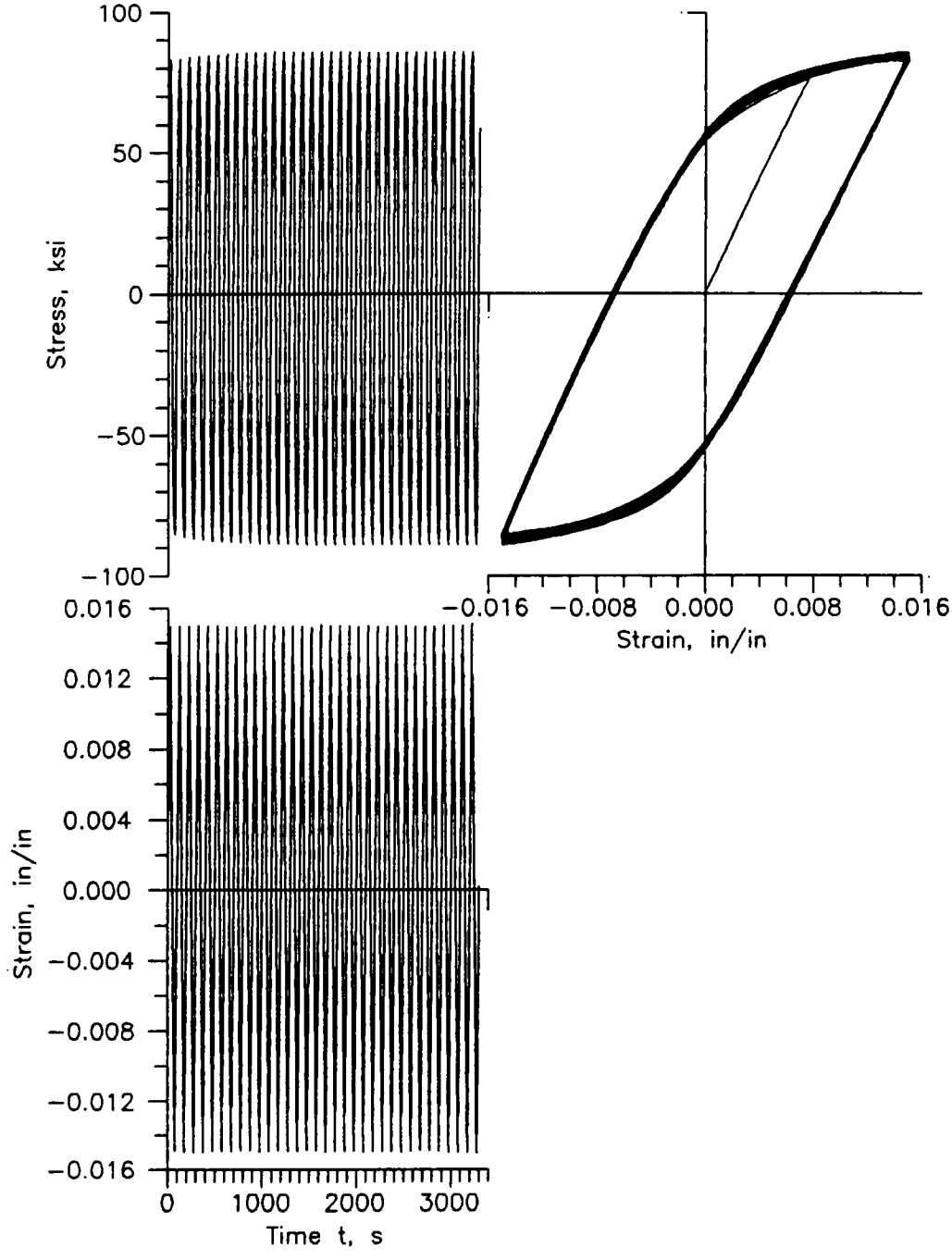


Figure 1. Cyclic hardening 7475-T651 Al, uniaxial strain control test, specimen A2A06C.

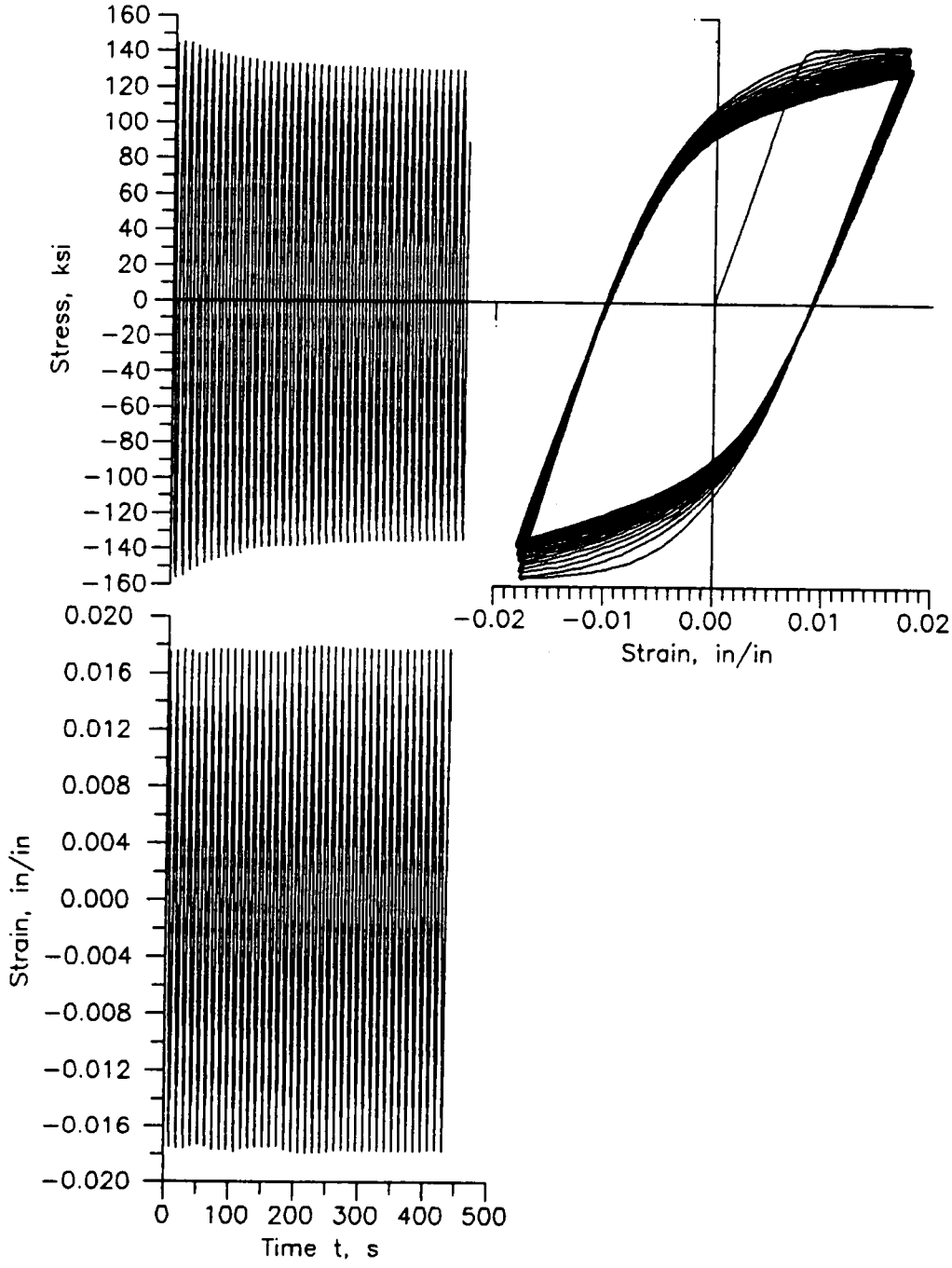


Figure 2. Cyclic softening Ti-6Al-4V, uniaxial strain control test, specimen T1A05A.

Cycle dependent creep/relaxation and time dependent creep/relaxation transients are observed in mechanical material behavior. Cycle dependent transients are referred to as cyclic ratcheting. Cyclic ratcheting is assumed to be completely independent of time, and it is only necessary to plastically cycle the material to produce the effect. Then, time dependent transients vary with time. Therefore, the time dependent transient does not require cycling; however, cycling may produce a time dependent transient since cycling depends on time. The time/cycle dependent behaviors will be observed as creep and/or relaxation. Creep is the occurrence of transient variations in strain resulting in a change in the mean strain over time for a constant load or with cycling for cyclic loading. Relaxation is when transient variations occur in stress so that the mean stress varies with time for a constant displacement or with cycling for cyclic displacement.

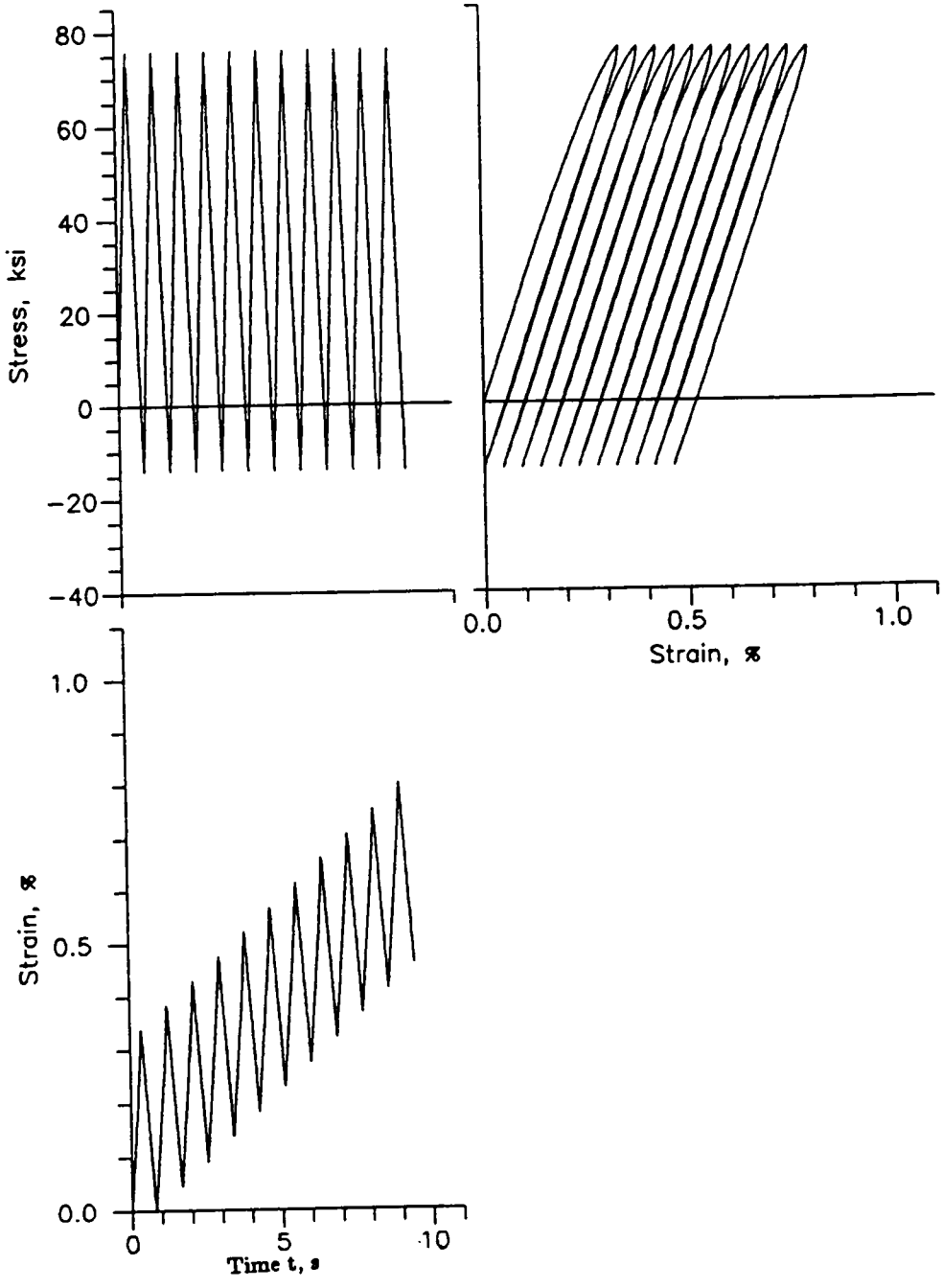
These combinations of mechanical behaviors are illustrated in Figures 3, 4, and 5 that were produced from computer simulations for the case of uniaxial loading. The simulated material exhibits transients due to time dependent and cyclic ratcheting effects. Cyclic hardening was excluded. The material constants were from Reference [5]. The simulations were for the three cases of strain, stress and Neuber control.

The computer model was based on the integral model given by Walker in Reference [5]. Walker's model was simplified for the uni-

axial case. Neuber's rule was included with Walker's model to simulate the stress-strain response of a notched member with local plasticity.

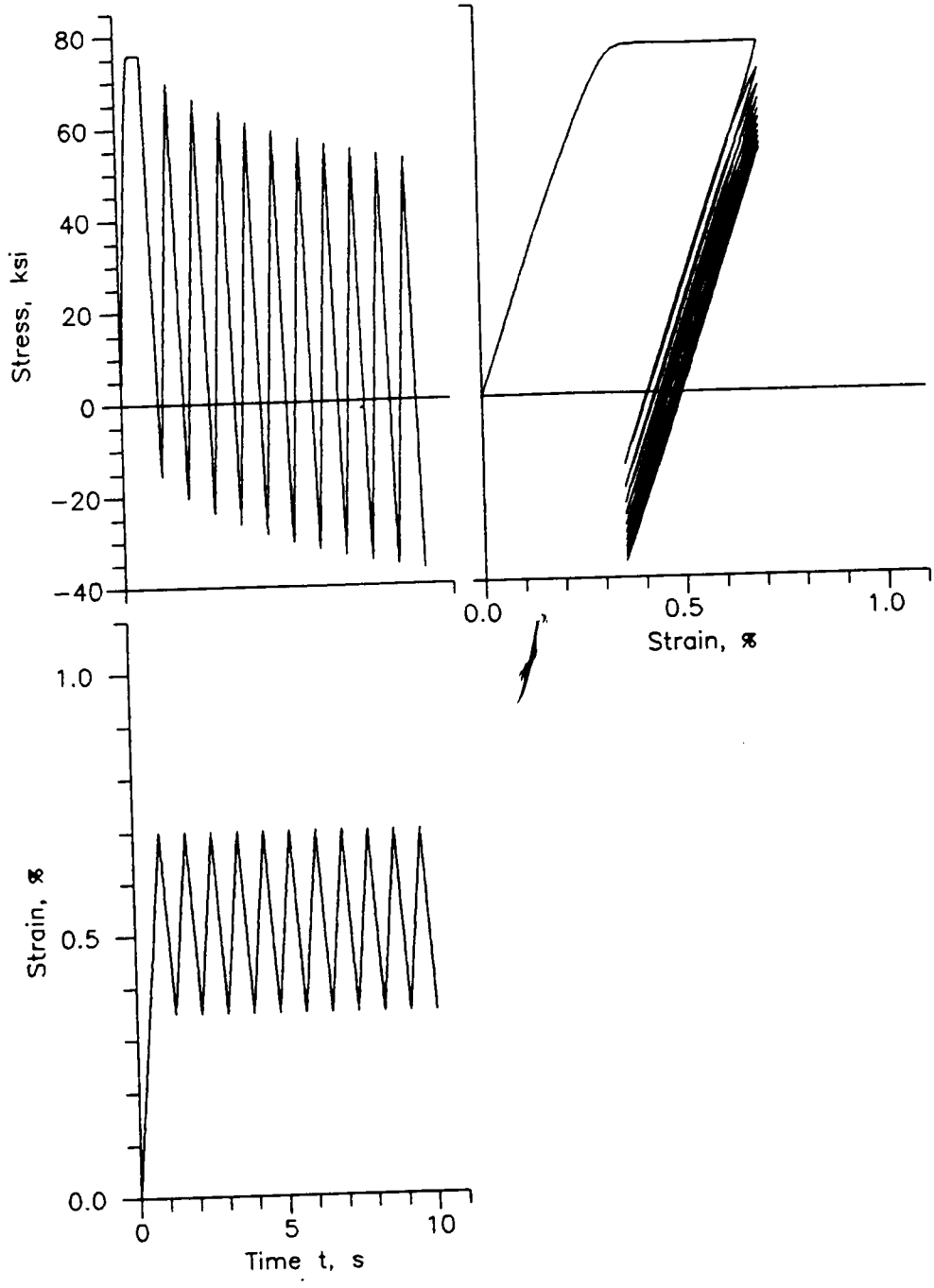
The computer program and the program's input files are in the Appendix that produced the results of Figures 3, 4, and 5. Samples of the output files, for the computer runs, are also given in the Appendix.

The transients of creep/relaxation may be due to the same phenomena. The difference in the observed stress-strain response may be attributed to the different loading conditions. As seen in Figure 3, creep occurs under stress control conditions. Relaxation, as in Figure 4, takes place during the imposition of strain control. In imposing the stress or strain, its time variation was controlled. No time variation produces the special case of static loading. Static loading, such as hold periods, results in purely time dependent response. However, time varying loading, without hold periods, results in cycle dependent response which may be either time dependent or independent. Therefore, cyclic creep/relaxation may be either time dependent or independent. If time dependent variations are significant, loading rate must be considered. However, if time independent variations occur, cyclic variations occur which are independent of the rate and time variation of the loading. Most current fatigue analysis does not include information on the time variation of the load history with time dependency considered to be of secondary importance.



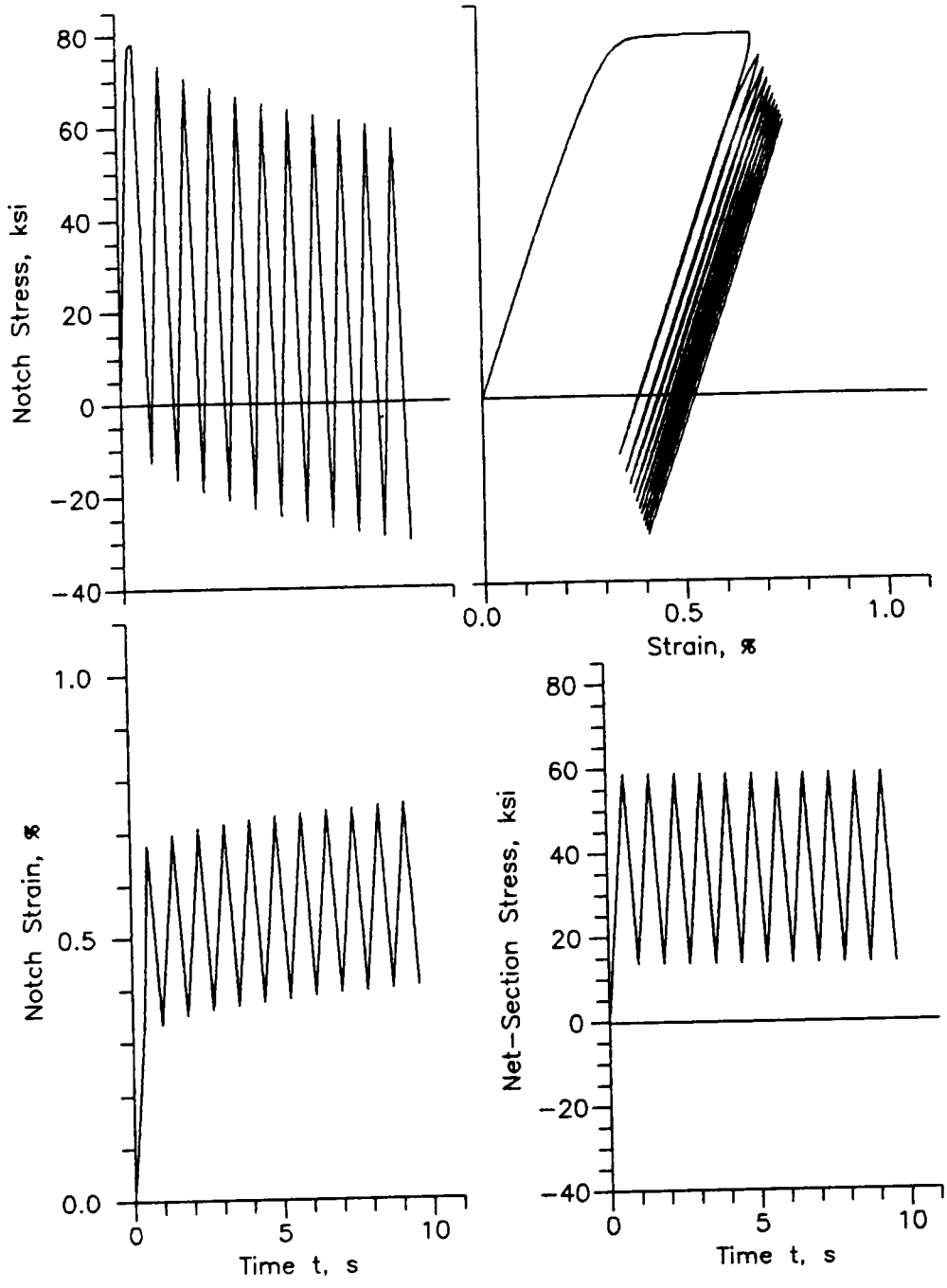
$R = \frac{\sigma_{min}}{\sigma_{max}} = 0.183$ $\sigma_{max} = 75.8$ ksi $\dot{\sigma} = 207 \frac{\text{ksi}}{\text{s}}$ $\dot{\epsilon}_e = \frac{\dot{\sigma}}{E} = 0.799 \frac{\%}{\text{s}}$

Figure 3. Computer simulated stress control including time/cycle dependent material transients.



$R = \frac{\epsilon_{min}}{\epsilon_{max}} = 0.5$ $\epsilon_{max} = 0.7\ \%$ $\dot{\epsilon} = 0.8\ \frac{\%}{s}$ $\dot{\sigma} = E\dot{\epsilon} = 207\ \frac{ksi}{s}$

Figure 4. Computer simulated strain control including time/cycle dependent material transients.



$$R = \frac{S_{min}}{S_{max}} = 0.2355 \qquad S_{max} = 58.6 \text{ ksi} \qquad \dot{S} = 104 \ \frac{\text{ksi}}{\text{s}} \qquad k_t = 2 \qquad \dot{\epsilon}_e = \frac{k_t \dot{S}}{E} = 0.802 \ \frac{\%}{\text{s}}$$

Figure 5. Computer simulated Neuber control including time/cycle dependent material transients.

Finally, both time dependent and independent effects may contribute together to the total mechanical cyclic creep/relaxation response.

Cycle dependent responses were viewed as not being due to time dependent material phenomena by Wetzel in Reference [6]. He accounted for cyclic creep/relaxation as being due to the influence of mean stress. The mean stress resulted in a bias that resulted in additional plastic increments with cycling. A tensile bias is seen in Figures 3, 4, and 5 for the mean stress. Not being time dependent would imply that cycle dependent phenomena are independent of rate effects.

Finally, the Neuber loading condition allows both creep and relaxation to take place simultaneously. The creep/relaxation may be either cycle or time dependent. Creep and relaxation would therefore be expected to occur for notches. The Neuber computer simulation, in Figure 5, demonstrates the simultaneous occurrence of both cyclic relaxation and creep that could be caused by either cyclic or time dependent material responses. A tensile bias is also evident for the mean stress in Figure 5 for this Neuber simulation.

Relaxation of stress in metals was attributed to these transients. The two types of relaxations are relaxation of mean stress and relaxation of residual stress. For the initial tensile mean stresses in Figures 4 and 5, relaxation of mean stress is clearly evident. For the Neuber computer simulation of Figure 5, the residual

stress is seen to relax. The residual stress is the notch stress at points in time when the net section stress is zero.

In the area of residual stress, Potter in Reference [7] reported no change, that is, relaxation of stress for shot peened uniform bars during elastic loading. The tests in Reference [7] were performed at room temperature under cyclic loading conditions. The tests were performed for elastic nominal loading so that the stresses and strains remained elastic. The tests were done in load control for $R = -1$ and $R = +0.5$ ratios. Also, the effect of a stress concentration was not included in the study. In Reference [7], for a uniform bar, the residual stress did change if specimens were loaded so that yielding occurred. When yielding was induced, the residual stress would tend to be completely removed.

Other examples of residual stress changes were reported by Morrow, Reference [2], including additional work due to other investigators. Reference [2] also used smooth specimens tested in strain control. Morrow reasoned that the stresses varied about the residual stress in a notched member for elastic stresses which meant, in this special case, that the residual stress and the mean stress were identical. Therefore, the stresses increased and/or decreased above and below the residual stress, so that the residual stress was the starting value for subsequent stress values during cycling. However, the stresses are known to vary about the mean stress during a strain control test. Therefore, an elastic strain control test would

simulate relaxation response, so that the variation of the mean stress is comparable to the variation of residual stress in a notched member when the stresses are elastic. Then, the strain control test, of a uniform bar, would qualitatively approximate the surface response of a notched member. The strain control test's mean stress would then approximate residual stress variation. Therefore, changes in residual stress may be studied by monitoring the changes in mean stress of smooth specimens during cycling if stress remains elastic so that residual stress and mean stress are identical. However, if subsequent plastic stresses are introduced at a notch, the memory of the previous state of residual stress is erased, and a new state of residual stress occurs. Also, for the more general case of elastic-plastic stresses at the notch, the residual stress and the mean stress are not coincident. In this case, relaxation of residual stress or relaxation of mean stress to zero will, in general, produce different shifts in stress-strain space for hysteresis loops during cycling, so that these phenomena are no longer the same. Morrow, in this case, used relaxation of mean stress while Potter supported relaxation of the residual stress during cycling. Therefore, fatigue models based on relaxation of residual stress or relaxation of mean stress to zero should produce fundamentally differing results.

Cyclic mean stress variation has been seen in strain control tests of metals, and this effect was reported in Reference [3]. Also, in Reference [4], cyclic mean stress variation has been seen in Neuber

control tests of metals. Therefore, during cyclic loading, Neuber control testing suggested a mean stress variation for notched members.

Cyclic mean strain variation has been observed in stress control tests, with an example of such variations being given in References [3, 8]. Cyclic mean strain variation is indicative of cyclic creep, also referred to as strain ratcheting, but could also be due to time dependent behavior. Cyclic mean strain variation occurred in the presence of a mean stress, a tensile mean stress causing a tensile variation in mean strain, whereas, a compressive mean stress causes a compressive variation. Also, Neuber control allowed variation of the mean strain during cycling: examples for Neuber control may be found in Reference [4]. Again, during cyclic loading, Neuber control testing suggested the occurrence of strain ratcheting for notched members.

Further examples of transient material behaviors at room temperature are given by several investigators. Stainless steel has been shown to exhibit cyclic and time dependent creep and relaxation: this material's behavior was documented in Reference [9]. In Reference [5], a Cr-Mo-V rotor steel displayed cyclic creep at ambient temperature. Finally, Kurath reported in Reference [1] steady state creep in a titanium alloy at room temperature. The behavior in Kurath [1] was that of primary creep where the creep rate eventually decreased to zero. For creep to occur, stresses near or above the elastic limit where necessary for the titanium alloy of Reference [1].

To summarize, cycle and time dependent responses exist in metals at room temperature. Cycle and time dependent behavior has been viewed by various workers as occurring simultaneously or independently. When cycle dependent response is time independent, its response will be independent of rate. Stress control allowed creep responses while strain control allowed relaxation to occur. Indications from Neuber control testing suggested that a notched member allowed both creep and relaxation to take place. Finally, it has been proposed that mean stress variation may be used to infer the variation of residual stress in a notched member.

2.2 NUMERICAL METHODS

2.2.1 Stress and Strain Response

Stress and strain response has been modeled using unified elastic-viscoplastic models. The unified elastic-viscoplastic models were time-temperature dependent: examples of these models are listed in References [5, 10]. These models were mostly given in the differential form. Reference [10] described these models as being stiff: the stiffness resulted due to modeling plastic behavior. The stiffness was observed when small changes in one variable and large changes in another variable occurred simultaneously. Stiff differential equations have resulted in the study of numerical procedures to solve them.

The viscoplastic models included time dependency with creep and plasticity interactions to describe the mechanical response of metals: models with these characteristics were described in Reference [10]. Mechanical response may be observed as creep/relaxation and plastic flow. Creep/relaxation and plastic flow were modeled as interdependent quantities. Each quantity depended on the current values of the others. The interdependent models were called unified viscoplasticity models. Due to the time dependency, transients and rate effects were seen. Time independent cyclic hardening or softening transients were also included. Also, the elastic-viscoplastic models included nontransient steady state cycling.

Given the viscoplasticity model, cyclic creep/relaxation may be displayed. Cyclic creep occurred during stress control. In the presence of a mean stress, the mean strain varied. Cyclic relaxation took place under strain control. In the presence of an initial mean stress. the mean stress relaxed to zero. Both cyclic creep and cyclic relaxation was allowed under Neuber control. The mean stress and mean strain were allowed to change. Therefore, the unified elastic-viscoplastic models formed the constitutive laws for the material.

Listed in Reference [10], the unified viscoplastic models had a number of similarities. Unified models may be developed with or without yield criterion. (Both forms are used.) The unified elastic-viscoplastic models assumed plastic incompressibility, and the models included internal state variables. The state variables depended on

the prior loading history and were combined with physical variables to form the model. The state and physical variables completely defined the material state at any time, and, once known, any other state may be determined in the future.

The first state variable was the isotropic hardening variable which was also called the drag stress or yield stress. The isotropic hardening variable was usually a scaler. This variable characterized cyclic hardening/softening. The other state variable was known as the kinematic or directional hardening variable. The kinematic hardening variable accounted for the Bauschinger effect that is also referred to as directional hardening. The kinematic hardening variable also modeled low plastic strains for theories without a yield surface. The kinematic hardening variable may be a tensor or a scaler. If it was a scaler, the scaler was the function of a tensor. Other terms for the kinematic hardening variable were the equilibrium, back, or rest stress.

The unified viscoplastic constitutive equations must be integrated. Integration of viscoplastic constitutive equations have been studied, and typical results were given in Reference [10]. To integrate the differential form of the viscoplastic constitutive equations, standard procedures have been proposed.

Complex higher order integration schemes were disliked since simpler ones could also produce satisfactory results and were as or

more computationally efficient. High order integration schemes require large amounts of computation for a single time step, and simpler integration schemes, such as Euler integration, are preferred; this being a simpler numerical procedure when compared to methods such as Runge-Kutta. The reason simpler methods were preferred was that, although the errors were higher, less computation for a time step was needed. Therefore, smaller time steps could be taken and still be computationally efficient and achieve a satisfactory reduction in error.

The stability of integration for the differential form was of concern; integration stability was discussed in Reference [10]. Because the viscoplastic formulations are computationally stiff, convergence may not take place for large time steps; it may occur giving erroneous results. Euler integration was found to be unconditionally stable if the strain increment was at most 0.0001: this restriction was only necessary during plastic increments. Therefore, adaptive time control methods were conceived to make integration more effective. Adaptive time control allowed larger increments during elastic loading and smaller increments during plasticity to be utilized. Allowing large time steps during elastic strains and needing smaller time steps during plasticity were a common feature of all the viscoplasticity models in the differential form. This difficulty was present independent of the type of numerical integration scheme chosen. The need for changing the step size was

characteristic of the differential equations being computationally stiff (which means, at a single instant in time, they are numerically ill conditioned since different variables produce both small and large values) during plasticity.

In view of the numerical difficulties of integrating the differential forms, Walker, in Reference [5], has proposed a recursive integral method. The numerical integration method given by Walker was stable for his recursive integral formulation for any size time step. Accuracy, however, may be undesirably influenced by step size. For large and plastic strains, convergence was never a problem no matter how large a time step was taken. The only consideration was the control of errors. Convergence to the correct value in the elastic range forced a smaller time step. If plastic stresses and strains were to be determined, satisfactory results were possible in one time step. Walker's method seems to be suited to a constant number of intervals in which there is a lower limit on the interval length. If the loading is completely elastic and the number of intervals is constant, the stress and strain intervals are shorter. However, if the loading interval is large enough to produce plasticity, the stress and strain internals are larger. The result is that the intervals are refined for elastic loading and coarser for plastic loading. This behavior was opposite to the requirements for the differential forms. The differential formulations dictated a maximum strain interval for plasticity which was usually small. The elastic intervals were

allowed to be much larger for the differential forms. The integral form seems to be more efficient in step size utilization, if one considers that in a notched member, the amount of time spent calculating plastic solutions may be much greater than that spent for elastic values.

Walker's integral equations had to be converted to a numerical algorithm for computer evaluation. The original procedure was first to divide the interval of integration into subintervals. Next, over each subinterval, the integrand was approximated as constant during a subinterval, permitting the integral's value to be evaluated approximately. Finally, over the complete interval of integration, the values of the subintervals may be summed to form the value of the integral. Therefore, the integration was in piece-wise constant steps. Since the variables in the constitutive equations were interdependent for the unified models, a nonlinear system of equations resulted for the numerical algorithm. The system of nonlinear equations was therefore implicit. Again, the variables in the nonlinear equations were interdependent; therefore, Walker called the integration scheme the recursive integration method.

However, Walker refined his original recursive integration method: these improvements were reported in References [11-13]. The differential equations were again recast as integral equations. The integral equations were then approximated for numerical analysis by an asymptotic expansion. The resulting implicit equations were then

formulated to be solved iteratively by the Newton-Raphson method. In Reference [12], this method was applied to a general first order differential equation: the details are given in the Appendix of Reference [12]. Reference [11] illustrates the improved procedures use with several viscoplastic models. Also, in Reference [11], numerical results were compared with a self-adaptive Euler method.

To obtain the constitutive laws in integral form, the differential equations must be rewritten. A method for rewriting a first order differential equation as an integral equation was presented in Reference [14]; in addition, the integration involved the use of an integrating factor. The integration factor method was also used by Walker. In the Appendix of Reference [12], Walker's procedure for producing the integral equation from a first order differential equation was outlined. but there must be a first order differential equation. The procedure for recasting the constitutive equations into first order differential equations was outlined in Reference [11].

Neuber's rule allows the determination of the local cyclic stress and strain. References [15, 16] describe Neuber's rule. Any constitutive relationship may be used for the stress-strain response: this may or may not include mechanical cycle and time dependent transient behavior. Combining the constitutive equations with Neuber's rule allowed the effects of a stress concentration to be included. Comparison of Neuber's rule with analytical studies were given in References [17-20].

By applying Neuber's rule with the appropriate constitutive equations, local stresses and strains are determined and sequence effects (the sequential order in which cycles occur) are modeled in a notched member. The constitutive relations may include cyclic hardening and nonlinear stress-strain behavior. Also, if cycle and time dependency occurred in the constitutive relations, relaxation along with rate effects were included.

For a notched member, once the local stress-strain behavior was determined, a fatigue analysis was possible. Examples of using Neuber's rule for fatigue analysis without transient effects in the material model are discussed in References [17, 18, 20, 21]. Previous numerical work considering fatigue and a time dependent material included that of Kurath in Reference [1] in which Neuber's rule was used. Kurath used a load sequence of ramping followed by a hold period; so, the major amount of time dependent behavior was during the hold period. Ramping was modeled by the cycle and time independent cyclic stress-strain curve. Time dependence was included during the hold period during which a creep equation was used. Kurath's model included sequence and mean stress effects. The mean stress value changed the fatigue life. A tensile mean stress would shorten the fatigue life, whereas, a compressive mean stress would lengthen the fatigue life. Also, relaxation was present during hold periods, and cycle and time dependent mechanical behavior was excluded during ramping.

The modeling of cyclic ratcheting and cyclic relaxation does not require a time dependent material model; these behaviors may be produced by time independent plasticity models. For time independent models, the ratcheting effects were inherently due to the occurrence of plasticity. The ratcheting occurred in the presence of a mean stress. Time independent plasticity models with ratcheting were provided in References [22–26]. With respect to ratcheting, the viscoplastic models were found to over predict the effect. For the viscoplastic models during cycling, Reference [25] pointed out the need to improve the prediction of ratcheting and closure of minor cycles. Closure is due to mechanical memory which manifests itself on loading, unloading, and reloading as a partial retracing of the unloading path followed by a continuation of the original stress-strain loading path. Closure is poor for viscoplastic models because on reloading while retracing the unloading path they deviate earlier than observed experimentally before continuing the original loading path.

2.2.2 Fatigue Life Prediction

Fatigue life prediction was made from the local stress-strain behavior. Even if the local stress-strain behavior contained transients, fatigue life may be computed based on traditional methods. The traditional methods were cycle counting and strain-life curves. Cycle counting was done to identify hysteresis loops. For each

hysteresis loop, the strain amplitude and mean stress were determined. For each loop, the strain amplitude and strain-life curves were combined to compute damage. Mean stress effects on damage were included using mean stress parameters such that a tensile mean stress shortens fatigue life, and a compressive mean stress lengthens fatigue life. Finally, for the collection of hysteresis loops, damage summations were performed for all loops counted.

Rainflow cycle counting is described in References [17, 21, 27-29]. Fatigue and cumulative damage procedures are outlined in References [17, 18, 21], and these procedures included strain-life curves, mean stress, and Miner's rule.

Before this study, typical damage calculations were previously based on stable hysteresis loops as outlined in References [17, 18, 30]. Stable hysteresis loops do not contain transients. Therefore, for stable hysteresis loops, there was a unique stress amplitude for each strain amplitude. Due to transient behavior a unique stress amplitude does not exist for each strain amplitude. Experimental and numerical examples and also a discussion of this behavior was given in References [3-5, 10-12, 23, 24] for cyclic loading. Also, mean stress is not repetitive for the transient case for Rainflow counted cycles as compared to the appropriate nontransient case when constant cycling is occurring.

2.3 PREVIOUS COMPUTER PROGRAMS

Computer based applications were developed by Martin and Wetzel for local strain fatigue analysis. Martin's work is detailed in Reference [31], and Wetzel's contribution is documented in Reference [6]. The computer modeling included loop shape, cyclic hardening, and relaxation of mean stress. The modeling was done by modeling the stable cyclic stress-strain curve. The stable cyclic stress-strain curve was modeled as a series of linear segments. To achieve the desired hardening and relaxation, the slopes of the cyclic stress-strain curve segments were modified. The above method was useful in predicting time independent cyclic effects.

On the other hand, the local stress-strain approach was incorporated in a computer program by Brose as described in Reference [3]. He based the stress-strain analysis completely on the stable cyclic stress-strain curve. Again, the cyclic stress-strain curve was divided into a series of linear segments. But, the slopes of the segments did not vary. The cyclic stress-strain curve segments were used repeatedly in a certain order to generate hysteresis loops. Following the load history, loops were generated while satisfying Neuber's rule. The model included sequence and mean stress effects where the mean stress caused a decrease or increase in fatigue life. Cycle/time dependent behavior was not included. Brose's computer model did not account for transient variations in mean stress or strain amplitude. In summary, Brose's computer model included

variable amplitude histories, the stable cyclic stress-strain curve, Neuber's rule, cycle counting, mean stress effects, sequence effects, Miner's rule, and strain-life curves. The variable amplitude load history accounted for the variation in the load and gave its sequential order. Miner's rule was used for fatigue damage summation.

More recent computer work was provided by Dowling and Khosrovaneh in References [30, 32]. The work of Dowling and Khosrovaneh was based on fundamental developments due to Conley and Socie in References [33, 34], respectively. These methods placed upper and lower bounds on the fatigue life. This method was based on observing that all minor hysteresis loops must be contained within a major limiting loop. The positioning of the minor loops inside of a major limiting loop was visible using stress-strain plots. Therefore, the major limiting hysteresis loop placed a bound on the mean stress values of the minor loops. Again, only the stable cycle stress-strain response was studied.

2.4 X-RAY STRESS MEASUREMENT

The determination of stress from x-ray measurements involves the use of the d-spacing versus $sin^2\psi$ method. The d-spacing versus $sin^2\psi$ method is described in References [35–39] and was applicable for the case of biaxial stress, and it was used to determine the normal stress in a plane which was the surface of the specimen. The d-spacing versus $sin^2\psi$ relationship is given in the following equation:

$$d_{\psi} = d_n \frac{(1+\nu)}{E} \sigma_{x-ray} \times sin^2\psi + d_n \qquad (1)$$

where:

d_{ψ} – The lattice spacing of a crystal plane whose normal forms an angle psi with the normal to the surface, and the dependent variable for the linear least squares fit.

d_n – The lattice spacing of a crystal plane which is parallel to the surface whose value is taken to be the intercept when ψ is zero from the linear least squares fit.

$\frac{(1+\nu)}{E}$ – The x-ray elastic constant.

σ_{x-ray} – The mechanical stress determined by x-ray diffraction after a linear least squares fit is performed.

$sin^2\psi$ – The dependent variable for the linear least squares fit.

ψ – The angle formed by the normal of any crystal lattice plane and the normal to the surface.

The stress was evaluated in a predetermined direction. The d-spacing versus $sin^2\psi$ relationship forms a straight line whose slope is proportional to the stress. The constant of proportionality contained the x-ray elastic constant $\frac{(1+\nu)}{E}$. The d-spacing is the distance between a particular set of crystal lattice planes which is symbolized as d_{ψ} and d_n in Equation 1.

Crystal planes diffract monochromatic x-radiation in a prescribed manner based on their spacing. By measuring the spacing of crystal lattice planes, the stress could be determined. The spacing of crystal lattice planes is directly measured by x-ray diffraction, and the diffraction is given by Bragg's law as described in References [35, 36, 38-40].

The crystal lattice spacing (the d-spacing) changed if the stress field changed. Also, for a single stress field, the d-spacing changes with ψ: ψ is the angle of tilt of the diffracting planes with respect to the surface of the specimen. Therefore, on the diffracting planes at each ψ tilt, for a single stress field, the stress exerted changed due to the change in ψ; it may be shown by static equilibrium that on the planes the stresses acting changed for various ψ tilts. Using Bragg's law, the d-spacing changes were observed as a change in the diffraction angle. Figure 6 illustrates the change in d-spacing and the change in diffraction angle, θ, resulting from a change in ψ tilt. The change in diffraction angle, θ, was recorded as a shift of the intensity peak which was measured by x-ray diffraction.

For the case of biaxial stress, the d-spacing can be shown to vary according to the d-spacing versus $sin^2\psi$ relationship of References [35-39]. The d-spacing is the dependent variable, while $sin^2\psi$ is the independent variable. The relationship is linear for materials with little or no texture as described in References [35-37, 40]. A minimum of two ψ tilts must be measured to define the line. For the

plotted line, the stress is proportional to the slope. The constant of proportionality is the x-ray elastic constant. The x-ray elastic constant must be determined experimentally. From the slope, the stress determined is in the plane of ψ tilt; that is, along the direction of ψ tilt. Also, the stress is parallel to the specimen surface. The process is as follows: making measurements at several ψ angles, determining the resulting diffraction angles, using Bragg's law to determine the d-spacing, plotting d versus $sin^2\psi$, determining the slope, and using the x-ray elastic constant to compute the stress. An actual plot of d-spacing versus $sin^2\psi$ is included in Figure 7.

X-ray stress measurements were made previously by other investigators. X-ray stress measurement of residual stress was done and documented in References [41, 42] which did not include changes in residual stress with cycling. Residual stress was also studied by Potter who studied residual stress changes in uniform shot-peened bars. The uniform shot-peened bars were subjected to cyclic loading. Potter's results were given in Reference [7]. For shot-peening induced residual stress, changes were also reported by Vohringer in Reference [43]. Vohringer studied residual stress changes due to both annealing and mechanical loading.

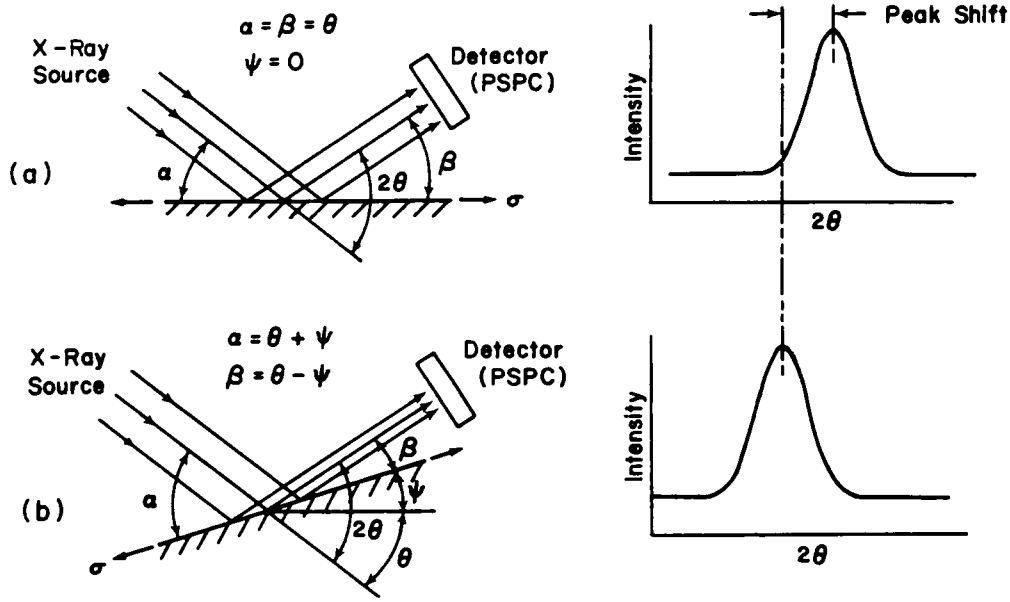


Figure 6. Residual stress measurement by x-ray diffraction.

Sample Description :
A2R04A/EP/LOADED Specimen# away from detector FSPC 1578V
measurement# 1

Stress Spectra File Specifications 000625.SPC

Residual Stress	(ksi)	51.40	(mpa)	354.37
Statistical Error (+/-)	(ksi)	1.82	(mpa)	12.54

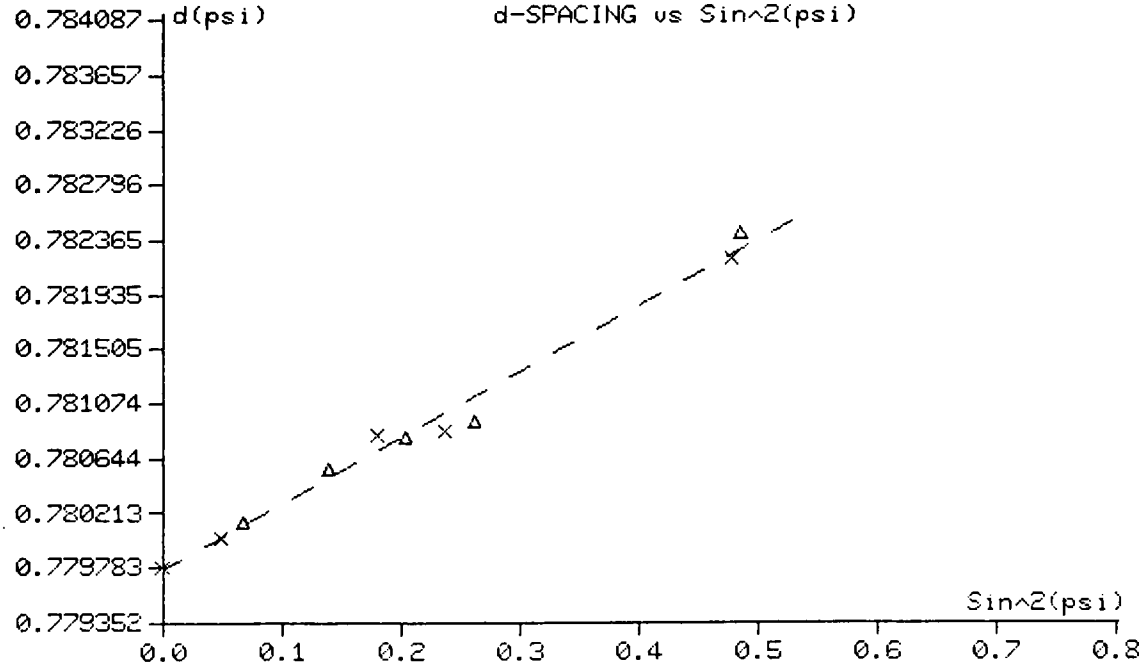


Figure 7. Residual stress analysis report graph.

Residual stress relaxation in uniform shot-peened specimens was observed for gross yielding, as well as when local yielding occurred at the surface. Considering the initial value of the residual stress at the surface, the net section stress needed to yield the surface could be predicted so that the surface seems to remember prior plastic history and its influence on future plasticity (memory). Memory may be attributed to phenomena such as the Bauschinger effect for kinematic hardening.

Unexpected surface behavior has been previously observed in Reference [44] for metals by x-ray stress analysis. The unexpected surface behavior can not be accounted for by mechanics which were based on observed bulk behavior. The unexpected surface behavior was called a surface effect.

For a carbon steel at room temperature during strain cycling, a uniform specimen displayed surface effect residual stresses in Reference [44]. The surface stress was measured by x-ray diffraction. After cycling, the residual stress was higher near the surface than in the interior with the residual stress tapering off to a depth of 1000 microns or 0.00394 inches. Since residual stresses are induced by plastic strain mismatch, the surface plastic strains were different than the bulk to the 1000 micron depth. Reference [44] reasoned that a lower yield strength existed in the vicinity of the surface. Also, a different strain hardening coefficient at the surface was reasoned as able to cause a surface residual stress in a uniform specimen.

3.0 MATERIALS

The materials studied were wrought titanium and aluminum alloys. The stock material was in plate form. The aluminum was in a 3/4 inch thick plate, and the titanium was in a 3/8 inch thick plate. The plates of each alloy were picked which had a microstructure suitable for x-ray stress analysis where the grain structure was fine enough and, therefore, a large enough number of grains were irradiated by the x-ray beam. Also, a large number of irradiated grains are needed if texture (preferred orientation) is not to be present although it is not sufficient to prevent texture. The aluminum was designated as 7475-T651 Al. 7475-T651 Al alloy shares a similar composition with 7075 Al alloy with the 7475-T651 Al alloy having a tighter tolerance on the composition. The composition is the quantity of elemental metals, alloying elements, and chemical impurities allowed in an alloy. Therefore, 7475-T651 Al should be very similar to 7075-T6 Al in mechanical properties and behavior. The titanium was designated as Ti-6Al-4V. The 7475-T651 Al and Ti-6Al-4V plates were used to make all specimens. The selected alloys' mechanical properties are listed in Table 1.

TABLE 1. Materials Identification and Properties

	Ti-6Al-4V	7475-T651 Al
Form	3/8 in. plate	3/4 in. plate
Condition	Mill annealed 1450°F	Solution treated and aged
Identification	Ingot No: 990211-02-00	Serial No: 511348-1
Source	RMI, Niles, OH	Alcoa Labs, Alcoa Ctr., PA
Ultimate, ksi	142	93.7
Yield, 0.2%, ksi	133	82.6
Elongation, %	14	6

4.0 MODELING

4.1 INTRODUCTION

The local stress-strain response was modeled for notched members. The notched members were to be made from metals which were to be subjected to uniaxial loads. For the given loads, the notch's mechanical behavior was modeled using the local strain approach. The local strain approach was used because cracks were assumed to initiate at notches which are also stress concentrations. The initiation of cracks at stress concentrations was discussed in Reference [18]. Therefore, notches were considered critical areas. The notch's mechanical behavior was to be modeled for cycling. To model the notch's cyclic mechanical behavior, the model required including the material behavior and the effect of notch geometry.

The following material details were included in the model. The material was to be at room temperature as assumed by Landgraf in Reference [45]. The simulation of the cyclic nonlinear stress-strain was modeled using Walker's unified viscoplasticity model. The occurrence of cycle dependent hardening or softening was included. Also, the generation of time dependent creep/relaxation was

represented. Therefore, the modeling described both transient and steady state mechanical response. Due to the choice of Walker's constitutive model, cycle dependent creep/relaxation was modeled. The cycle dependent creep/relaxation included both time and cycle dependent effects as discussed in References [5, 22–26]. Walker's model was taken from Reference [5].

The influence of the notch geometry was included by the incorporation of Neuber's rule in the analysis. Previous examples of such usage of Neuber's rule was given in References [17, 18, 21]. The use of Neuber's rule was limited to the uniaxial load case. Neuber's rule accounted for the stress-strain concentration effects of the notch. The notch's expected local stress-strain behavior was then determined for any load spectrum of interest. The most general case of variable amplitude spectrum loads was included. The use of Neuber's rule avoided the use of two-dimensional finite element analysis.

In the present study, net section stresses and strains were nominally elastic. The nominally elastic stresses and strains condition avoided needing time dependent behavior modeling for the net section. Limiting the net section to the nominally elastic case was justified since loads high enough to cause net section yielding occur infrequently in structural components. Therefore, the usage of Neuber's rule was greatly simplified.

4.2 VISCOPLASTIC CONSTITUTIVE LAW

The constitutive equations in integral form are given in Table 2 for the uniaxial case. Table 3 gives the names of the various variables and constants. The model has 13 material constants. The magnitude of $N_1 + N_2$ controlled the rapidity of approach of the equilibrium stress, Ω, to its saturated value which was discussed in Reference [5]. A large value of $N_1 + N_2$ caused rapid growth of Ω resulting in elastic, perfectly plastic response. Intermediate values $N_1 + N_2$ resulted in nonlinear hardening. A small value $N_1 + N_2$ resulted in the slow growth of Ω. A small value of N_2 results in a viscoelastic response. The model does not use a yield surface.

The equations in Table 2 must be rewritten in a form suitable for numerical evaluation. Several approximations must be performed during this revision. The methods used in obtaining the rewritten equations were given in Reference [5], and the detail of the derivations of the equations used is given in Appendix A. An illustrative example of converting a linear first order differential equation follows. A linear first order differential equation has the form:

$$\dot{x} + \dot{P}x = \dot{Q}$$

where x, P, and Q are all functions of time only. An integrating factor can be defined as:

$$\mu = e^{\int \dot{P} dt} = e^{P(t)}$$

so that

$$\frac{d}{dt}(\mu x) = \mu \dot{Q}$$

with solution

$$\mu x = \int \mu \dot{Q} dt + c$$

rearrangement gives

$$x(T) = \frac{1}{\mu(T)} \int_0^T \mu(t) \dot{Q} dt$$

$$x(T) = \frac{1}{e^{P(T)}} \int_0^T e^{P(t)} \dot{Q} dt$$

$$x(T) = \int_0^T e^{-P(t)} e^{P(t)} \dot{Q} dt$$

$$x(T) = \int_0^T e^{-\{P(t) - P(t)\}} \dot{Q} dt$$

The final equation above is in the form of Walker's integral equations which are given in Table 2.

TABLE 2. Walker's Unified Viscoplasticity Theory.

Recursive integration method in Reference [5].

$$\sigma(t) = \Omega(t) + \int_0^t e^{-[q(t)-q(\zeta)]}\left(E\frac{\partial \epsilon}{\partial \zeta} - \frac{\partial \Omega}{\partial \zeta}\right) d\zeta \qquad \text{(T2.1)}$$

$$c(t) = \int_0^t \left(\frac{\partial \epsilon}{\partial \zeta} - \frac{1}{E}\frac{\partial \sigma}{\partial \zeta}\right) d\zeta \qquad \text{(T2.2)}$$

$$\Omega(t) = \overset{8}{\Omega} + N_1 c(t) + N_2 \int_0^t e^{-[g(t)-g(\zeta)]}\frac{\partial c}{\partial \zeta}\, d\zeta \qquad \text{(T2.3)}$$

$$k(t) = K_1 - K_2 e^{-N_7 r(t)} \qquad \text{(T2.4)}$$

$$q(t) = \int_0^t \frac{E}{k(\zeta)}\left(\frac{\partial r}{\partial \zeta}\right)^{1-\frac{1}{N}} d\zeta \qquad \text{(T2.5)}$$

$$g(t) = \int_0^t \left\{\left(N_3 + N_4 e^{-N_5 r(\zeta)}\right)\frac{\partial r}{\partial \zeta} + N_6 \left|\Omega(\zeta)\right|^M - 1\right\} d\zeta \qquad \text{(T2.6)}$$

$$r(t) = \int_0^t \left|\frac{\partial c}{\partial \zeta}\right| d\zeta \qquad \text{(T2.7)}$$

TABLE 3. Viscoplasticity Variables and Constants.

σ -Mechanical stress	Physical variables
ϵ -Total strain	
c -Plastic strain	
r -Cumulative plastic strain	
k -Isotropic hardening coefficient	State variables
Ω -Kinematic hardening coefficient	
q -Variable in $e^{q(t)}$ integrating factor for mechanical stress	
g -Variable in $e^{g(t)}$ integrating factor for equilibrium stress	
t -Time	
ζ -Time: variable of integration	
$E, N, M, N_1, \cdots, N_7, K_1, K_2, \overset{\circ}{\Omega}$ -13 material constants	

4.3 DETERMINING VISCOPLASTIC MODEL CONSTANTS

The evaluation of the constants for Walker's model involved using data from strain control hysteresis loops. The test data were gathered at different strain rates. One method to determine the constants, indicated by References [46, 47], involved the use of computer simulation combined with nonlinear optimization or minimization techniques. The hysteresis loops were numerically generated with given initial constants and compared to the actual test data. One method of doing the comparison was to form the square of the difference between the generated curve and the experimental curve according to the following equation:

$$f=\sum_{i=1}^{n}\{\sigma_{data}(t_i)-\sigma_{cal}(t_i)\}^2 \tag{2}$$

This summation may be extended to a numerical integration as in the following equation:

$$f=\int_0^T\{\sigma_{data}(t)-\sigma_{cal}(t)\}^2dt \tag{3}$$

The above equations are both the method of least squares. The summation by numerical integration would smooth errors and noise in the data as an expected benefit. If enough data points per hysteresis loop were calculated, the arithmetic sum should be satisfactory even in the presence of noise. The constants were adjusted to minimize the

magnitude of the sum. Automated search procedures were available as part of standard minimization and optimization routines. One source of such routines was the IMSL FORTRAN library.

For the Ti-6Al-4V titanium and the 7475-T651 aluminum, it was not possible to determine constants for the two alloys due to efforts by Walker [47] using optimization procedures in a computer code developed by him. We were unable to compute constants because the hysteresis loops do not vary significantly with strain rate for the chosen materials. Therefore, little or no time dependence occurred during fully reversed cycling. The cycling conditions included uniaxial loading, strain control, fully reversed strain, constant strain amplitude, varying strain rates, and several specimens, one for each strain rate. The varying strain rates essentially did not result in observed changes in stress amplitude. The unchanging stress amplitude meant the tests displayed rate independent behavior, and the rate independent case was not included in the viscoplastic modeling. In general, the viscoplastic models and, in particular, Walker's model do not reduce to the rate independent case which was indicated by Walker in Reference [47]. When the viscoplastic models are used to simulate rate independent behavior, they become computationally stiff and difficult to solve numerically resulting in the optimization routine failing to converge to a solution. Also, performed static relaxation tests showed small amounts of time dependence for plastically deformed uniaxially loaded specimens where the static

relaxation was observed during a constant displacement hold period. For the static relaxation tests, the stress dropped with time as anticipated.

4.4 INCLUDING NEUBER'S RULE TO APPROXIMATE NOTCH EFFECTS

Neuber's rule allowed approximating the local notch stresses and strains given the nominal load history. The rule limited the analysis to the case of uniaxial loading, and it made the approximation possible because it gives the relationship between net section stress-strain and notch stress-strain responses. From the load history, the net section stresses and strains were given as input to Neuber's rule. Neuber's rule may be simplified if there is a one-to-one correspondence between the net section's stress and strain; for this one-to-one correspondence, only the net section stress is needed as input. For metals, the simplification is definitely valid for elastic nominal loading since the nominal stresses and strain follow a linear relationship which is true even when cyclic and time dependent material behavior is considered. Cycle and time dependent response does not occur for elastic loading in metals. Therefore, for elastic net section loading, the stress-strain values input to Neuber's rule did not need modification to include cycle and time dependent net section behavior.

The determination of the local uniaxial stress and strain history in the notched member utilized Walker's integral viscoplastic

constitutive equations combined with Neuber's rule. The previously given Figure 5 is a plot of the notch stress-strain response due to Walker's constitutive laws extended with Neuber's rule in this investigation. Walker's constitutive laws interpolated the stress-strain response between reversal points. This response included creep, relaxation, cyclic, rate, time dependent behaviors, and the notch's effect on these behaviors.

The typical application of Neuber's rule defined hyperbolae along the stress-strain path. The hyperbolae were given by the variation of the net section nominal stress with time from one reversal point to the next. Typically, the variation of hyperbolae were given only from reversal point to reversal point according to the following equation:

$$\Delta\sigma\Delta\epsilon = \frac{(k_t\Delta S)^2}{E} \tag{4}$$

For the equation above, intersection of the stress-strain response with a hyperbola located each reversal point in stress-strain space. The above form of Neuber's rule was correct for nontransient behavior. For nontransient behavior, the complete stress-strain path between reversals was not needed to locate each reversal point. Therefore, the above form of Neuber's rule only involved the ranges and amplitudes for stress and strain for cycling.

When transients of cycle and time were considered, the intervening stress-strain response must be determined since the loading rate and all transients must be considered. Therefore, Neuber's rule was written in a subincrement form that allowed the determination of the stress-strain response between reversal points. The subincrement Neuber's rule was

$$c = (\delta\sigma' + \delta\sigma)(\delta\epsilon' + \delta\epsilon) = \frac{\left(k_t(\delta S' + \delta S)\right)^2}{E} \tag{5}$$

according to the notation of Figure 8.

The quantities $\delta\sigma'$, and $\delta\epsilon'$ corresponded to the previously computed cumulative stress and strain increments for the previous cumulative load increment $\delta S'$. The quantities $\delta\sigma'$, $\delta\epsilon'$, and $\delta S'$ were measured cumulatively from the previous reversal point until the next reversal point was reached. After reaching the next reversal, it became the reference point for future measurements of $\delta\sigma'$, $\delta\epsilon'$, and $\delta S'$. The quantities $\delta\sigma$, and $\delta\epsilon$ correspond to the present stress and strain subincrements for the present load subincrement δS. The details of incorporating Equation 5 with Walker's model are given in Appendix A. The result of this incorporation is an iterative numerical routine for notch stress-strain response.

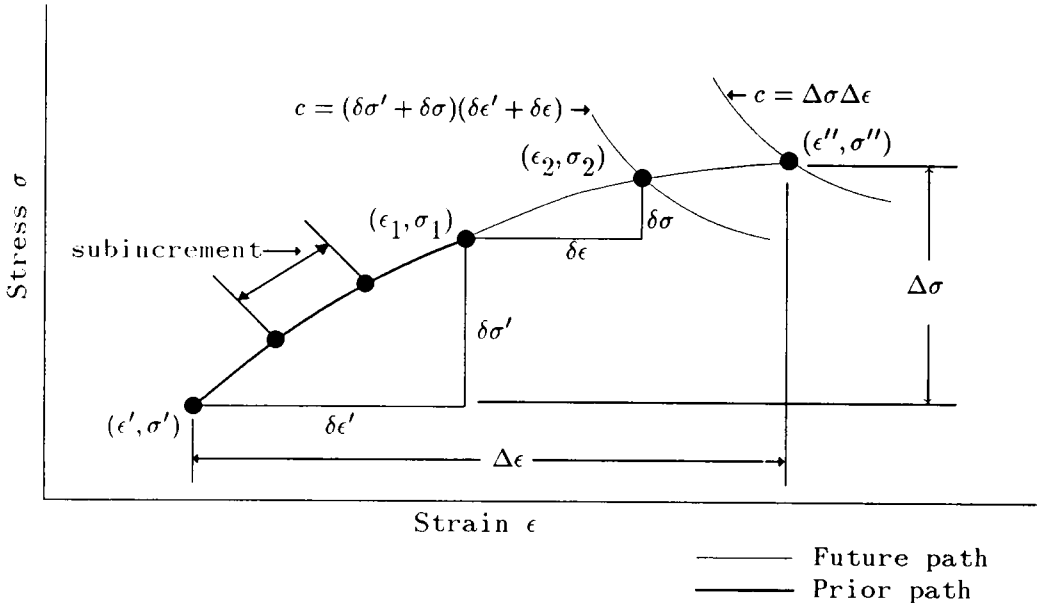


Figure 8. Subincrement Neuber's rule.

4.5 FATIGUE DAMAGE

It was proposed that if total strain-life data were obtained from constant amplitude strain control tests, a strain-based damage calculation could be made in the presence of transients. For strain control testing, the total strain does not vary during a test. So, for strain control testing, the presence of transients should not invalidate the resulting total strain-life data. For a total strain-based damage calculation, two values at least should be monitored. They are the strain amplitude and the mean stress. Other possible candidates were the stress amplitude or the strain energy: these candidates do not conform to the information available from typical total strain-life curves once transients are considered. For a constant amplitude strain control test, when transients are present, stress amplitude and strain energy varies during a test. The variation may be due to hardening or varying load rates. Therefore, using prevalent total strain-life data, a damage formulation may be based on total strain amplitude and mean stress which would be consistent. If this damage formulation is found to be satisfactory, elaborate testing may be avoided to account for hardening and loading rate effects not determined by conventional total strain-life data.

However, when an actual local stress-strain history is determined numerically, hardening and rate affect the computed strain amplitudes and mean stresses which would then be included in any damage computations: this hardening and rate effect depends on the type of loading.

The damage was calculated from the predetermined local stresses and strains. A variable amplitude history was analyzed to compute the local stress-strain response. Rainflow cycle counting was then done on the resulting local stress-strain history. Therefore, the damage calculation was done by post processing the results of a stress-strain modeling program.

The determination and counting of a complete strain cycle, cycle counting, was done on the previously computed stress-strain history. Since local elastic-plastic behavior with transients would not form closed hysteresis loops, a choice must be made to apply cycle counting to the local stress history or the local strain history. Because of transients, the results of cycle counting the stress history or the strain history were not equivalent as is the case of the stable cyclic analysis. Since a significant amount of fatigue life data were available based on strain control testing, to be consistent with these data, cycle counting was done on strain, and the strain amplitude and mean stress determined as appropriate.

Figure 9 is a representative diagram of two possible cases of cycles determined by the rainflow cycle counting method. In this example, from the prior local stress-strain history analysis, only the values of the stress and strain at the reversal points were recorded. The use of reversal points values resulted in a considerable reduction in data when compared with the data needed to record the complete stress-strain path. Since cycle counting was done on strain, strain amplitude was easily determined. From Figure 9, the strain range and amplitude for the representative cycle are

$$\epsilon_a = \frac{\Delta\epsilon}{2} = \frac{|\epsilon_1 - \epsilon_2|}{2} \tag{6}$$

Note that point 3 was chosen in the history so that by definition the strain at points 1 and 3 were equal due to cycle counting of the strain history. The stress at points 1 and 3 in general will not be equal due to cycle and time dependent transients. However, the stress at point 3 was unknown since it is not a reversal point. So, the mean stress may be determined based on the values of the stress at points 1 and 2 as

$$\sigma_0 = \frac{\sigma_1 + \sigma_2}{2} \tag{7}$$

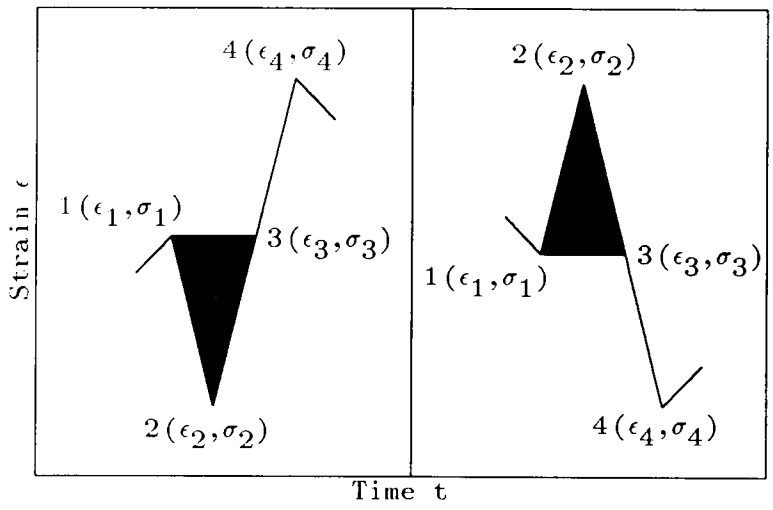


Figure 9. Representative strain counted cycles.

A more elaborate stress averaging may be done such as time integration of the stress over a cycle. Also, integration and averaging the stress with respect to the strain may be done. These more elaborate averages require extra computational effort and numerical data. No dramatic differences in these methods were expected. Also, the simpler averaging method should predict essential changes and values which are good for mean stress. For damage determination, more elaborate averaging may add complexity without improving predictive capabilities. The above average of Equation 7 was convenient and suitable when compared to the approximations already introduced. For the damage analysis, these approximations were the viscoplasticity model, Neuber's rule, rain-flow cycle counting, strain life curves, Miner's rule, and the modified Morrow mean stress parameter.

Once the strain amplitude and mean stress have been determined by cycle counting, damage was assigned to a cycle. The number of cycles to initiation was calculated from the two following equations:

$$\epsilon_a = \frac{\sigma'_f}{E}(2N^*)^b + \epsilon'_f(2N^*)^c \tag{8}$$

$$N = N^*\left(1 - \frac{\sigma_0}{\sigma'_f}\right)^{-\frac{1}{b}} \tag{9}$$

The value, N^*, is the life when the mean stress was zero. Equation 8 is a common form of the strain-life empirical relationship. The above

strain-life equation excluded the affect of mean stress. The value, N, was the life for a nonzero mean stress of σ_0. The mean stress relationship of Equation 9 is a common form of the modified Morrow's empirical relationship. The values σ'_f, E, b, ϵ'_f, and c are material constants. Using Equations 8 and 9, strain amplitude and mean stress were used to predict fatigue life.

Once damage for a single cycle was determined, damage was summed according to Miner's rule given in the following equation:

$$B\sum_{\substack{per.\\ block}}\left(\frac{n_i}{N_i}\right)=1 \tag{10}$$

The load history was assumed to repeat until failure. When the sum was unity, crack initiation to a finite size had occurred.

4.6 DETERMINING STRAIN-LIFE CONSTANTS

The strain-life constants for the two materials were obtained from companion specimens cycled to failure. The test conditions were fully reversed strain control, sinusoidal waveform, varying strain amplitudes and frequencies, and one strain amplitude and frequency per specimen. For the Ti-6Al-4V titanium alloy, the strain life constants reported by Khosrovaneh in Reference [48] are summarized here in Table 4. The strain-life curve for the Ti-6Al-4V titanium is given in Figure 10: this figure was adapted with changes from Reference [48]. The results of testing 7475-T651 Al are also included in Table 4. A

plot of the strain life curve for 7475-T651 aluminum is given in Figure 11.

The procedure to obtain the constants is briefly described. Starting with Equation 8, we identified the elastic and plastic strain terms as

$$\epsilon_a = \epsilon_{ae} + \epsilon_{ap}$$

$$\epsilon_{ae} = \frac{\sigma'_f}{E}(2N^*)^b \tag{11}$$

$$\epsilon_{ap} = \epsilon'_f(2N^*)^c \tag{12}$$

The above expressions for Equations 11 and 12 are both power law relationships. The constant E was the customary value for the modulus of elasticity. To obtain the other constants, the logarithms of the data were fitted by linear least squares regression. Linear regression was valid because, on a log-log graph, a plot of either Equation 11 or 12 would form a straight line. So, for the fits, the data for each specimen was tabulated as strain amplitude, stress amplitude, elastic strain amplitude, plastic strain amplitude, and fully reversed life to failure. All the amplitudes were computed close to the half-life of the specimen.

TABLE 4. Strain Life Constants.

Symbols, Units	Ti-6Al-4V[1]	7475-T651 AL
E, ksi	16500	10300
ϵ'_f	6.22	2.43
c	-1.01	-1.061
σ'_f, ksi	220.9	163.8
b	-0.0763	-0.1100

[1]Values from Reference [48], same plate of material used.

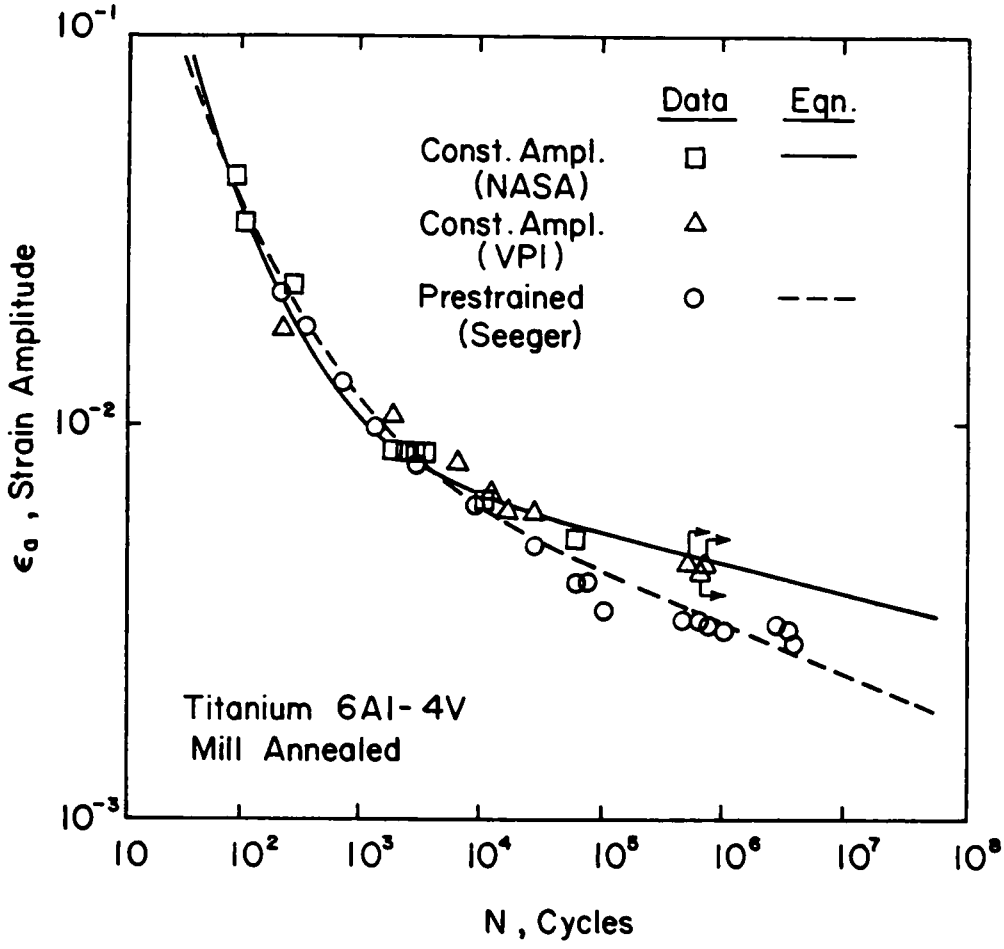


Figure 10. Strain Life Curve Ti-6Al-4V.

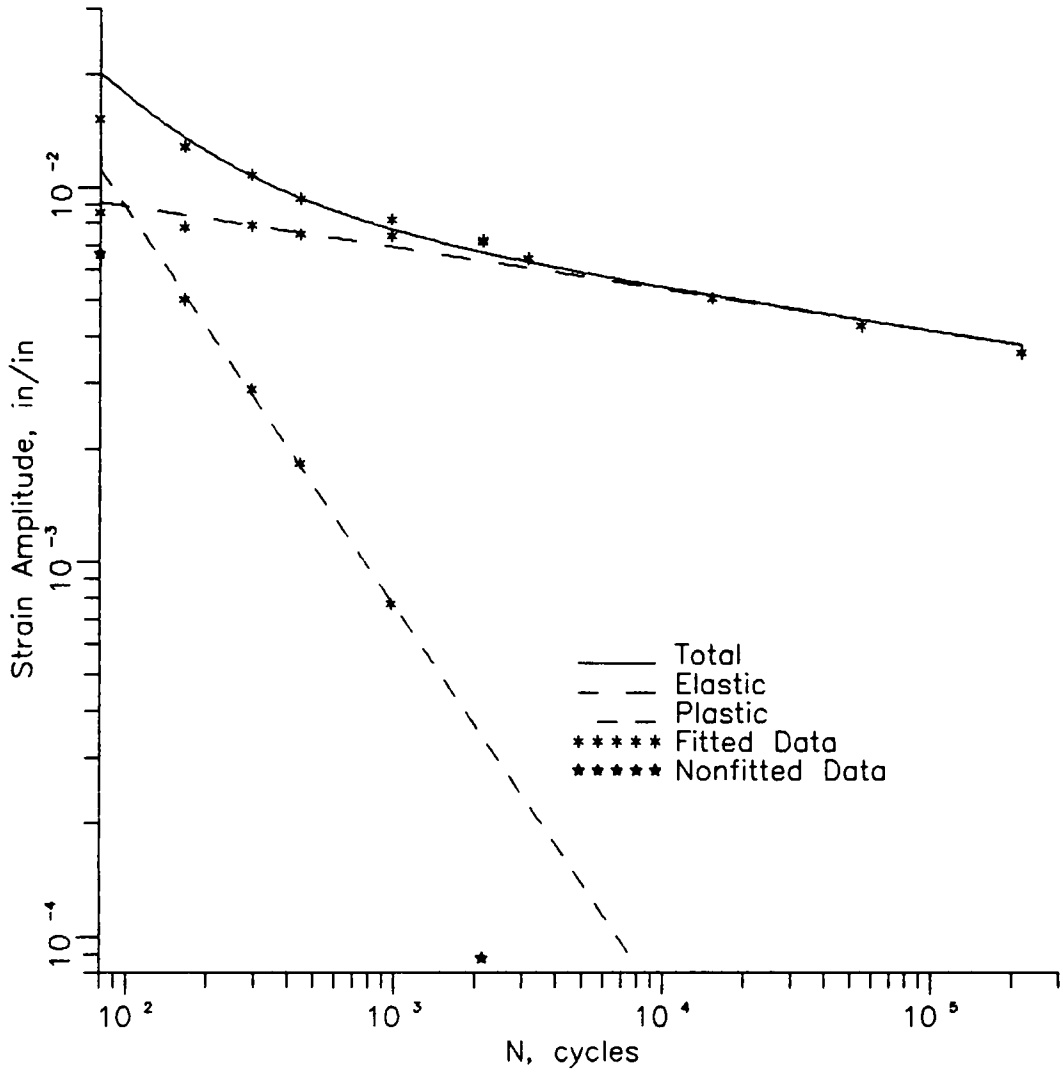


Figure 11. Strain Life Curve 7475-T651 Al.

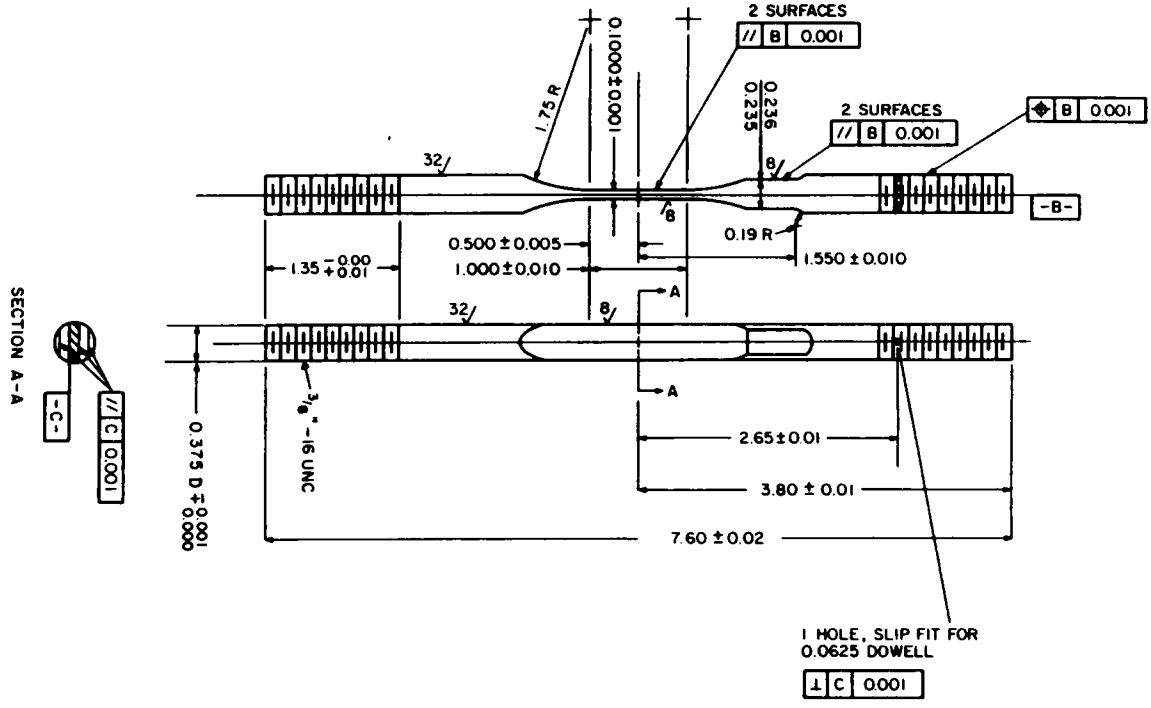
5.0 STATIC RELAXATION STUDIES

5.1 INTRODUCTION

Long term relaxation tests were performed on both the Ti-6Al-4V titanium and the 7475-T651 aluminum alloys. The specimens were loaded at a moderate rate in displacement control. Each specimen was loaded to a different displacement and held. Then, the drop in the load was monitored over time. The details are to be further described.

5.2 APPARATUS

For applying and maintaining the load, the apparatus included uniaxial static specimens, two loading grips, a $\frac{1}{2}$ inch gage length extensometer, two extensometer attachment bars, several relaxation frames, several hard steel dowels, and nuts which had a ball surface with mating socket washers. Also, a 20 kip MTS servo hydraulic testing machine was used for loading. For monitoring the load, strain gage instrumentation and x-ray stress measurements were used. Figure 12 shows the static specimen, whereas Figure 13 is a diagram of the relaxation frame.



Notes:

1. Stamp specimen number on one end before reducing center section.

2. In the final stages of machining of the reduced section, remove material in small amounts until 0.005 in. minimum of excess remains.

3. Remove the next 0.004 in. by grinding at a rate of no more than 0.002 in./pass.

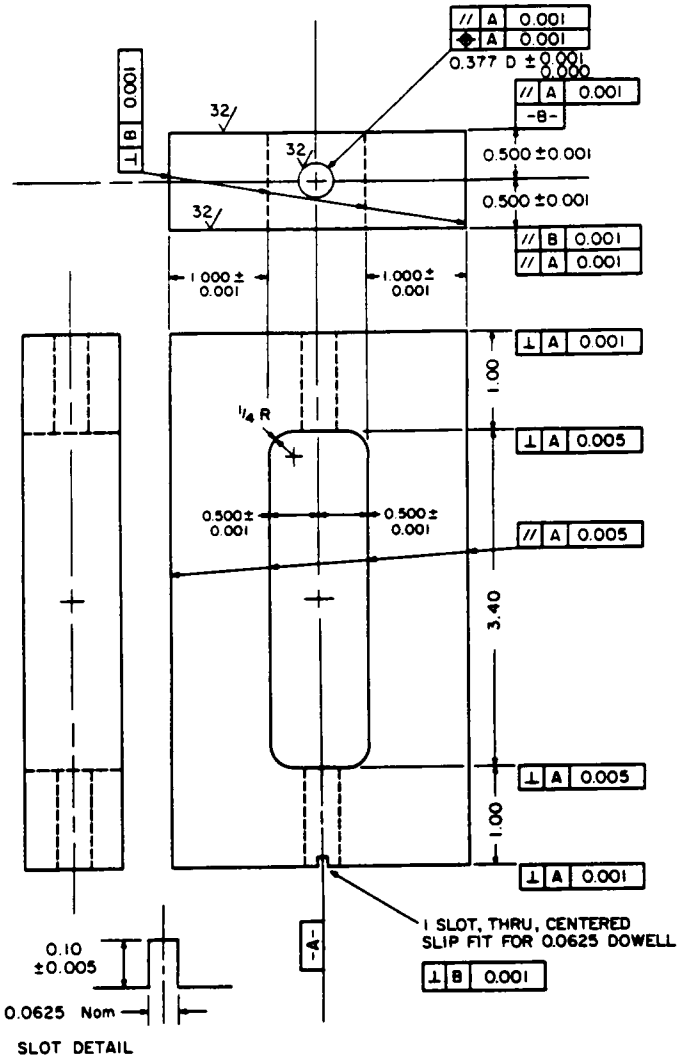
4. Remove the final 0.001 in. minimum by polishing longitudinally to a 8 μin. surface roughness. The polishing should remove material cleanly without smearing.

5. After polishing, all remaining marks should be longitudinal. No lateral machining should be seen at 20 x magnification.

6. Center section removed 0.002 min of surface by electropolishing.

7. Dimensions shown are inches.

Figure 12. Static Specimens.



Notes:

1. Material 1018 Plain carbon steel or equivalent. At least 60 ksi yield strength.

2. Dimensions shown are inches.

Figure 13. Relaxation Frame.

Figure 12 is an engineering drawing of the static specimen. The thinnest region was the test/gage section. The relaxation was assumed to be confined to the gage section. The gage section was machined with flats which provided an area for x-ray stress measurements. Next to the gage section, two opposing parallel flats where machined; in this area two strain gages were mounted. The strain gages formed a load cell for monitoring load. The flats for the two strain gages were not essential, but they made for simpler strain gage placement. The two strain gages were placed to exclude specimen bending from the load cell measurements. Also, for two strain gages, the sensitivity of the load cell was increased. For the specimen, two other features of importance were the threads and the hole. The threads were provided for nuts. Nuts will be used to maintain the load for an extended period. The hole was provided for a hardened steel dowel. The steel dowel mated to a machined slot in the relaxation frame. The steel dowel and slot aligned the specimen for x-ray stress measurements.

The combination of loading grips, attachment bars, extensometer, and MTS testing machine provided a means for controlling the displacement rate and level. The extensometer provided the control signal for the servo hydraulic testing machine. Then, an attachment bar was mounted to the loading grip to form a rigid unit, and the extensometer was mounted to the attachment bars. The axial separation of the grips, to which the specimen was threaded, was

monitored with the extensometer to control the loading of the specimen.

The relaxation frame, two nuts, and 2 washers provided a means of maintaining a fixed displacement for an extended period of time. Since a relaxation test was performed, the relaxation frame ideally should be perfectly rigid. The relaxation frame was sized to approximate this rigid condition. However, to a degree, there was evidence that possibly this condition was not met. Possible thermal expansion and contraction of the frame may have been evident in the results. The relaxation frame was made from a mild steel whose coefficient of thermal expansion was different for the 7475-T651 aluminum. Also, the mild steel and the Ti-6Al-4V titanium alloy have slightly differing coefficients of thermal expansion. The axis of the hole and the four sides of the frame were machined so that they are all parallel. The machining was done so the four sides could be used for alignment during x-ray stress measurement. When mated with a steel dowel mounted on the specimen, the slot of the frame further ensured that the specimen test section was properly positioned. In particular, the flat surfaces of the test section would be parallel to the front and back surfaces formed by the broad dimensions of the frame. The frame is shown in Figure 13. Nuts and washers were placed on each end of the specimen. Their mating surfaces formed a ball and socket arrangement.

Upon tightening the nuts and washers against the frame, the ball and socket surfaces reduced the possibility of loading a specimen with a bending moment.

The final arrangement of a specimen ready for loading consisted of a specimen placed in a frame with washers put on the specimen ends next to the frame. Next, nuts were threaded on each end. Finally, the loading grips were threaded on the ends of the specimen.

Strain gages, strain gage instrumentation, and x-ray stress measuring equipment were used. The strain gages were mounted next to the gage section to determine load in the machined region provided. The strain gages enabled the computation of the stress in the thin gage section. On each specimen, two strain gages were placed with one strain gage on each of the two prepared parallel surfaces. The strain measuring axis was closely placed along the axis of the specimen. The strain gages were connected to a strain indicator in a $\frac{1}{2}$ bridge configuration with the bridge having 4 branches. The 2 strain gages were wired in series to form one branch, and a temperature compensating gage was placed in a second separate branch of the bridge. The common temperature compensation gage was mounted to a block of the same material as the specimen. This $\frac{1}{2}$ bridge arrangement provided for both temperature and bending load compensation. The strain indicator provided both digital display and an analog signal. The x-ray stress analyzer was used to make x-ray stress measurements

directly in the specimen gage section. For x-ray stress measurements, no special procedures were required besides proper alignment of the specimen. The x-ray stress alignment and testing procedures are given in Reference [35].

5.3 TESTING PROCEDURE AND CONDITIONS

The procedure for loading the specimens is outlined in the following discussion. The frame, specimen, dowel, washers, nuts, grips, and extension bars combination were placed in a 20 kip MTS servo hydraulic testing machine for loading. The clip gage was then mounted to the extension bars to monitor displacement. Next, the strain gages, strain indicator, plotter, and strip chart recorder were connected as appropriate. The MTS testing machine was then placed in strain control. The specimen was then ramped at a moderate rate of 0.005 inch per second to achieve the proper displacement. The displacement varied for each specimen from 0.0018 inch to 0.010 inch. Next, while still in the MTS machine, the loaded specimen was monitored for a period of time from at least one hour to one day; at this point the load drop became slight over a time scale of several minutes. At this time, the nuts were tightened to carry the load. The MTS testing machine was simultaneously unloaded. The specimen strain gages were used for coordination so that the specimen load remained constant.

The specimen was then removed from the testing machine. Finally, the load was monitored with both strain gages and x-ray stress analysis for several months.

The following test conditions were present. The testing was done at ambient room temperature. The ramp rate was chosen as a compromise between an ideal infinite ramp and preventing overshoot due to testing machine response characteristics. Finally, using strain control to monitor and control the extensometer gave a more direct indication of specimen displacement than using the testing machine's actuator stroke indication.

5.4 RESULTS

The testing procedure exhibited one major difficulty that will be discussed. Also, the relaxation test data were compiled into graphs for further study. Finally, slight oscillations in the relaxation data were observed.

Originally, the nuts were to be firmly tightened allowing the testing machine load to be removed. After tightening the nuts and removing the testing machine load, only a slight drop in the specimen load was expected. However, this procedure produced a greater drop off in load. Therefore, to maintain the specimen load, alternate nut tightening and testing machine unloading was done. As the nuts progressively carried more load, the specimen would tend to twist due

to frictional forces between the threads of the nuts and the specimen. Provisions to prevent twisting or another load transferring method such as wedges should be utilized. Otherwise, the test method was adequate. However, completely transferring the load to the nuts was not done, and the resulting drop in load was accepted. Overloading the specimen followed by tightening the nuts and removing the testing machine load was not done to try and obtain the original load level. Overloading would have required several trial repetitions of loading and tightening until complete removal of the testing machine load resulted with the specimen at the original load before tightening. Overloading would give a different load history from the desired ramp followed by an indefinite hold. Especially, overloading would cause changes in the amount of plastic deformation from its original value. The plastic deformation due to overloading could then alter the relaxation response of the specimen. However, tightening the nuts as much as possible and accepting a load drop would produce elastic unloading. The elastic unloading was considered more desirable since time dependent behavior in metal was shown to be dependent on the presence of plastic deformation. However, the amount of elastic deformation available to be transformed into time dependent plastic deformation was reduced. Therefore, by accepting a drop in load, the time dependent relaxation may be altered the least.

The relaxation data for the 7475-T651 aluminum are given in Figures 14 and 15. The relaxation data for the Ti-6Al-4V titanium are

given in Figures 16 and 17. The data are plotted on various scales where the short term time scales are used to show the initial transients at the beginning of the test. Also, scales in time that cover the complete test are used to display the entire test data. The logarithmic scale visually emphasizes equally the variation in time of the stress. The strain gage data at the beginning of the tests had to be combined with x-ray stress data at the end of the test. Adjustments had to be made for mismatch in the strain gage data and the x-ray stress data. The adjustment involved, for simultaneous strain gage and x-ray measurements, the addition of a constant value to the x-ray stress value to obtain the same stress as that determined by the specimen's strain gages. The constant value was added to all subsequent x-ray stress values. The addition of a constant was required due to net section stress differing from x-ray stress values after plastic deformation has occurred. However, on elastic unloading, the "change" in both stress values are equal. Only the magnitude of the the x-ray stress data is less. The difference in net section and x ray stress values after yield is discussed later. Adding a constant value of the stress to the x-ray stress data allowed the splicing of the two data sets together to form one relaxation curve for each specimen tested.

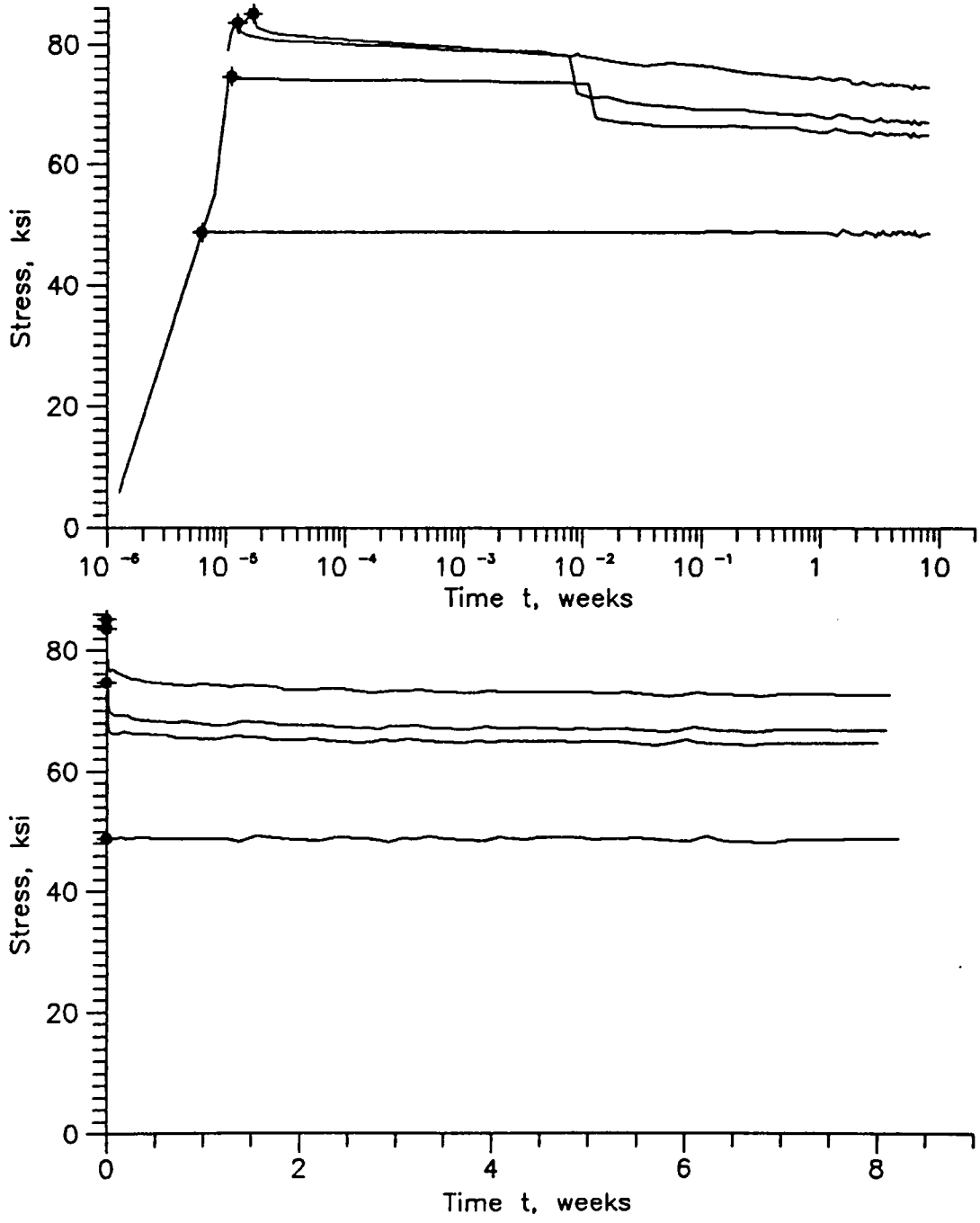


Figure 14. Static relaxation 7475-T651 for test duration.

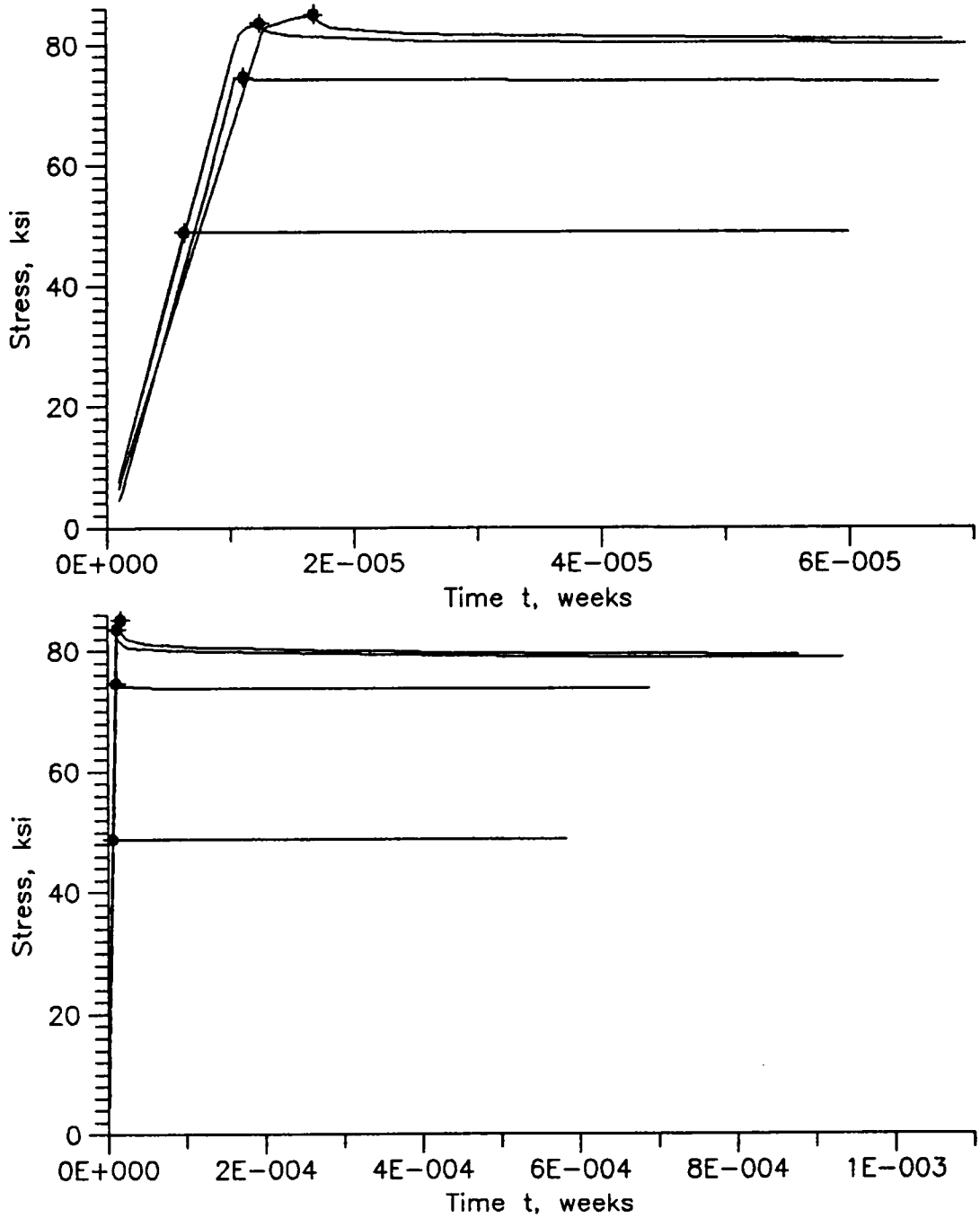


Figure 15. Static relaxation 7475-T651 initial transients.

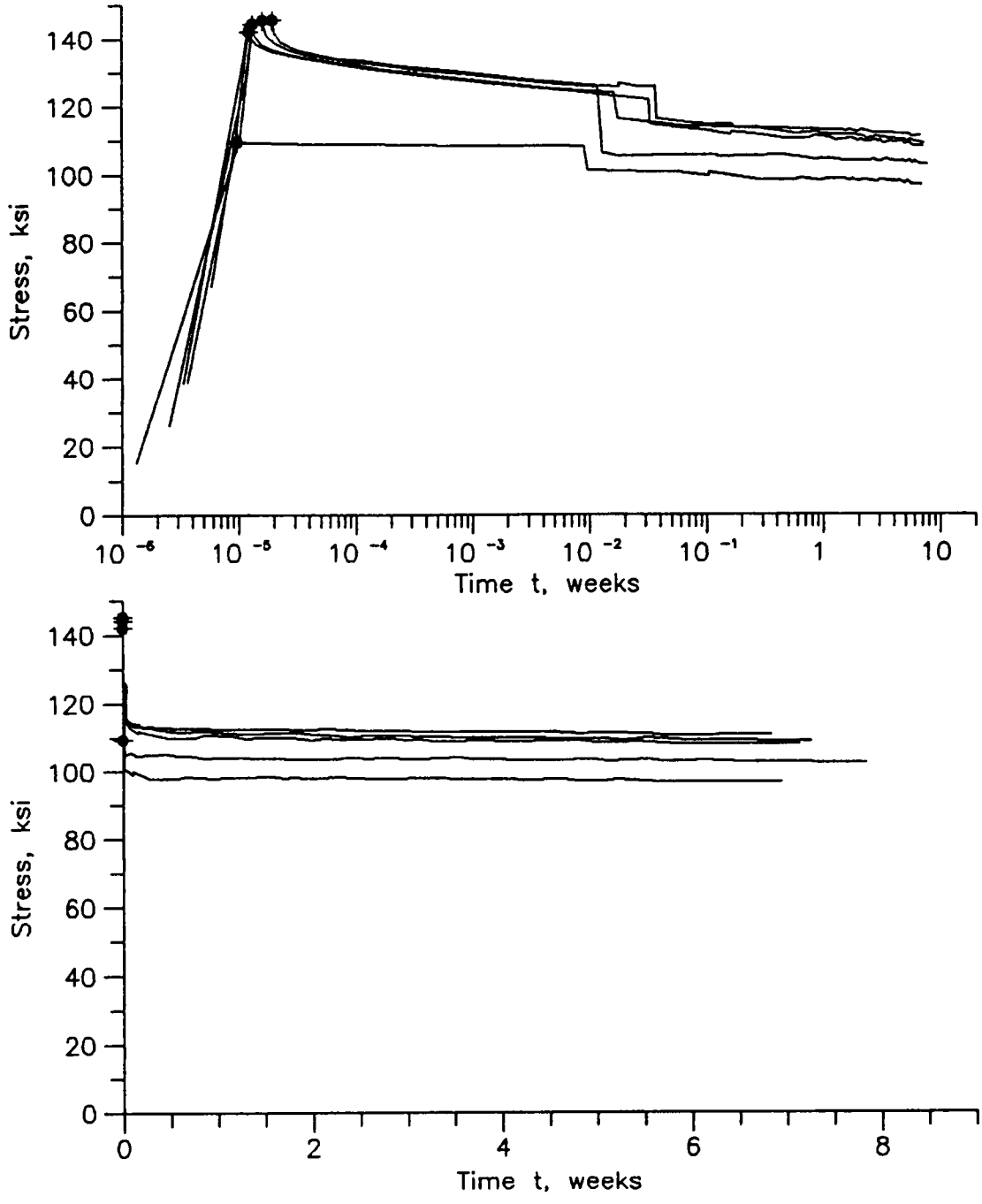


Figure 16. Static relaxation Ti-6Al-4V for test duration.

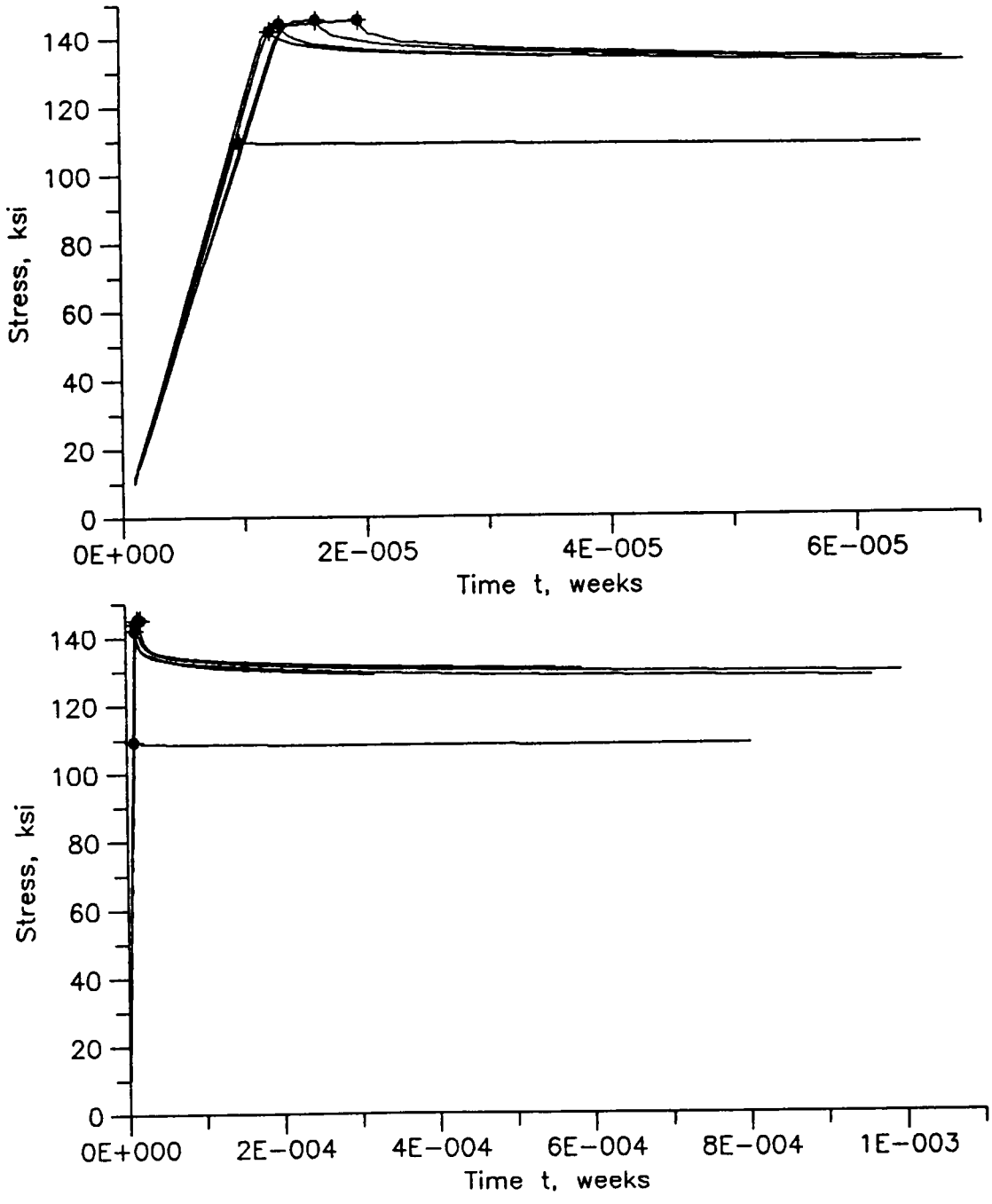


Figure 17. Static relaxation Ti-6Al-4V initial transients.

Oscillations were also observed in the relaxation data. The oscillations could be accounted for due to the discrete quantization of the 3 digit display and was about $\frac{1}{2}$ ksi. However, the variation may be attributed to the thermal dimension changes of the relaxation frame since all specimens approximately followed the same time history for the oscillations.

Examination of the semilog plots in Figures 15 and 17 show a step corresponding to a drop in the stress. The step occurred when the specimen was removed from the servo-hydraulic testing machine and the nuts tightened to maintain the load as discussed previously.

6.0 X-RAY STUDIES

6.1 EXPERIMENTAL APPARATUS

The TEC stress analyzer was used in this study to determine the residual stress. The d-spacing versus $sin^2\psi$ method, given by Equation 1, was used in the TEC stress analysis. The machine parameters for the x-ray measurements are given in Tables 5 and 6.

TABLE 5. X-ray Equipment Parameters

TEC x-ray stress analyzer

Item	Material	
	Ti-6Al-4V	7475-T651 Al
Radiation	Cu	Cu
λ, Å	1.54178	1.54178
(hkl) plane(s)	(213)	(511) and(333)
2θ, degrees	142	160
Voltage, kV	37, 45	37, 45
Amperage, mA	1.85	1.85
Rectangular[1] Slit Size, mm	0.5 by 5 1 by 5 2 by 5	0.5 by 5 1 by 5 2 by 5
Measurement Time (per stress value), min.	20 to 60	5 to 60

[1]5mm dimension parallel to specimen thickness.

TABLE 6. X-ray Slit Sizes for Notched and Unnotched Specimens

TEC x-ray stress analyzer

Notch Radius inches	Slit Size[1,2,3] mm
$\frac{1}{8}$	0.5 by 5
$\frac{1}{4}$	0.5 by 5 1 by 5
∞ (i.e. unnotched)	0.5 by 5 1 by 5 2 by 5

[1]5mm dimension parallel to specimen thickness.
[2]Largest slit area used to reduce measurement time.
[3]Maximum slit size is limited by errors due to stress gradients.

6.2 X-RAY ELASTIC CONSTANT DETERMINATION

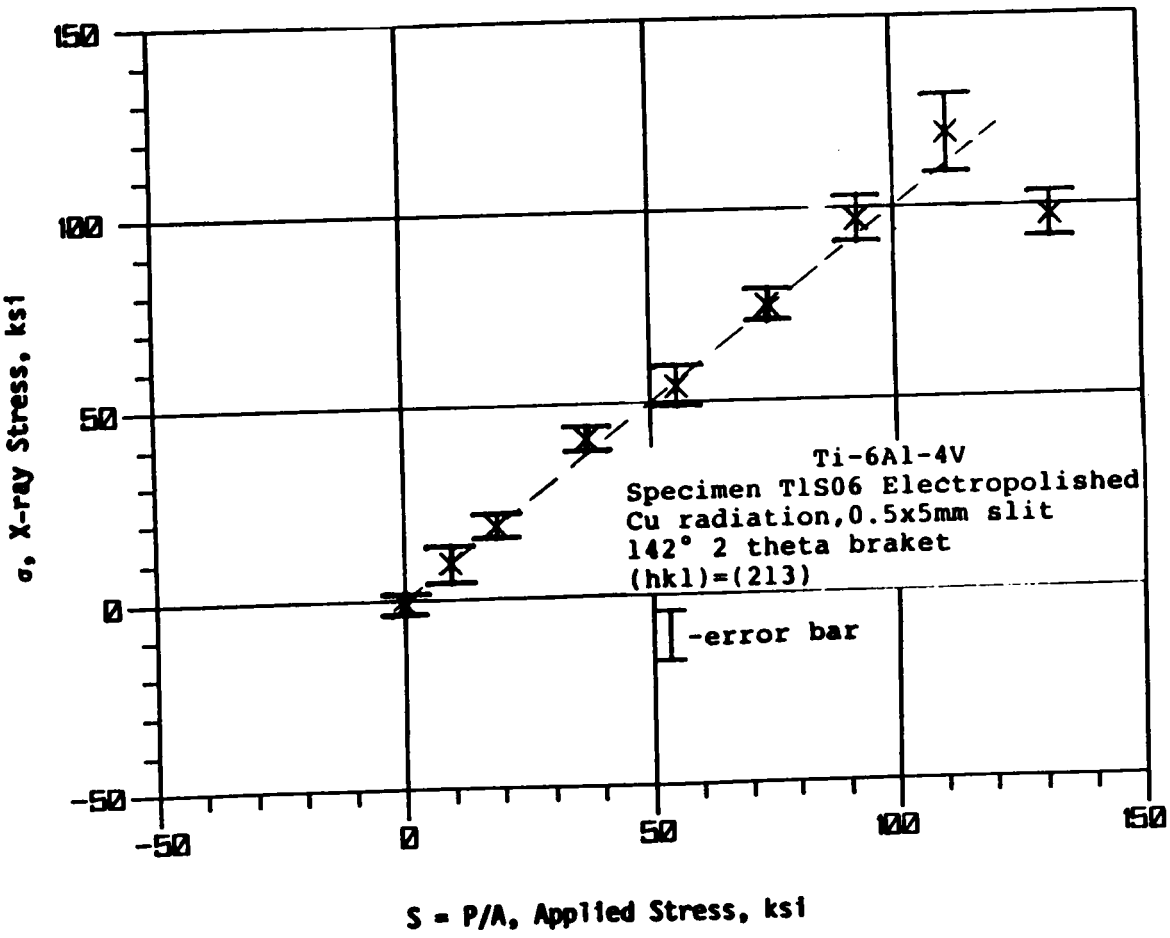
The x-ray work required the determination of the x-ray elastic constant, which is important especially since an appropriate value for the aluminum measurements done in this study was not tabulated in the literature. Graphical comparisons of x-ray stress and net section nominal stress were made. The specimens were of uniform cross section. In the graphical comparison, a discrepancy during yielding is discussed elsewhere. The x-ray results compared favorably with the net section stress for the titanium alloy in the elastic loading range. The tabulated stress analyzer configuration and material constant was used in this case. The results for the aluminum alloy show consistently high x-ray stress values compared to the net section average stresses during elastic loading. This time a configuration of the stress analyzer was utilized that did not have a tabulated x-ray elastic constant. Therefore, a typical value was chosen from the many available for aluminum, and this value will be adjusted to correct the x-ray results. The x-ray elastic constants, once determined, will allow correct x-ray measurements.

X-ray measurements have been made on specimens of uniform cross section at various levels of known static load. Graphs were then made of resulting x-ray stress values versus P/A. The resulting slopes could then be compared.

For the Ti-6Al-4V titanium alloy, the x-ray values were accepted due to the good fit to the data as seen in Figures 18 and 19. Considering the error bars, the data were on the expected line. The error bars were the measurement errors reported by the TEC analyzer. Therefore, the value of P/A was within the region of error about the x-ray stress value. The agreement of P/A and x-ray stress was valid up to yielding. TEC's tabulated value x-ray elastic constant for the titanium alloy was used.

For the aluminum 7475-T651 Al, the configuration of the stress analyzer was not one that had a tabulated x-ray constant by TEC. Therefore, a temporary value was selected. The resulting plot of x-ray stress versus P/A indicated a deviation in the x-ray stress from P/A as seen in Figure 20. Figure 21 is used to discuss the deviation and a procedure to correct the x-ray elastic constant. Also, for measurement of stresses by x-ray with an inappropriate x-ray elastic constant, a discussion is given for correcting the x-ray stresses recorded without remeasuring.

A value had to be found for the x-ray elastic constant, $\frac{(1+\nu)}{E}$, which resulted in agreement of σ_{x-ray} with net-section stress, $\frac{P}{A}$, data for aluminum. As seen in Figure 21, the variation of plotted x-ray stress, σ_{x-ray}, versus net uniform section stress, P/A, was expected to be linear. The linearity is seen to be valid in Figure 20 for increments of elastic loading.



The dashed line corresponds to perfect agreement with the applied stress.

Figure 18. Agreement of x-ray with P/A for unnotched Ti-6Al-4V, specimen T1S06.

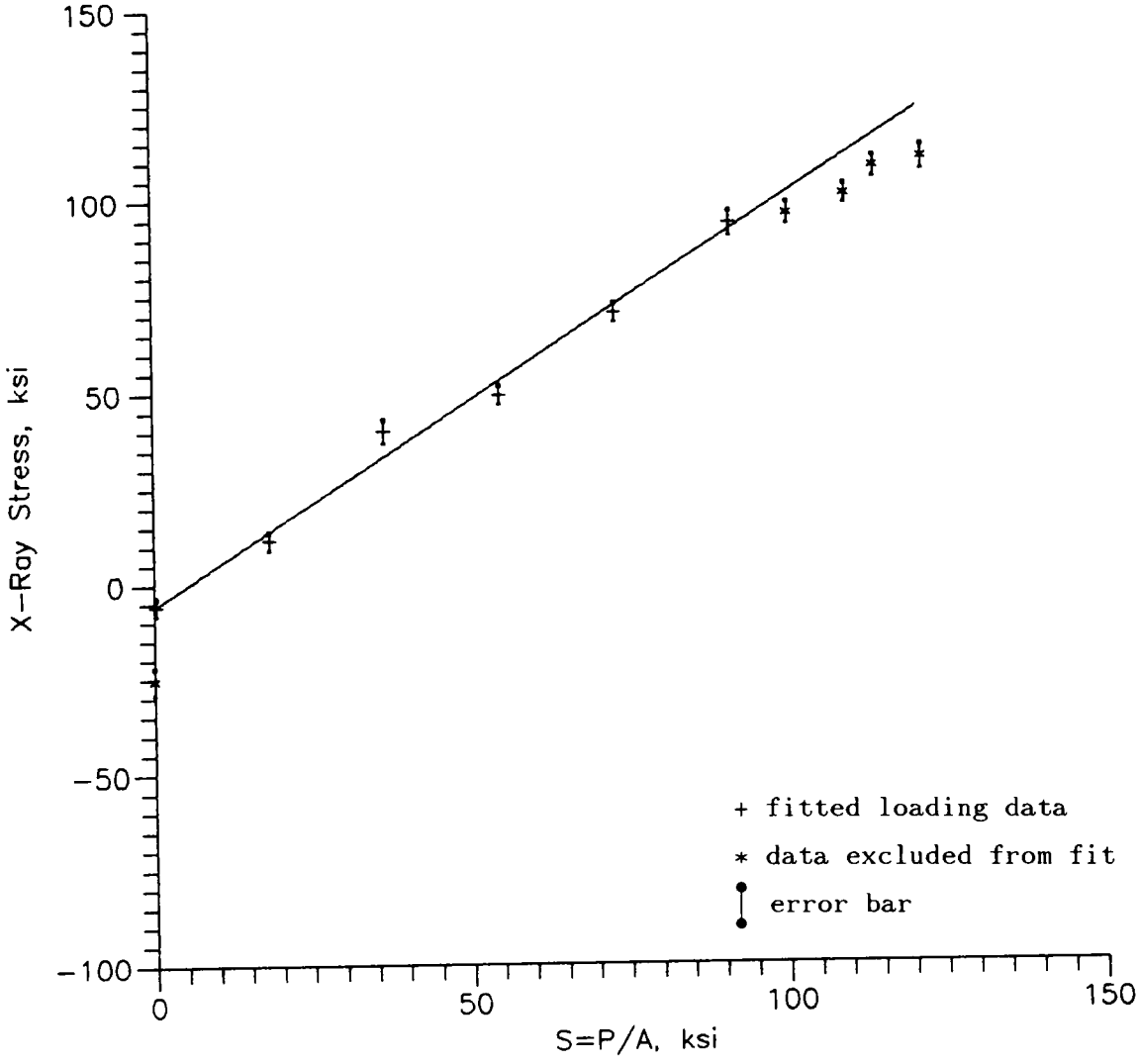


Figure 19. Agreement of x-ray with P/A for unnotched Ti-6Al-4V, specimen T1S02.

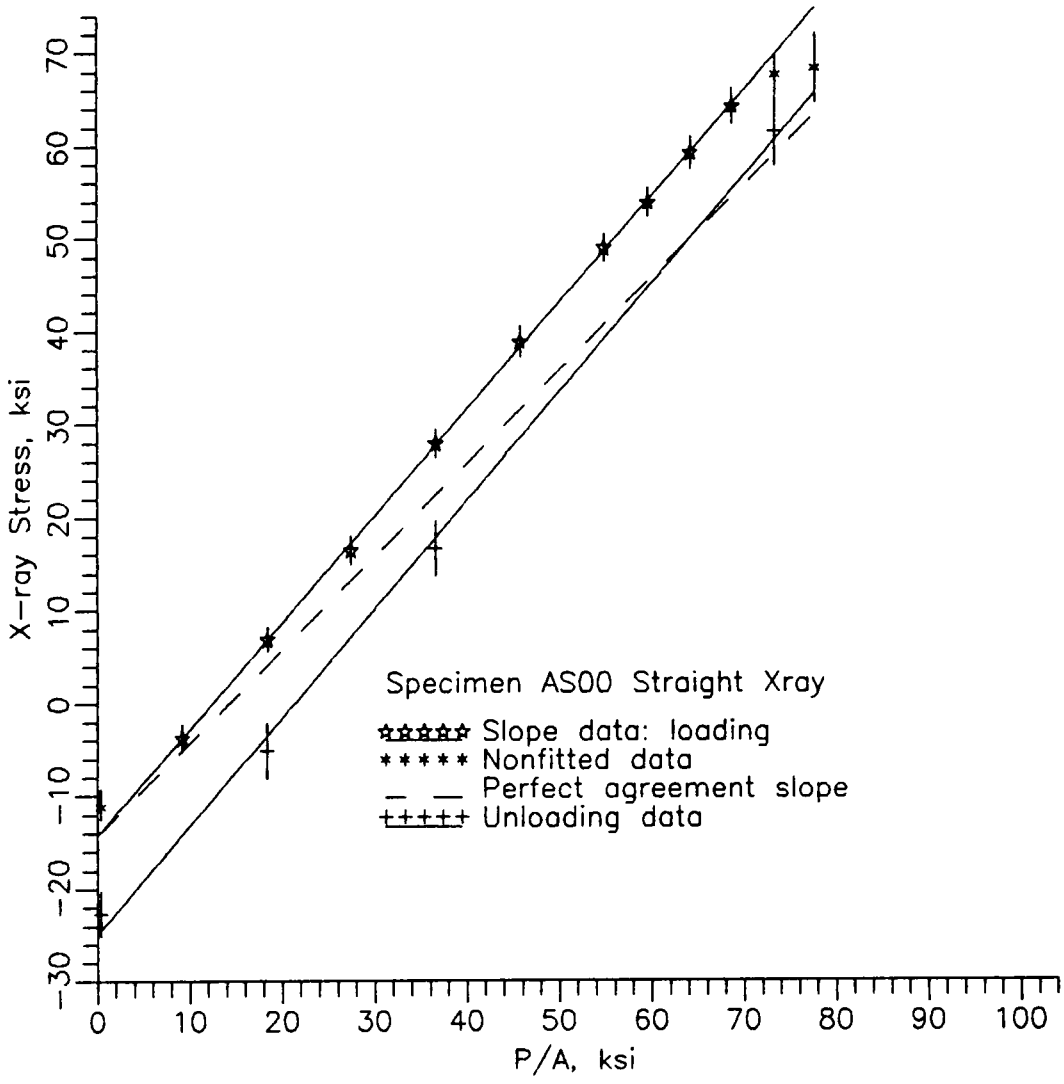


Figure 20. Agreement of x-ray stress with P/A for unnotched 7475-T651 Al.

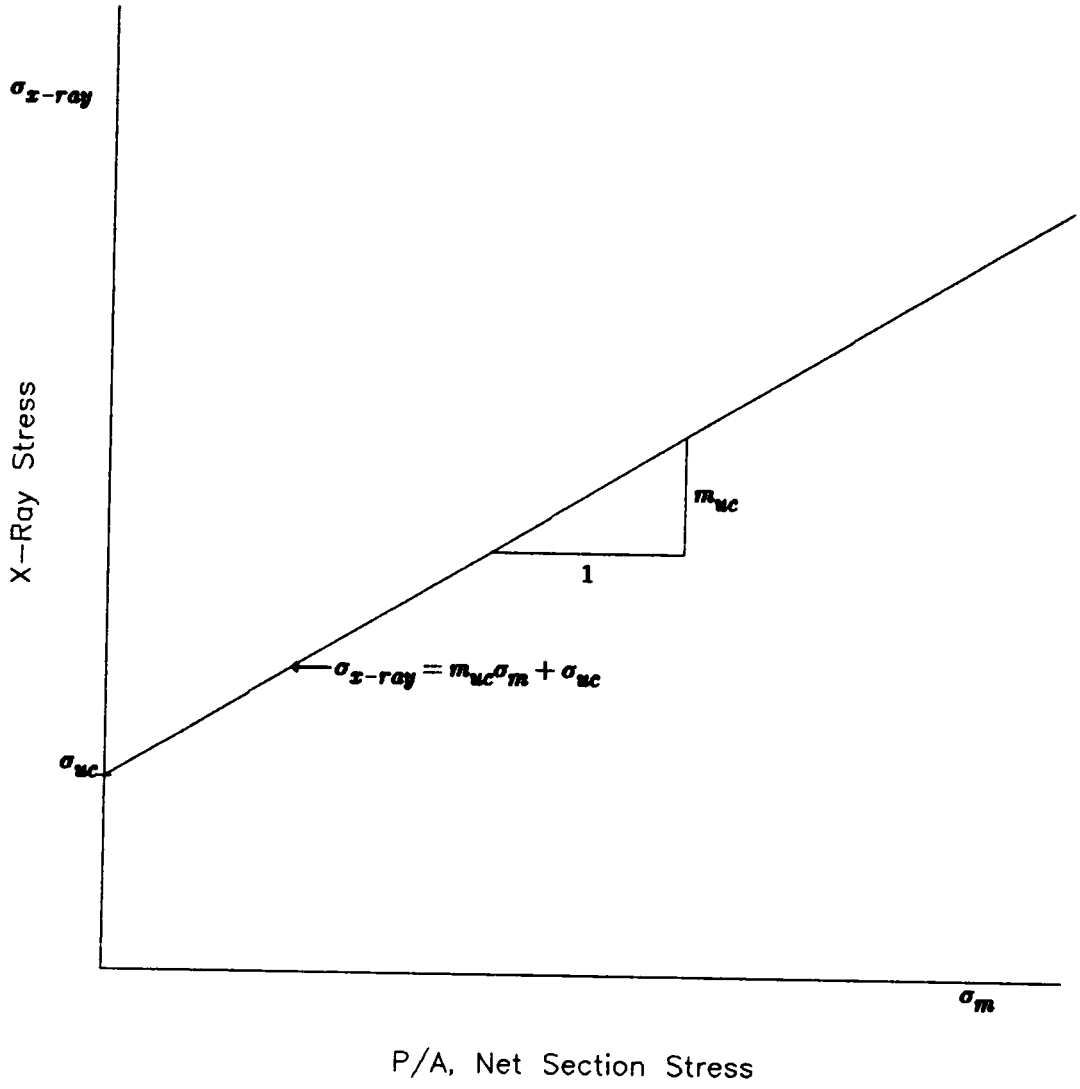


Figure 21. Illustration of x-ray stress versus *P/A*.

Changes in P/A should produce equal changes in σ_{x-ray} so that the slope of the line, m_{uc}, in Figure 21 should be equal to one, $m_{uc} = 1$. The values for σ_{x-ray} and P/A may not agree due to the presence of a residual stress measured by x-ray and not included in the calculated P/A value. In Figure 20, the dashed line indicates a slope of one. The solid lines show the trend of the data during elastic loading and unloading. Comparing the solid and dashed lines, a lack of agreement existed between σ_{x-ray} and P/A. A procedure for correcting the x-ray elastic constant to improve agreement is discussed in the following. Also, a correction of previously recorded x-ray stress is described.

Returning to Figure 21, the assumption was made that the linear relationship was valid but the slope was not equal to one. Then, the following equation was written.

$$\sigma_{x\ ray} = m_{uc}(\sigma_m + \sigma_r)$$

where

σ_{x-ray} —uncorrected x-ray stress

m_{uc} —slope uncorrected x-ray stress plot

$\sigma_m = P/A$ —applied mechanical stress

σ_r preexisting correct residual stress

then

$$\sigma_{x\text{-}ray} = m_{uc}\sigma_m + \sigma_{uc}, \quad \text{if} \quad \sigma_{uc} = m_{uc}\sigma_r$$

Physically σ_{uc} was the expected x-ray measured uncorrected residual stress. Continuing to rearrange terms

$$\sigma_m = \frac{1}{m_{uc}}\sigma_{x\text{-}ray} - \frac{\sigma_{uc}}{m_{uc}} = \frac{1}{m_{uc}}\sigma_{x\text{-}ray} - \sigma_r \tag{13}$$

An expression containing the correct x-ray value may be written while requiring the slope to be one by inspection.

$$\sigma_m = \sigma^{corr}_{x\text{-}ray} - \sigma_r \tag{14}$$

$\sigma^{corr}_{x\text{-}ray}$ was the desired correct x-ray value. Comparing Equations 13 and 14 the relationship for the correct stress may be written.

$$\sigma^{corr}_{x\text{-}ray} = \frac{1}{m_{uc}}\sigma_{x\text{-}ray} \tag{15}$$

Now, it may be determined how to correct the x-ray elastic constant. First, for the biaxial stress case, the d-spacing versus $sin^2\psi$ relationship, which was assumed to be valid, was written and manipulated. References [35, 36, 38] should be consulted for information concerning when the biaxial stress case is appropriate, and what difficulties make it invalid.

The equation was the following expression:

$$d_{\psi} = d_n \frac{(1+\nu)}{E} \sigma_{x-ray} \times sin^2\psi + d_n$$

Substitution of the corrected stress value was done by Equation 15.

$$d_{\psi} = d_n \frac{(1+\nu)}{E} m_{uc} \times \sigma_{x-ray}^{corr} \times sin^2\psi + d_n \tag{16}$$

Now, the relationship was rewritten by inspection assuming the correct value of the x-ray elastic constant was included.

$$d_{\psi} = d_n \left[\frac{(1+\nu)}{E}\right]^{corr} \sigma_{x-ray}^{corr} \times sin^2\psi + d_n \tag{17}$$

Finally, the correction was determined by inspection to be

$$\left[\frac{(1+\nu)}{E}\right]^{corr} = \frac{m_{uc}(1+\nu)}{E} \tag{18}$$

Equations 15 and 18 were used to correct the x-ray stress value and the x ray elastic constant, respectively.

The procedure for determining the correction slope, m_{uc}, and the corrected x ray elastic constant, $\left[\frac{1+\nu}{E}\right]^{corr}$, is discussed for the data of Figure 20. A summary of the numerical results is given in Table 7. A separate least squares linear regression was done for the elastic loading and unloading data. The loading and unloading data

are identified by differing symbols in Figure 20 along with values which were excluded from the linear fits. Reasons for elimination of data from the fits were noted deviation from linearity due to plasticity and/or the error bars indicate that the data point was not on the computed line. The error bars are indicated by the vertical lines for each data point. They were reported by the x-ray analyzer software as an indication of the accuracy of the reported x-ray stress value.

From the least squares linear regressions, the slope, standard deviation for the scatter of the data about the line, and standard deviation of the slope were computed according to Reference [49]. From Table 7 and Figure 20, the reported standard deviations, and error bars were significantly smaller for the loading data. The values of slope, m_{uc}, along with the x-ray elastic constant, $\left[\frac{1+\nu}{E}\right]^{corr}$, for loading were chosen for this reason, and also due to the greater number of data points for loading. Using three times the standard deviation for the line fit slope, the accuracy of the chosen values were three decimal places. The error bars for the fitted loading data varied from a minimum of ± 1.3 to a maximum of ± 1.9 ksi. These values were comparable to three to five times the scatter standard deviation of 1.19 to 1.98 ksi, respectively, for the fitted loading line. As a further comment, for unloading, the error bars for the fitted data varied from a minimum of ± 2.4 to a maximum of ± 3.7 ksi.

TABLE 7. X-ray Elastic Constant Correction 7475-T651 Al.

Uncorrected $\frac{(1+\nu)}{E}$ (psi^{-1})	1.255×10^{-7}	
	Loading	Unloading
m_{uc}	$(1.146349)^1$	1.166277
Standard Deviation Scatter Fitted Line (ksi)	0.396435	1.62723
Standard Deviation Slope, m_{uc}	0.006633	0.30123
$\left[\frac{1+\nu}{E}\right]^{corr}$ (psi^{-1})	$(1.4387 \times 10^{-7})^2$	1.46368×10^{-7}
Percent Relative Error m_{uc}	$(1.74)^3$	

[1] Value chosen as the correction slope.

[2] Value chosen for the corrected x-ray elastic constant.

[3] Loading value of m_{uc} chosen as the reference value and compared to the unloading value.

These values were comparable to the one to three times the scatter standard deviation of 1.63 to 4.88 ksi, respectively, for the fitted unloading line. Finally, the 1.74 percent relative error between the slopes for loading and unloading indicates there was good agreement between the slopes, even though there were a limited number of unloading data points with larger error bars. The good agreements of the fits would seem to support the use of the values of m_{uc} and $\left[\frac{1+\nu}{E}\right]^{corr}$ during loading for determining all elastic x-ray stress, σ_{x-ray}, values.

6.3 POLE FIGURES Ti-6Al-4V AND 7475-T651 Al

Texture was examined since the d-spacing versus $sin^2\psi$ relationship assumes the texture is uniform with no preferred orientation of the grains being measured. The textures of the wrought Ti-6Al-4V and 7475-T651 Al alloy plates chosen was measured by x-ray diffraction. Lambda Research Incorporated, Reference [50], performed the actual pole figure measurements. Figure 22 displays the x-ray sample geometry in the pole figure coordinate system. However, pole figure measurements were not made directly on the x-ray specimen, but measurements were performed on a representative rectangular slab machined from the same plate as the x-ray specimens. Final pole figure sample preparation was done by removing 0.004 inches of the surface by electropolishing to diminish texture changes due to machining.

The geometry of the sample coincided with the pole figures as follows: [1]the rolling direction, RD, is along the specimen axis, [2]the plane of the pole figure and the edge surface of the specimen were parallel lying in the plane formed by the transverse direction, TD, and rolling direction axes, [3]the normal direction, ND, was perpendicular to the edge of the specimen, [4]the pole figure covered psi tilts of up to 70° from the normal direction and a 360° revolution about the normal direction, [5]in the pole figure, x-ray stress measurements cover psi tilts of up to 30° ± 5° from the normal direction lying along the line formed by the rolling direction.

Figure 23 gives the details of the coordinate system in the plane of the pole figure, and the amount of the psi tilt was given by concentric circles. The direction to points in the plane of the pole figure was given by radial lines from the center. The dashed circles indicated the region covered by the x-ray stress analyzer, while the dashed line along the rolling direction indicated the path taken by the x-ray stress analyzer during stress measurements in the pole figure. All angles were in degrees.

The following was done to collect the pole figure data in the procedure done by Lambda[50]. The 7475-T651 Al sample was oscillated in the plane of the pole figure. Oscillation was not done on the Ti-6Al-4V sample due to inappropriate sample size. The intensity data were taken at 0.16° intervals in psi.

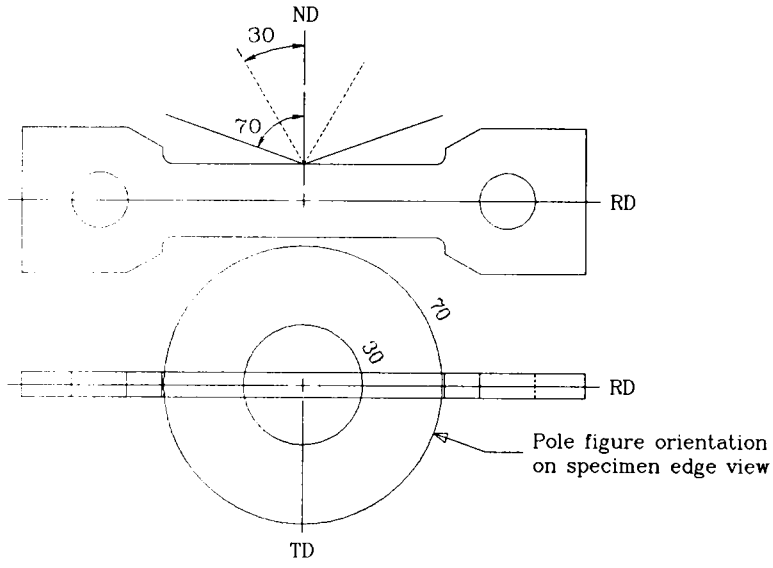


Figure 22. Pole figure and x-ray sample orientation.

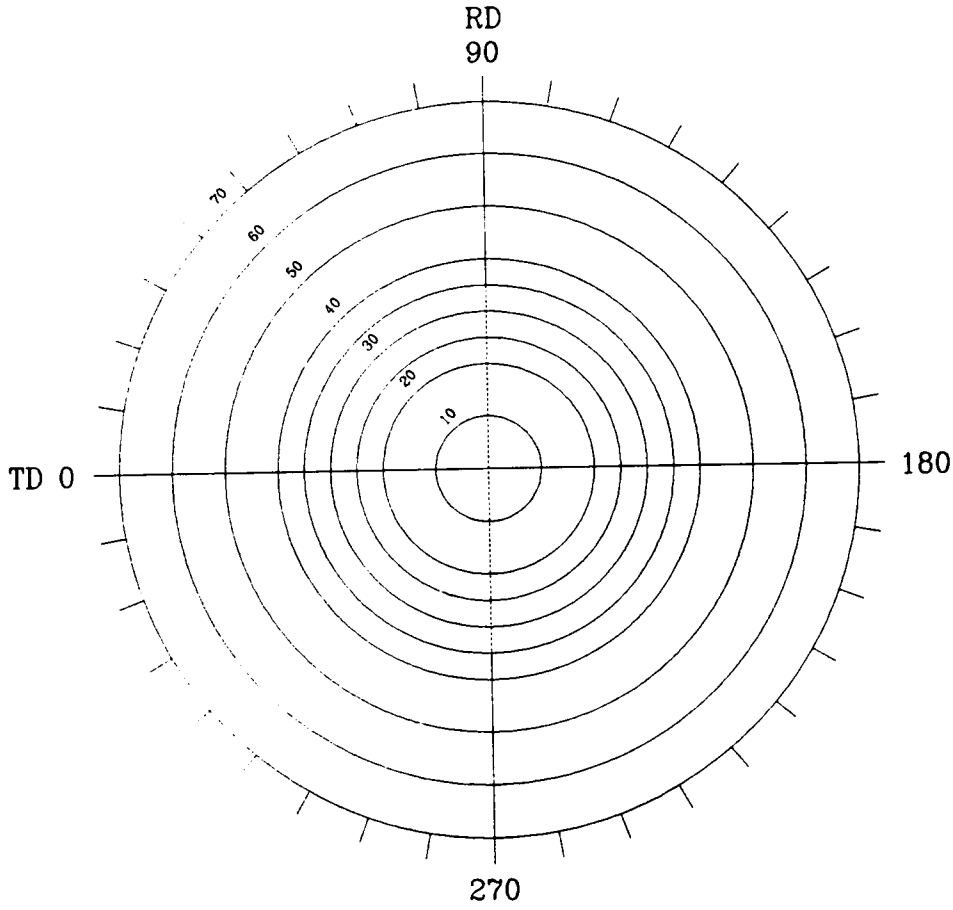


Figure 23. Pole figure coordinate grid, angles in degrees.

Also, intensity was measured in 12° intervals, in the plane of the pole figure, revolving about the normal direction. The recorded intensity data were smoothed by linear interpolation after correcting for background intensity.

Each measurement was plotted in two formats to produce pole figures. The times-random format scaled the diffracted intensities so that intensities above the average, integrated over the central 70°, were accentuated, and intensities below the average were attenuated. The accentuation and attenuation above and below the average was accomplished by expanding and compressing the plotting scale, respectively. The times-random format enables easier identification of texture and preferred orientation variation from random since comparisons are made against the integrated average intensity. The second format was a normalized format where the diffracted intensities were divided by the maximum intensity. The normalized format displayed the actual detailed variation in the intensity encountered during x-ray measurements.

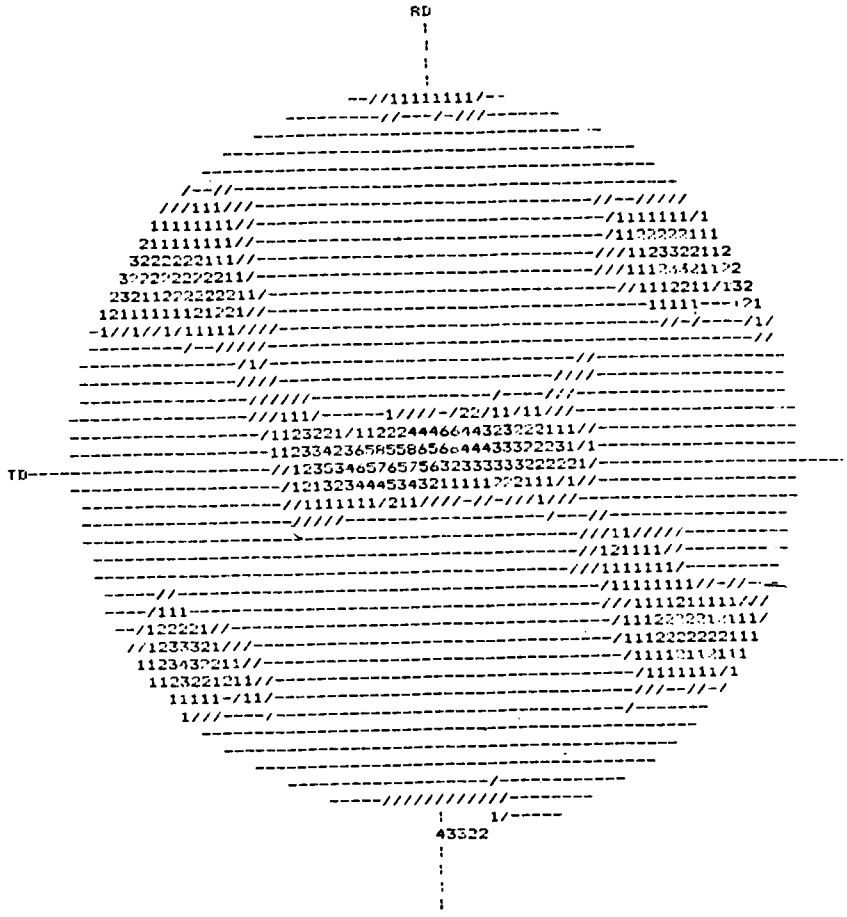
Figures 24, 25, 26, 27, 28, and 29 give the texture of the aluminum alloy. Figure 24, in the times-random format indicated a strong texture for the (111) family of planes. Figure 24 shows reflections centered on the center of the pole figure, and at the edges of the pole figure on the rolling direction and at approximately 30° above and below the transverse direction. A cube positioned looking down one corner would give reflections in the islands of

intensity in Figure 24. Figure 28 shows a cube positioned on its corner and the points corresponding to the reflections from this single ideal cube. The cube was also placed so that a diagonal plane of the cube contains the rolling direction, and two corners of the cube are on the normal direction. The center of the (111) reflection is located at the center of the "{111},{333}" symbol. The given cube orientation and a 180° rotation of it about the normal direction may establish the major features of the aluminum alloy's (111) pole figure. Also in Figure 28 and its 180° rotation, the ideal reflections for (311) are located in the islands of intensity given in Figures 26 and 27.

Figure 29 displays the intensity variation during x-ray stress measurements for the aluminum alloy where diffraction measurements were made from the (511) and (333) planes. In Figure 29 over a range of psi tilt of up to around 40°, there was a peak in intensity near 0° and 40°, and a dip to minimal intensity about 20° to 30° in psi. Turning to the ideal reflections given in Figure 28 along the rolling direction, the peak near 0° in psi corresponded to the (333) reflection while the peaks near 40° were in the region of the (511) reflection. In the region between 0° and 40° of psi tilt along the rolling direction, there were no reflections for (333) or (511) where in Figure 29 the intensity was minimal. The island of intensity along the rolling direction of Figure 25 located the (333) reflection about 0° psi since the (111) and (333) reflections were coincident. Figure

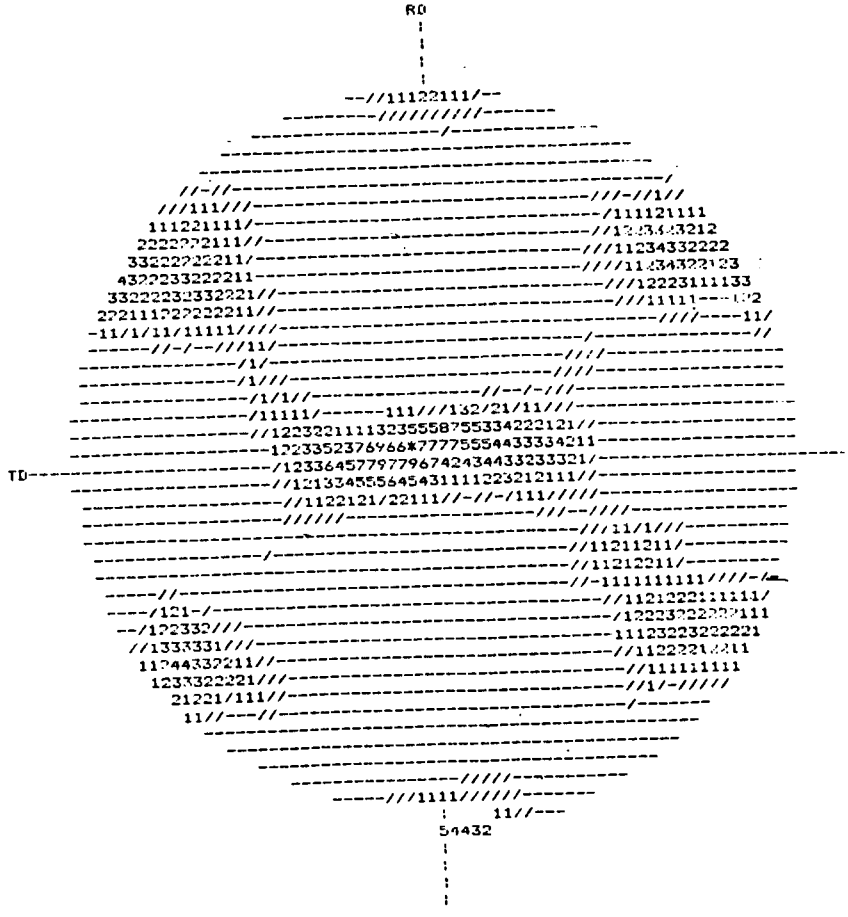
25 also displays the absence of intensity along the rolling direction away from 0° psi for the (333) reflection when measuring x-ray stress. From Figures 25, 28, and 29, the (333) or the (511) planes were being sampled depending on the psi tilt which could affect the results of x-ray stress measurements.

Figures 30, 31, 32, 33, 34, and 35 give the texture of the titanium alloy for the hcp crystal. Figure 30, in the times-random format indicates a strong texture for the (0002) family of planes. Figure 30 again shows reflections centered on the center of the pole figure, and at the edges of the pole figure at approximately 25° and 35° above and below the transverse direction. The (0002) reflections were from planes parallel to the basal plane of the hexagonal crystal for titanium. The island of intensity centered in the pole figure indicated that the base of the hexagon and, therefore, the basal plane was parallel to the plane of the pole figure for one orientation of the hexagon. This orientation was used in developing the pole figure of Figure 34 for a single ideal titanium crystal showing a top view of the basal plane. One of the orientations for titanium was to then look down the axis of the hexagon from the top. The islands of intensities on the perimeter of the (0002) pole figures indicated other positions for some of the hexagonal titanium crystals.



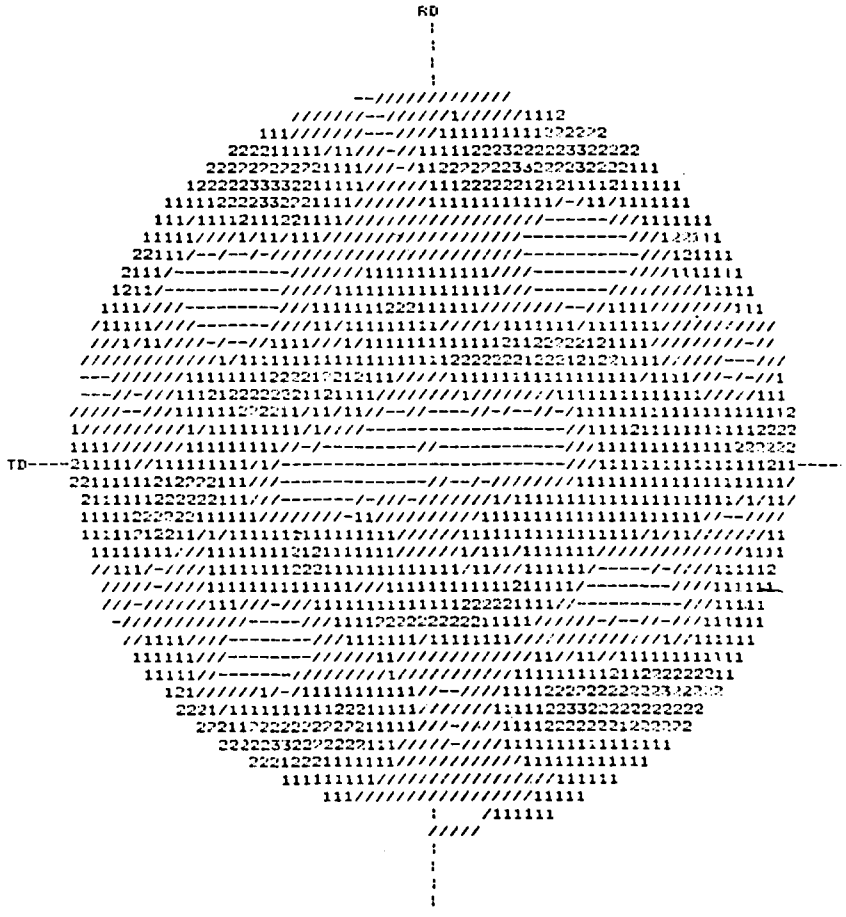
Symbol	$\frac{I}{\bar{I}}$	Symbol	$\frac{I}{\bar{I}}$	Parameter	Value
-	< 0.5	5	5–6	2θ	38.40°
/	0.5–1	6	6–7	I_{max}	20092 units
1	1–2	7	7–8	$\bar{I}$	2424 units
2	2–3	8	8–9	Background	1042 units
3	3–4	9	9–10	Radiation	Cu
4	4–5	*	> 10		

Figure 24. 7475-T651 Al times-random (111) pole figure. Reference [50].



Symbol	$\frac{I}{I_{max}}$	Symbol	$\frac{I}{I_{max}}$	Parameter	Value
–	< 0.05	5	0.5–0.6	2θ	38.40°
/	0.05–0.1	6	0.6–0.7	I_{max}	20092 units
1	0.1–0.2	7	0.7–0.8	$\bar{I}$	2424 units
2	0.2–0.3	8	0.8–0.9	Background	1042 units
3	0.3–0.4	9	0.9–1.0	Radiation	Cu
4	0.4–0.5	*	= 1		

Figure 25. 7475-T651 Al normalized (111) pole figure. Reference [50].

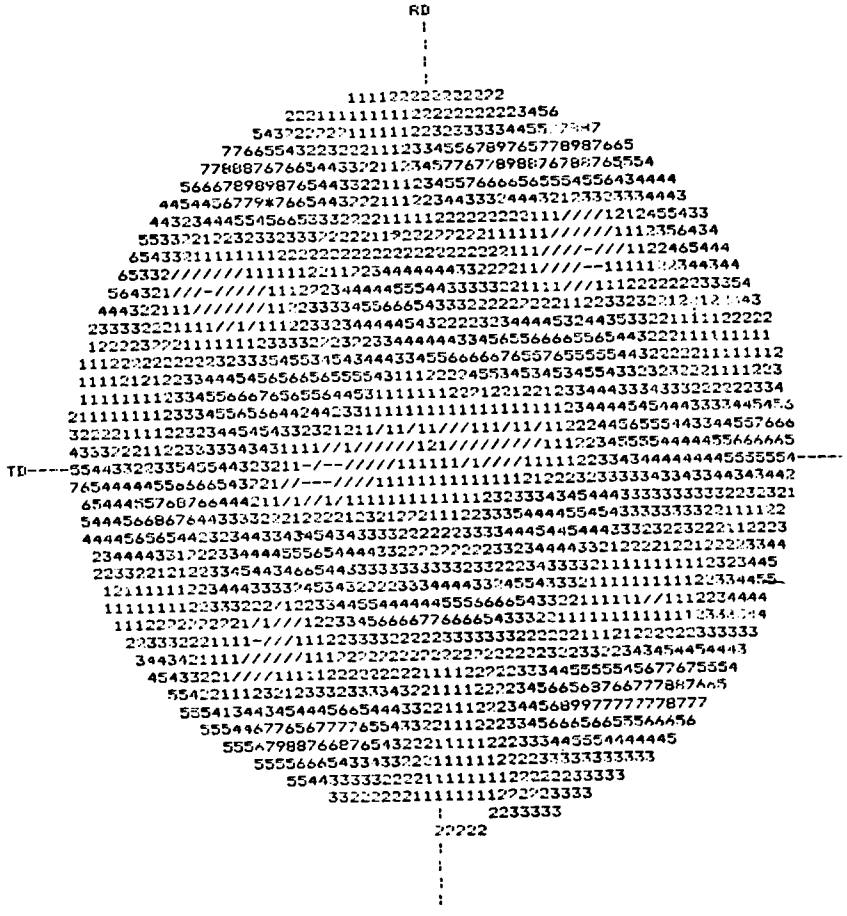


Symbol	$\frac{I}{\bar{I}}$
–	< 0.5
/	0.5–1
1	1–2
2	2–3
3	3–4
4	4–5

Symbol	$\frac{I}{\bar{I}}$
5	5–6
6	6–7
7	7–8
8	8–9
9	9–10
*	> 10

Parameter	Value
2θ	77.98°
I_{max}	78538 units
$\bar{I}$	22591 units
Background	4216 units
Radiation	Cu

Figure 26. 7475-T651 Al times-random (311) pole figure. Reference [50].



Symbol	$\frac{I}{I_{max}}$	Symbol	$\frac{I}{I_{max}}$	Parameter	Value
–	< 0.05	5	0.5–0.6	2θ	77.98°
/	0.05–0.1	6	0.6–0.7	I_{max}	78538 units
1	0.1–0.2	7	0.7–0.8	$\bar{I}$	22591 units
2	0.2–0.3	8	0.8–0.9	Background	4216 units
3	0.3–0.4	9	0.9–1.0	Radiation	Cu
4	0.4–0.5	*	= 1		

Figure 27. 7475-T651 Al normalized (311) pole figure, Reference [50].

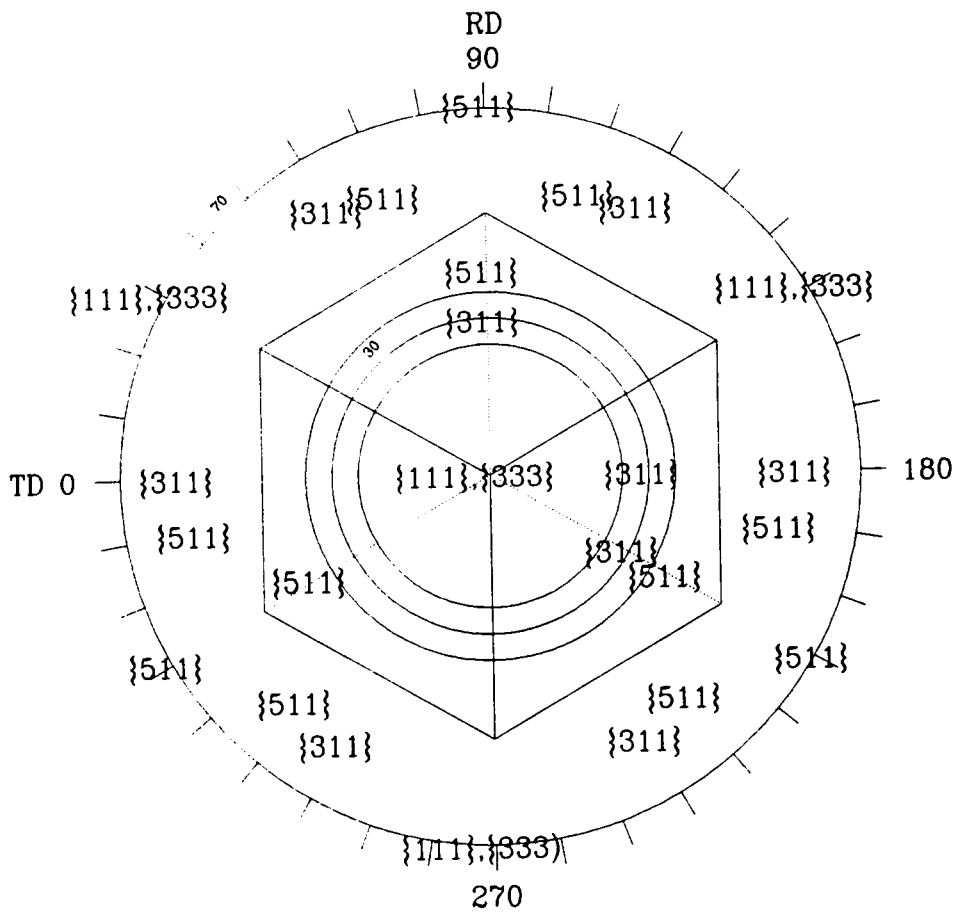


Figure 28. Aluminum crystal orientation and reflections pole figure, cubic crystal.

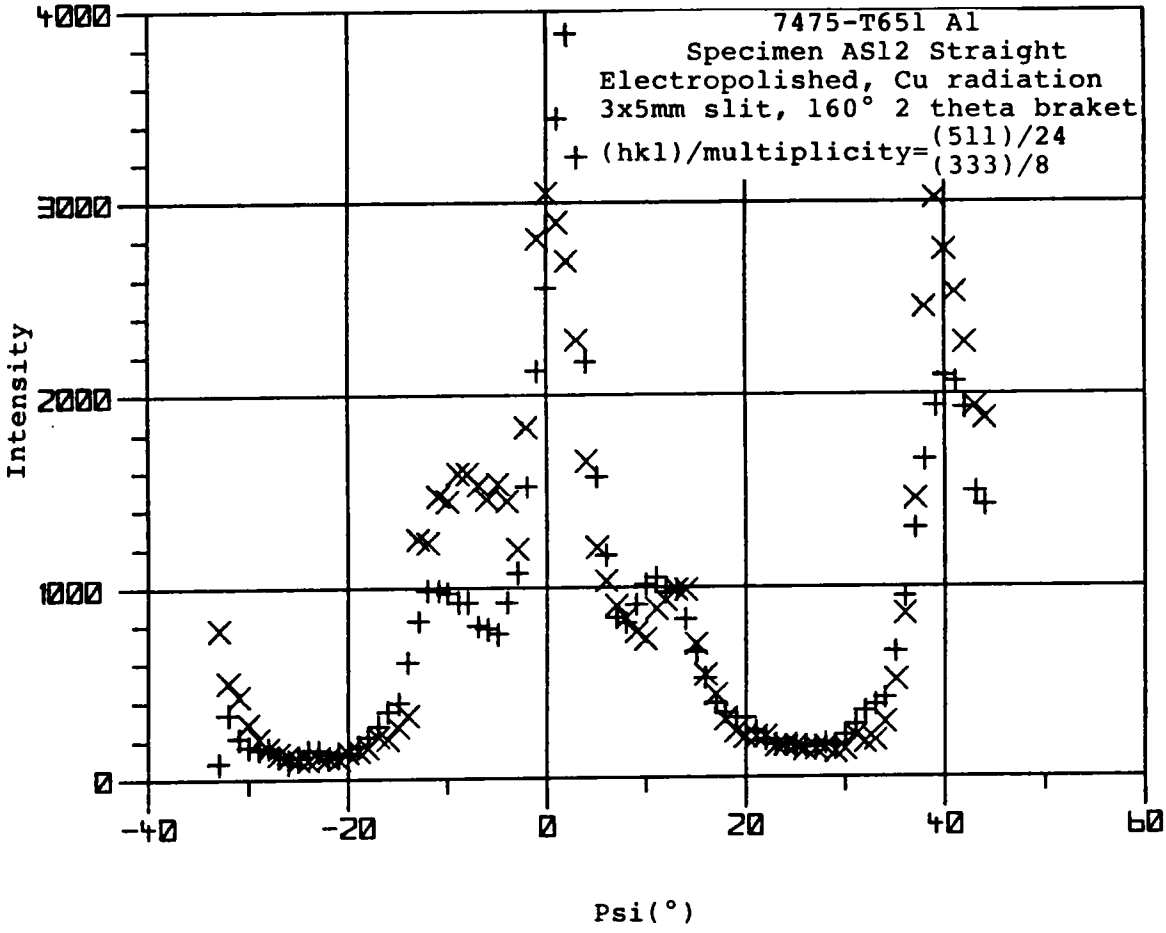
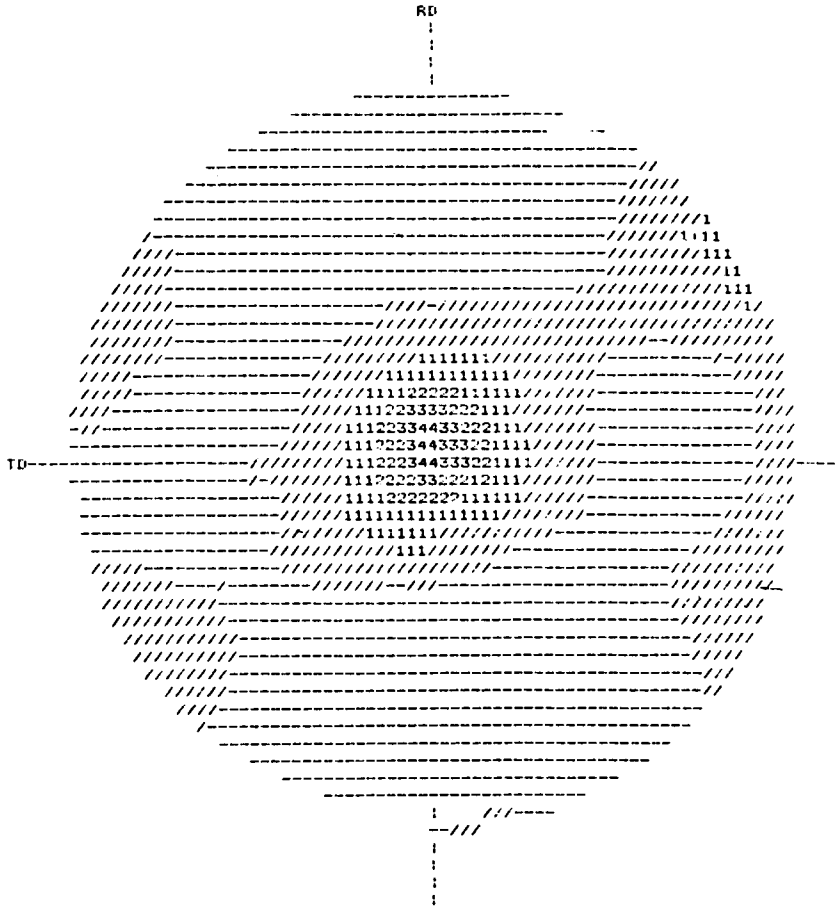


Figure 29. 7475-T651 Al intensity variation on the x-ray stress analyzer, for a large number of measurements.

Knowing that the basal plane was always perpendicular to a ray from the center of the pole figure to a point on the figure, there were positions of titanium crystals which were tilted from the normal direction of the pole figure. These tilted crystals were mostly tilted approximately 70° in psi and 25° to 35° away from the transverse direction. Since the (0002) reflection was only along the c axis of the hexagon, the central 70° Pole Figures 30 and 31 do not indicate all possible crystal positions because psi tilts for crystals of between 70° and 90° would not produce a reflection in the central 70° of the pole figure. In Figures 32, the times-random format gives islands of intensity around the perimeter and diagonally across the pole figure for the (011) planes. Comparing Figure 34 with Figure 32, the (011) reflections of Figure 34 may be made to lie in the islands of intensity of Figure 32 if one corner of the basal plane was pointed in the 0° position of the transverse direction. Also, note that for the crystal position given in Figure 34, all the (011) reflections are around the outside of the pole figure and away from its center. The orientation with the basal plane parallel to the plane of the pole figure cannot produce the diagonal island of intensity given in Figure 32. A tilted orientation such as those shown in Figures 30 and 31 or between psi tilts of 70° and 90° have to produce the diagonal island of intensity that appears in Figure 32.

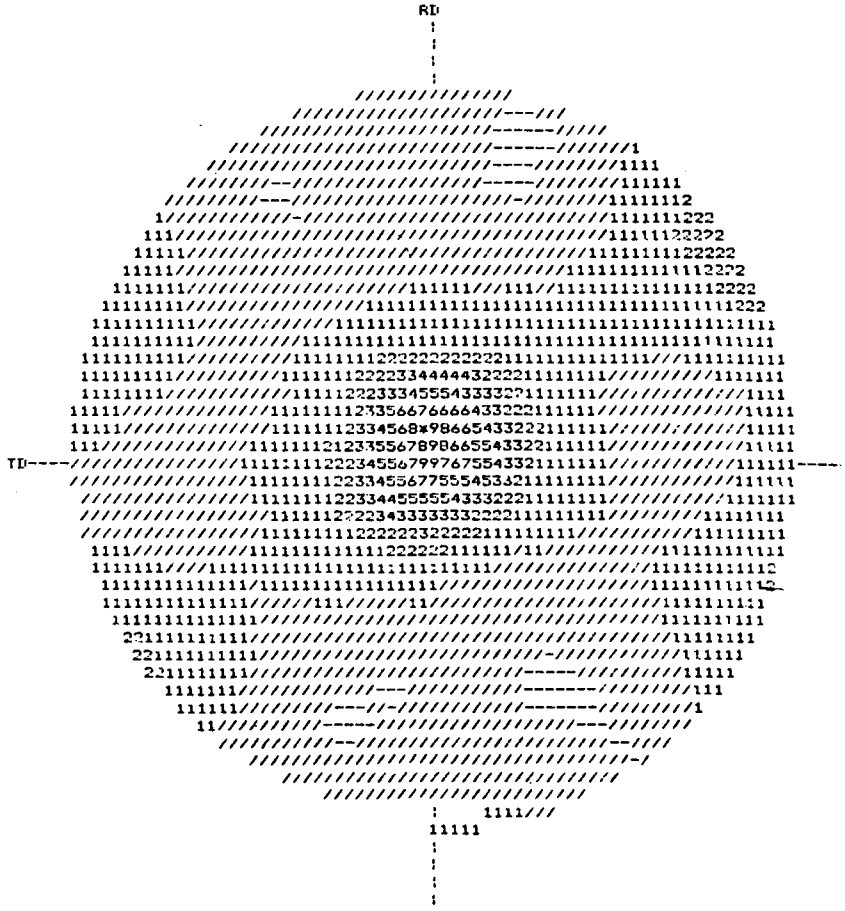
Figure 35 displays the intensity variation during x-ray stress measurements for the titanium alloy where diffraction measurements

were made from the (213) planes. In Figure 35 over a range of psi tilt of up to around 35°, the intensity had a minimum about 0° and gradually rises away from 0° for psi tilts. Turning to the ideal reflections given in Figure 34 along the rolling direction, there are no (213) reflections for the hexagonal orientation given in the central 30° psi tilt range over which x-ray stress measurements were made. The intensity in Figure 35 occurred from crystal positions tilted in psi. The tilted hexagons could either be those given by Figures 30 and 31 at the perimeter or those having tilts of between 70° and 90° for psi.



Symbol	$\frac{I}{\bar{I}}$	Symbol	$\frac{I}{\bar{I}}$	Parameter	Value
-	< 0.5	5	5–6	2θ	38.85°
/	0.5–1	6	6–7	I_{max}	12131 units
1	1–2	7	7–8	$\bar{I}$	2681 units
2	2–3	8	8–9	Background	946 units
3	3–4	9	9–10	Radiation	Cu
4	4–5	*	> 10		

Figure 30. Ti-6Al-4V times-random (0002) pole figure, Reference [50].

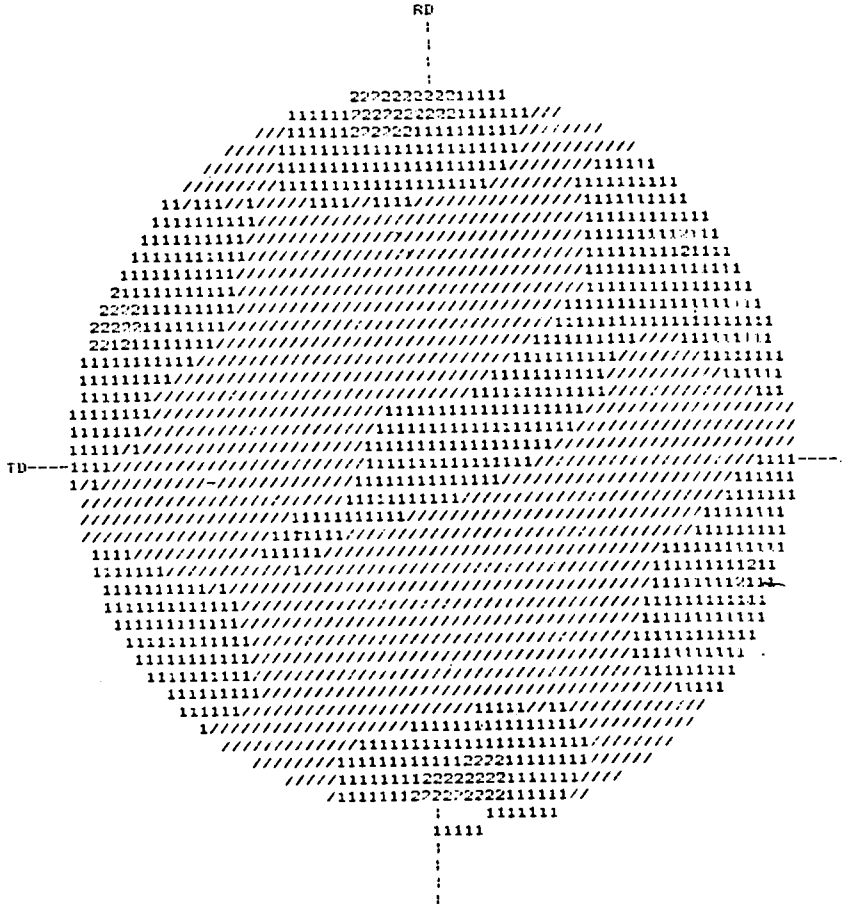


Symbol	$\frac{I}{I_{max}}$
–	< 0.05
/	0.05–0.1
1	0.1–0.2
2	0.2–0.3
3	0.3–0.4
4	0.4–0.5

Symbol	$\frac{I}{I_{max}}$
5	0.5–0.6
6	0.6–0.7
7	0.7–0.8
8	0.8–0.9
9	0.9–1.0
*	= 1

Parameter	Value
2θ	38.85°
I_{max}	12131 units
$\bar{I}$	2681 units
Background	946 units
Radiation	Cu

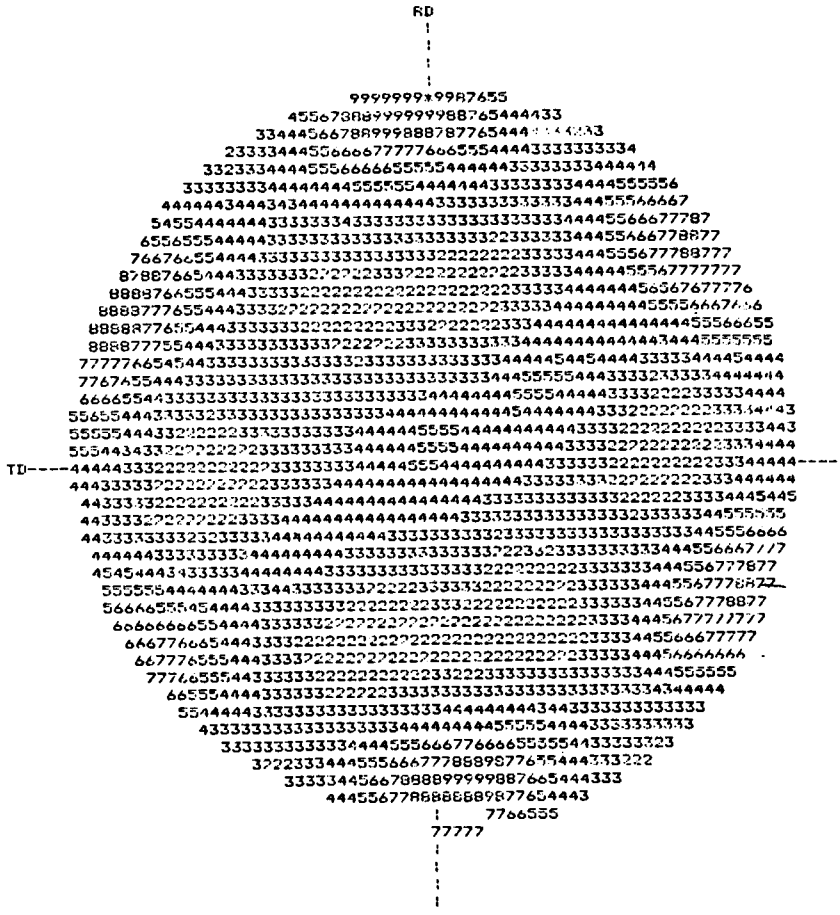
Figure 31. Ti-6Al-4V normalized (0002) pole figure. Reference [50].



Symbol	$\frac{I}{\bar{I}}$
–	< 0.5
/	0.5–1
1	1–2
2	2–3
3	3–4
4	4–5
5	5–6
6	6–7
7	7–8
8	8–9
9	9–10
*	> 10

Parameter	Value
2θ	40.42°
I_{max}	14540 units
$\bar{I}$	6075 units
Background	849 units
Radiation	Cu

Figure 32. Ti-6Al-4V times-random (011) pole figure, Reference [50].



Symbol	$\frac{I}{I_{max}}$	Symbol	$\frac{I}{I_{max}}$	Parameter	Value
–	< 0.05	5	0.5–0.6	2θ	40.42°
/	0.05–0.1	6	0.6–0.7	I_{max}	14540 units
1	0.1–0.2	7	0.7–0.8	$\bar{I}$	6075 units
2	0.2–0.3	8	0.8–0.9	Background	849 units
3	0.3–0.4	9	0.9–1.0	Radiation	Cu
4	0.4–0.5	*	= 1		

Figure 33. Ti-6Al-4V normalized (011) pole figure, Reference [50].

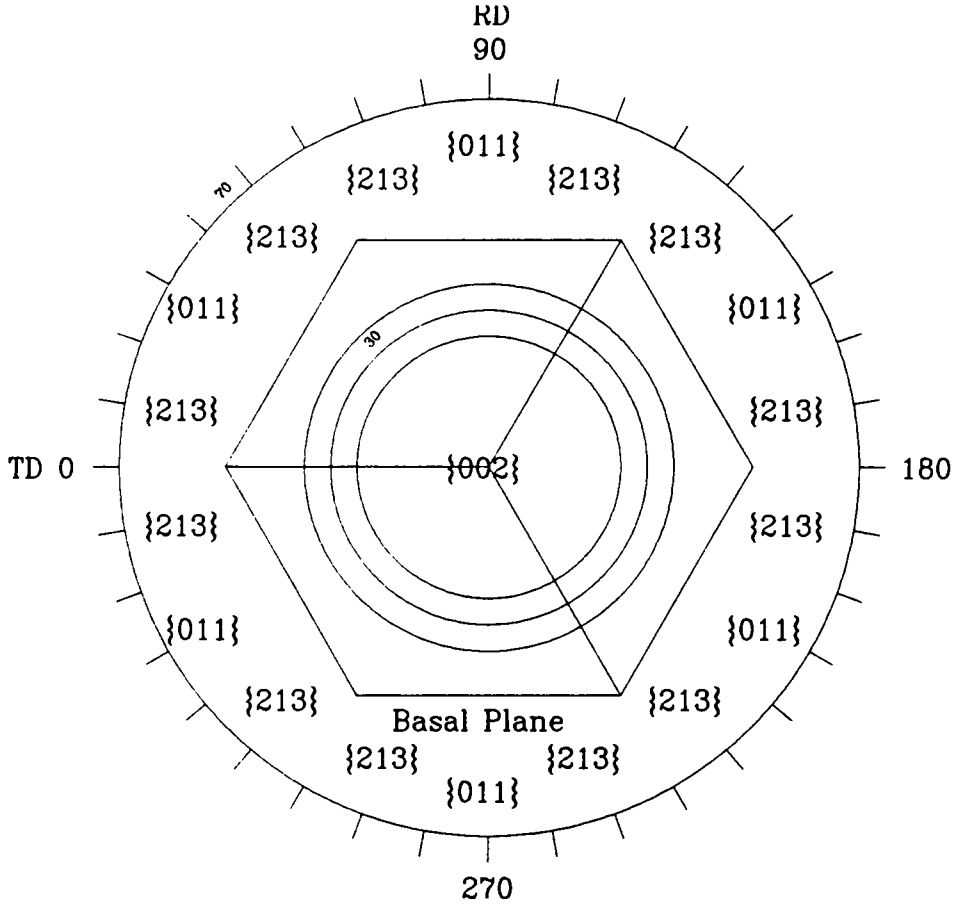


Figure 34. α-Titanium crystal orientation and reflections pole figure, hcp crystal, a=2.950 Å, c=4.686 Å.

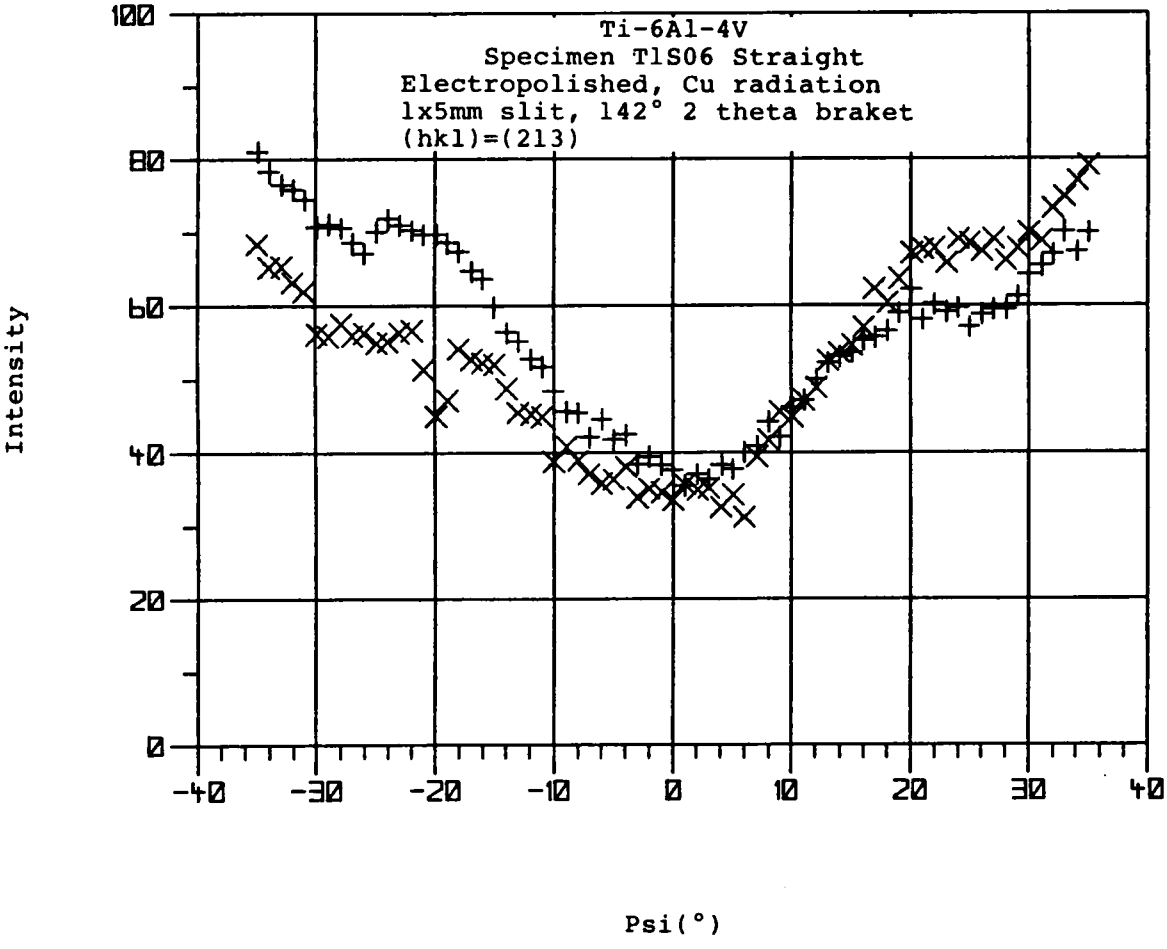


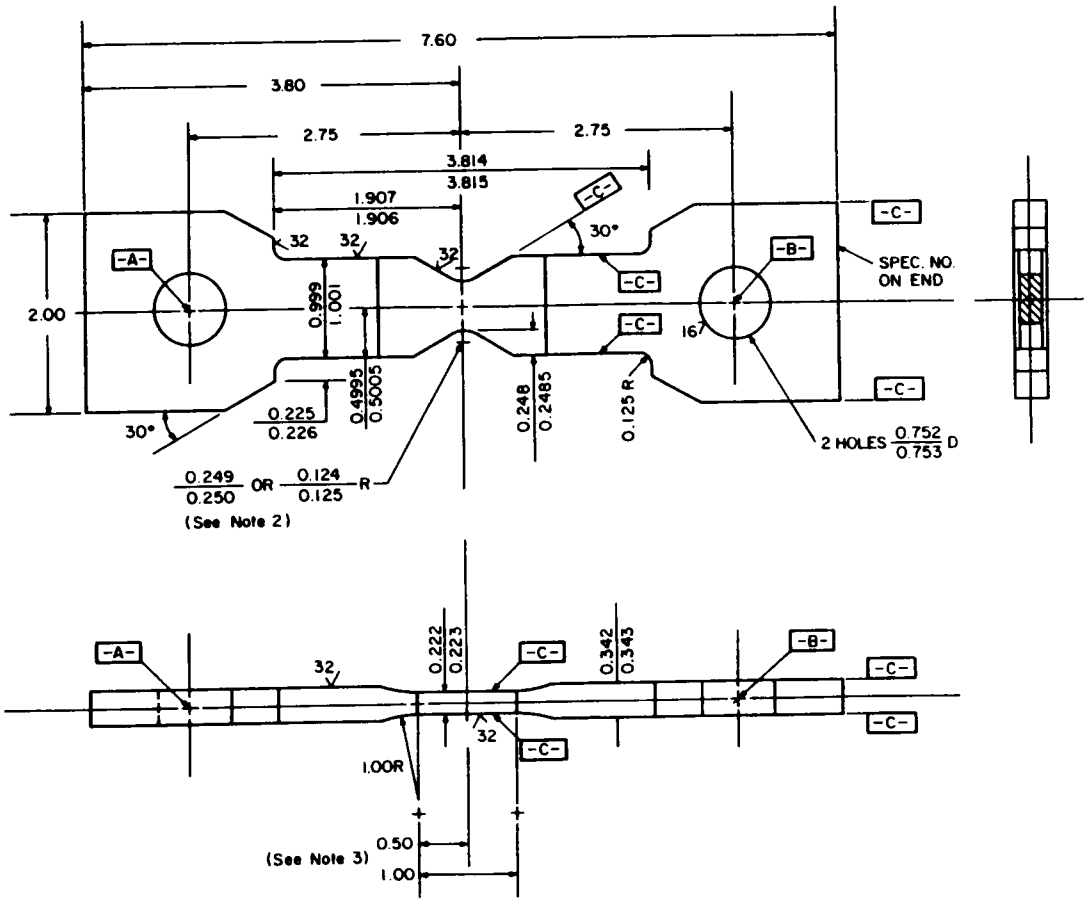
Figure 35. Ti-6Al-4V intensity variation on the x-ray stress analyzer, for a large number of measurements.

6.4 EXPERIMENTAL MEASUREMENTS

Monitoring the stresses in a stress raiser was sought. The variation of stress and the change in the residual and mean stress during cyclic loading was of interest. Changes were expected due to transient cycle and time dependent creep and relaxation along with cyclic hardening or softening.

X-ray stress measurements were made for the case of uniaxial loading. The uniaxial measurements were possible while under load due to advanced x-ray stress analysis. Specially designed uniform and notched specimens as in Figures 36 and 37 were used, loaded in a MTS servo-hydraulic test frame as shown in Figure 38. The reasons for the sample design are given in Reference [37]. While loaded, x-ray stress measurements were made, and are displayed in Figure 39.

The measurements were made during brief static pauses during loading. The duration of the pause varied from 5 minutes to 1 hour. Besides the specimen geometries, the only special preparation was removal of 0.002" of the surface by electropolishing. X-ray measurements under load were restricted to only tensile loading. However, residual stress measurements may be made after tension or compression loading. Figure 40 gives a typical x-ray report. The previously displayed Figure 7 gives the plot produced with an x-ray report.



Notes:

1. Surfaces labeled "C" to be parallel or angled to A-B, and mutually perpendicular, and equidistant from A-B, as applicable, within 0.001.

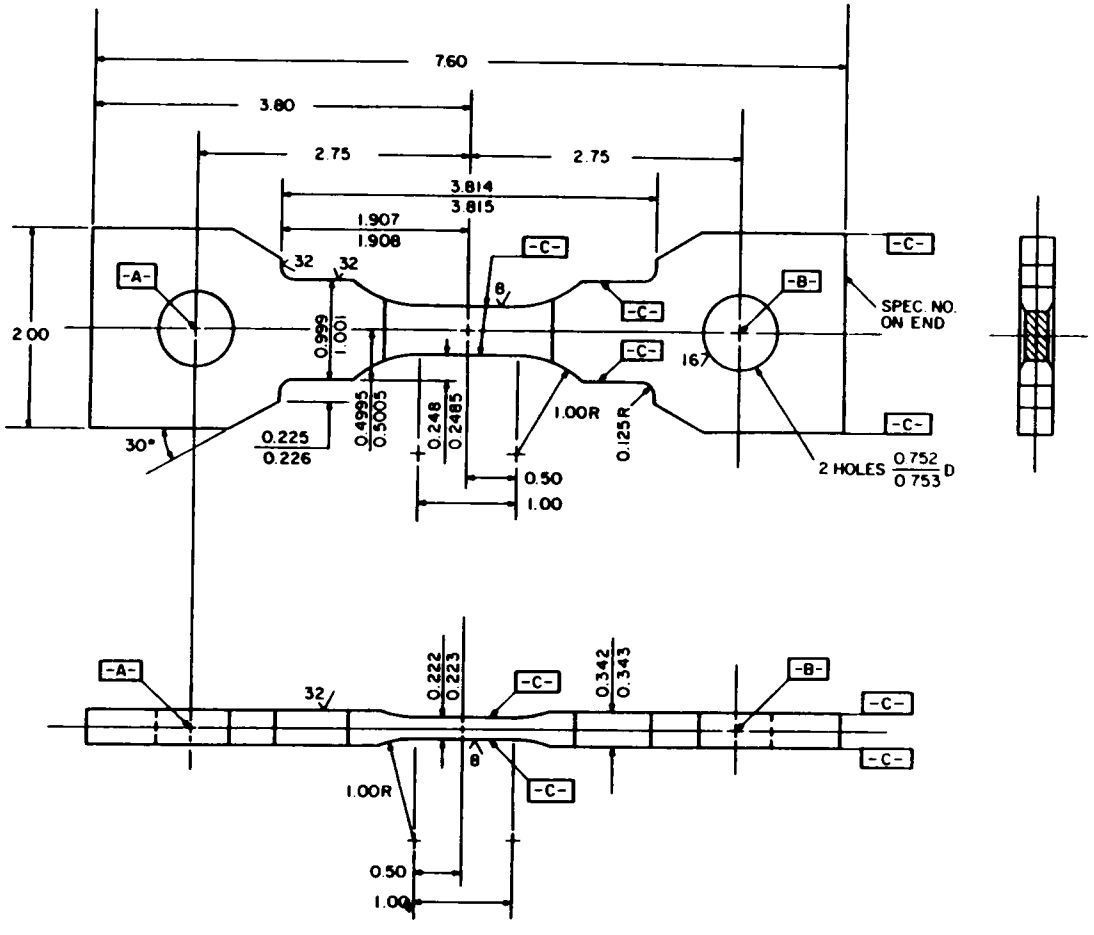
2. Use value of radius requested for each specimen. Radius 8 μin. finish. Remove 0.003 max on 2nd last pass, 0.002 max on last pass. Remove final 0.002 min by very light cylindrical grinding or honing.

3. Dimensions to tangency of 1.00R

4. Center section removed 0.002 min of surface by electropolishing.

5. Dimensions shown are inches.

Figure 36. Notched x-ray specimen geometry.



Notes:

1. Surfaces labeled "C" to be parallel to A-B, and mutually perpendicular, and equidistant from A-B, as applicable, within 0.001.

2. Straight center section 8 μin. finish. Remove 0.003 max on 2nd last pass. 0.002 max on last pass. Remove final 0.002 min by very light cylindrical grinding or honing. Final machining marks to be longitudinal.

3. Center section removed 0.002 min of surface by electropolishing.

4. Dimensions shown are inches.

Figure 37. Unnotched x-ray specimen geometry.

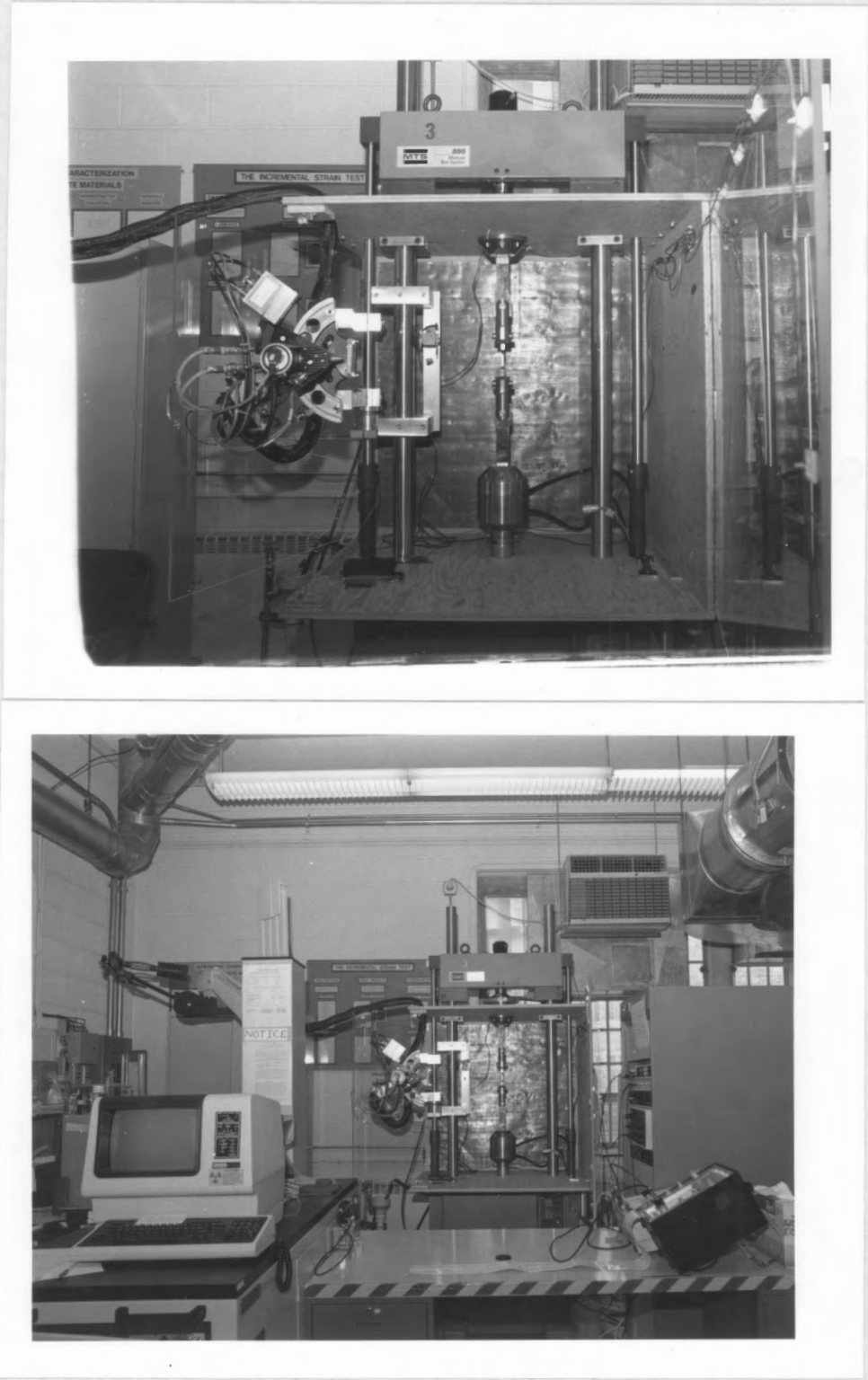


Figure 38. X-ray and MTS testing arrangement.

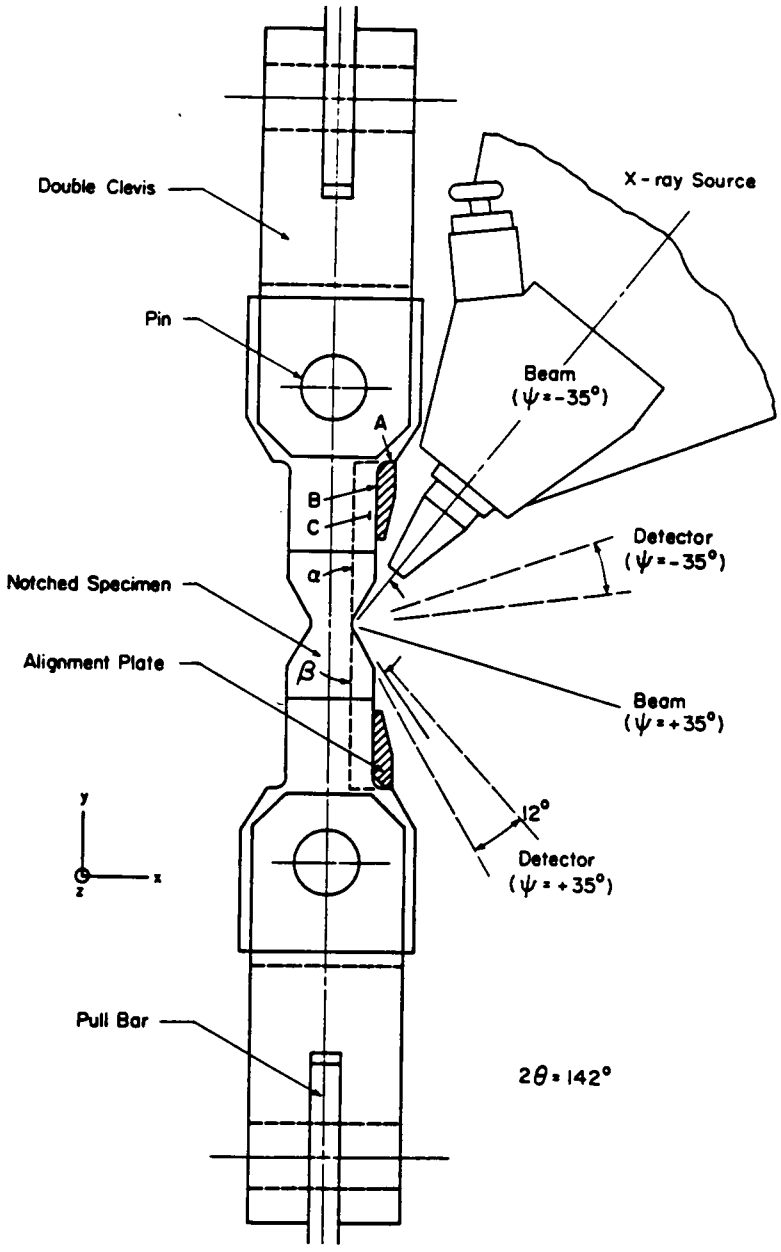


Figure 39. Arrangement of specimen, fixtures, and x-ray source, for measurements during mechanical testing.

********** Residual Stress Analysis Report **********

Date: 06-AUG-88 Time: 11:45:37

Sample Description :
A2R04A/EP/LOADED Specimen# away from detector PSPC 1578V
measurement# 1

System Hardware Configuration :
Auto Psi Angle Drive
Psi Angle Position Encoder

ADC Channels Full Scale		256
Collimator Slit Type -	Rectangular	2.00
X-ray Target Material and Wavelength	Copper	1.54178
Detector Mounting Block Bragg Angle		160.00
Oscillating Psi Angle		3.00
High Voltage and Beam Current	37000.	1.81
Peak Bounding Range (percent)		20.
Material ID Number		9
Material Type	Al Alloy	(Cr 311)
Stress Spectra File Specifications		000625.SPC
Stress Spectra Acquisition Date:	09-JUL-88	19:10:42
Stress Spectra Count Time (sec)		30
Calibration File Specifications		CLB2CU.160

Detector Calibration Coefficients
A 0.353147E-07 B -0.971201E-05 C 0.0592918 D 152.4716

Psi	Sin^2(Psi)	Pk Chan	Intens	FWHM	Kalp Cor	2-Theta	D Spacing	St. Dev.
-44.0	0.48524	133.67	219.8	1.46	0.00000	160.31	0.782414	0.000033
-30.0	0.26220	155.71	558.8	1.41	0.00000	161.60	0.780934	0.000015
-26.0	0.20409	157.62	1614.5	1.36	0.00000	161.71	0.780810	0.000011
-21.0	0.14000	161.54	499.6	3.08	0.00000	161.94	0.780559	0.000022
-14.0	0.06845	168.11	227.2	1.58	0.00000	162.33	0.780144	0.000039
0.0	0.00055	173.95	275.4	2.96	0.00000	162.68	0.779783	0.000031
14.0	0.04886	170.25	383.2	1.57	0.00000	162.46	0.780011	0.000026
26.0	0.18056	157.52	1190.4	1.72	0.00000	161.71	0.780816	0.000014
30.0	0.23740	157.07	963.5	1.67	0.00000	161.68	0.780845	0.000017
44.0	0.47829	136.75	290.4	2.98	0.00000	160.49	0.782202	0.000025

Fitted Delta D vs Sin^2(Psi) Data			
D Spacing Intercept			0.779762
Slope of Fitted Line			5.175256E-03
Material Stress Constant			1.291300E-07
Residual Stress		51.4 ksi	354.4 MPa
Counting Statistics Stress Error	(+/-)	0.6 ksi	4.1 MPa
Goodness of Fit Stress Error	(+/-)	1.7 ksi	11.9 MPa
Total Stress Error	(+/-)	1.8 ksi	12.5 MPa

Figure 40. Residual stress analysis report table.

The surface effect was seen in the measurement of uniform and notched specimens as shown in the prior Figures 18, 19, and 20 as lower than bulk yielding of the surface. Also, the following Figure 41 displays a similar surface effect. In Figure 41, the solid line is predicted based on Neuber's rule and the stable cyclic stress-strain curve. The measurements were made by x-ray stress analysis and were limited to a surface layer of no more than 0.007 inches as determined by Reference [39]. The effect seemed to manifest itself very near the proportional limit of the metals studied. The metals were 7475-T651 Al and Ti 6Al 4V titanium. The data were collected by tension only quasi-static ramping with static loading during measurements. The surface seemed to yield at a lower stress in reference to the bulk since a residual stress was evident on unloading to zero load in all cases. Also, the surface became perfectly plastic after yielding with slight increases and decreases in stress observed after the break in the loading curves. The bulk responded with nonlinear deformation including strain hardening seen in the load, stroke, and half inch extensometer variations. On unloading, compressive residual stresses were seen as shown in Figure 20. Different yielding behavior was seen as responsible for the effect. From mechanics, x-ray stress analysis should be applicable beyond yielding, if the planner spacing of atoms is assumed to only depend on the hydrostatic component of stress, and the shear components are assumed to only induce distortion with no change in the planar spacing.

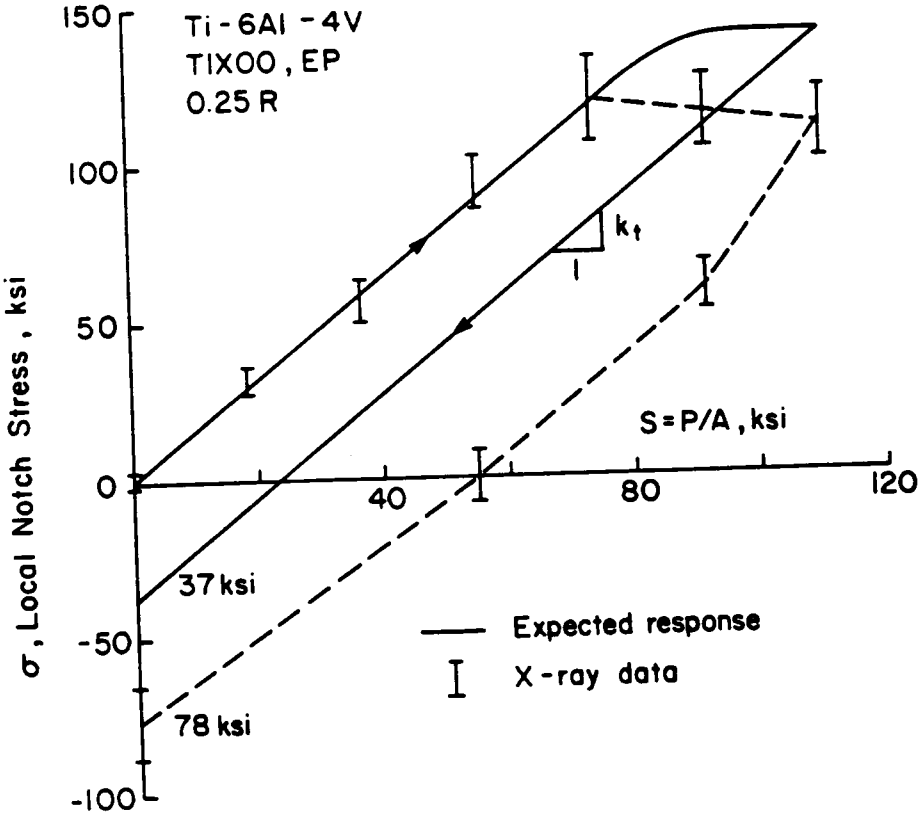


Figure 41. Agreement of x-ray stress with Neuber's estimate for notched Ti-6Al-4V.

This distortion occurs when atomic planes slide past each other mostly due to the movement of dislocations. Further study of the surface response is given by Laurent in Reference [51].

6.5 CYCLIC RESIDUAL STRESS VARIATION

A variation in residual stress was observed during tension only load cycling of a notched 7475-T651 Al specimen. Figure 42 displays the results of cycling the notched aluminum specimen. The loading was nominally elastic for the test. Figure 42 indicates an increase in compressive stress which seemed to approach an asymptote for the data provided.

Two empirical fits were done to smoothly fit the trend in the data of Figure 42. As seen in Figure 42, the two empirical fits varied only slightly over the range plotted. Either empirical fit was satisfactory for recording the trend displayed in Figure 42.

A natural logarithmic and a power curve were used to approximate the cyclic residual stress variation of Figure 42. The form of the power curve and appropriate values to fit Figure 42 are

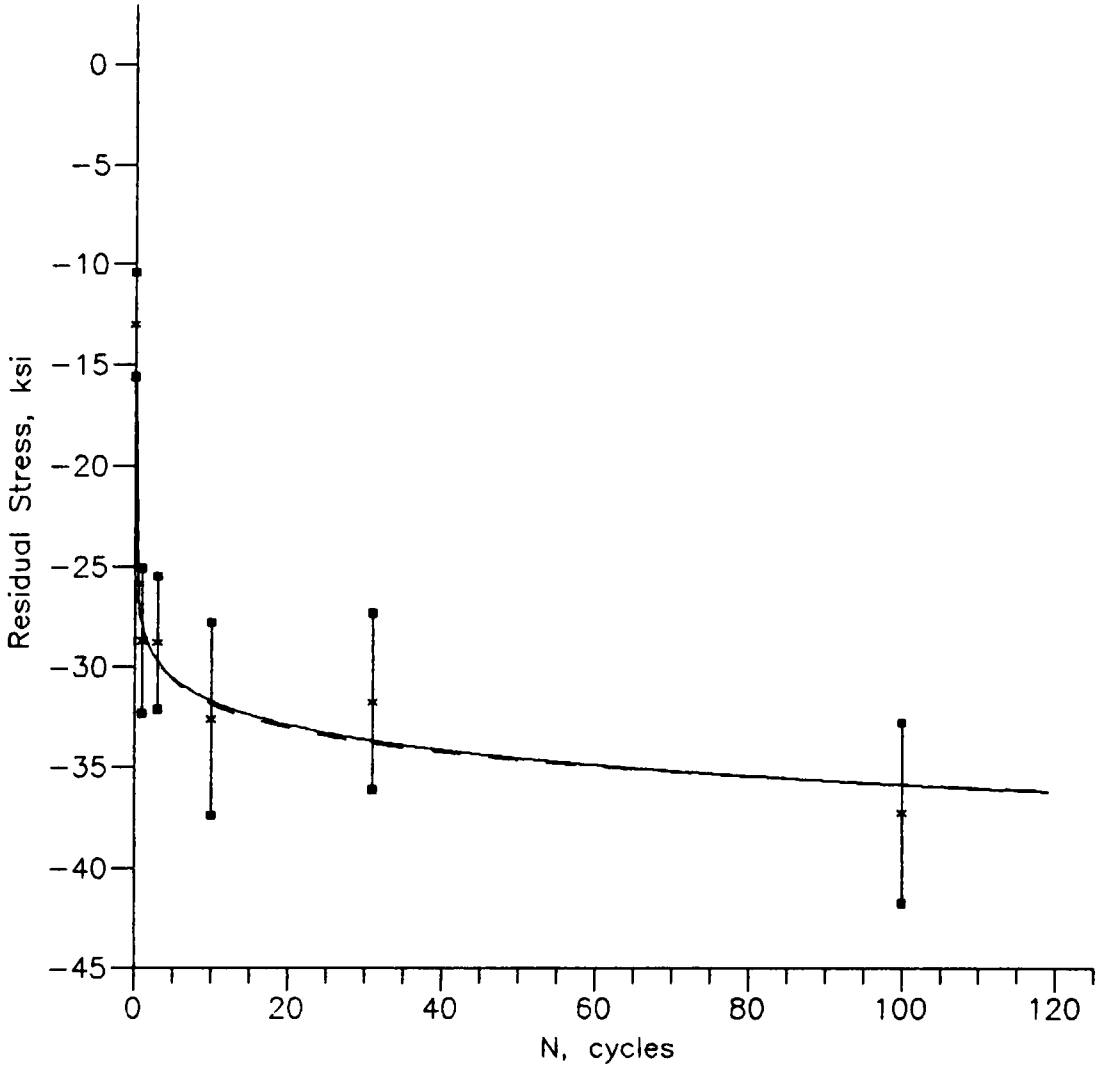
$$\sigma_r = b_r(N + N_r)^{m_r}; \qquad b_r = -28.0088, \quad N_r = 5.82 \times 10^{-7}, \quad m_r = 0.053464 \quad (19)$$

The form of the natural logarithmic curve and the values used to fit Figure 42 are

$$\sigma_r = m_r \cdot ln(N + N_r) + b_r; \quad m_r = -1.730562, \; N_r = 1.89 \times 10^{-4}, \quad b_r = -27.84024 \quad (20)$$

The constants in Equations 19 and 20 were found by varying N_r and computing m_r and b_r by using least squares. First, a value of N_r was selected. Next, for the given value of N_r, m_r and b_r were computed by least squares. Finally, N_r was varied and b_r and m_r were computed until a combination was found that gave a minimum for the sum of the squares of the errors. The resulting curve fits are shown in Figure 42. Both of the fitted curves were found to pass through the data point at $N = 0$, and it may be possible to use the least squares equations for m_r and b_r along with either Equation 19 or 20 evaluated at a point such as $N = 0$ to solve for the constants and produce a satisfactory fit. Finally, it should be mentioned, neither of the two empirical equations has an asymptote for large values of their domains.

Given the residual stress variation, an attempt was made to compute the variation of the maximum stress, σ_{max}, and the mean stress, σ_0, during cycling. To estimate the cyclic response, Neuber's rule was used. In the previous Figure 41, for the Ti-6Al-4V titanium alloy, the complete stress response for the notch root, as determined by x-ray diffraction, was plotted.



Parameter	Value
K_t	2.02
$R = \frac{S_{min}}{S_{max}}$	0
S_{max}	49.5 ksi
$\dot{S}$	229 $\frac{ksi}{s}$

error bar

Figure 42. Variation of residual stress with cycling 7475-T651 Al.

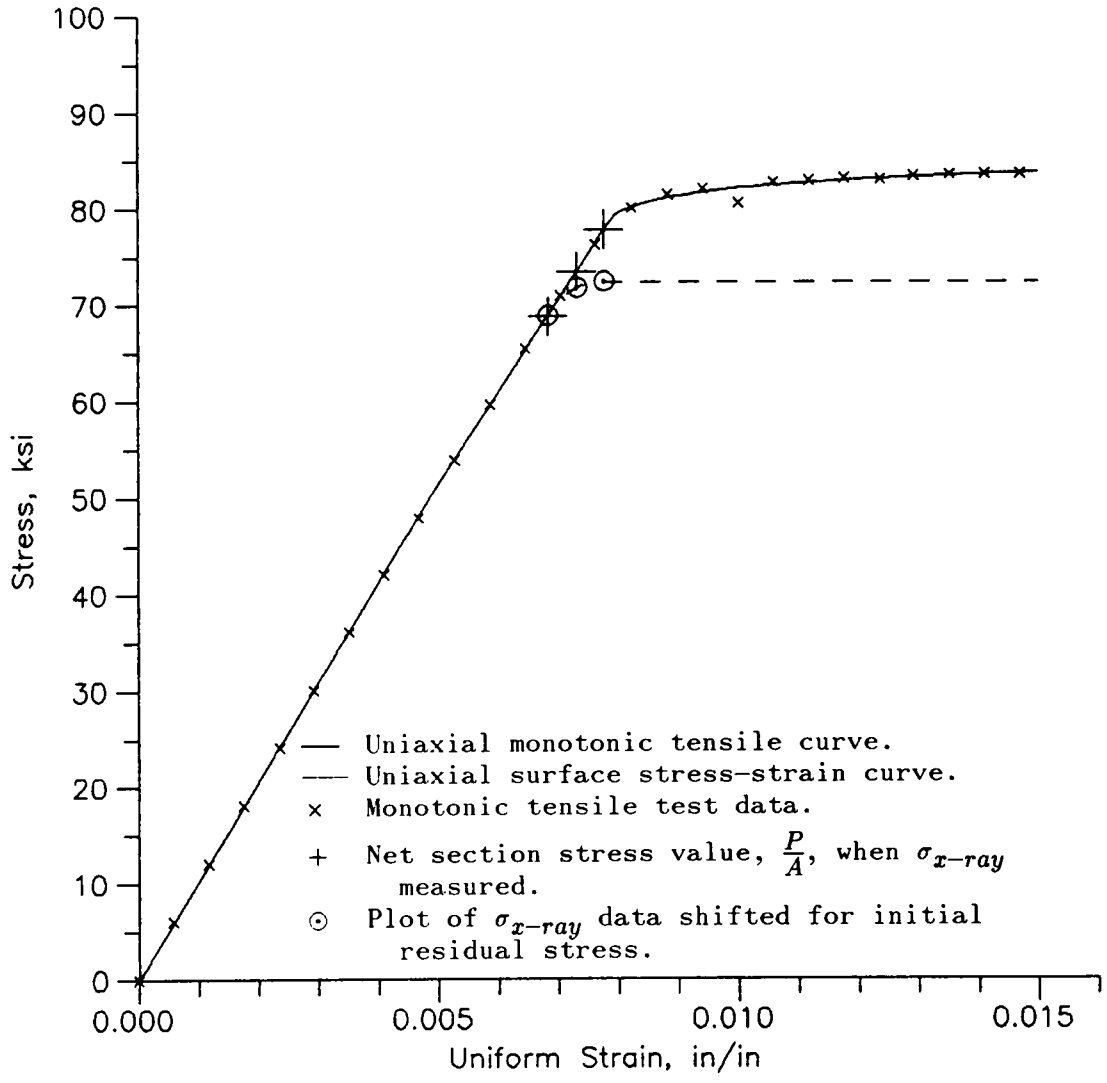
Also, plotted as a solid line, the expected response as determined by Neuber's rule was given: the expected response was formed using the cyclic stress strain curve. The x-ray and Neuber's response were both plotted against $S = \frac{P}{A}$, the net section stress.

The Neuber's rule prediction was based on net section uniaxial tensile stress-strain behavior. It was noticed that a higher maximum, σ_{max}, and residual, σ_r, stress was produced by the Neuber prediction. However, the two responses agree during the elastic loading and have the same slope on elastic unloading. Assuming the surface yielded before the bulk, accounting for early yielding of the surface by lowering the maximum stress computed by Neuber's rule would result in the residual stress computed by Neuber's rule to be nearly equal to the value given by x-ray measurement.

A cyclic stress-strain response for the test conditions of Figure 42 was determined by using Neuber's rule. Based on the results of Figure 41, results were determined using both bulk and surface mechanical response. The surface response included a different yielding behavior. In addition to accounting for the difference in yielding, a proposed surface stress-strain curve is estimated.

Figure 43 is a plot of both the experimental uniaxial mechanical stress-strain response and a proposed surface stress-strain response for the 7475-T651 Al aluminum alloy. The surface stress-strain response was determined by combining the results of uniaxial

mechanical testing with observed x-ray stress measurements made on an unnotched x-ray specimen such as the one shown previously in Figure 37. First, the uniform uniaxial net section stress-strain response was determined. The resulting monotonic tensile curve is given in Figure 43. Next, the variation of x-ray stress with $\frac{P}{A}$, the net section stress, was made on a unnotched x-ray specimen. The prior Figure 20 records the results to be used in determining the surface stress-strain response. The measured x-ray stress values were first corrected by using Equation 15 along with the value of m_{uc} given in Table 7. The assumption was made that the strains were uniform, and the strains at the surface were the same as the net section strains during the measurements of Figure 20. For the linear portion of Figure 20, the monotonic tensile curve and the surface stress-strain curve were assumed to coincide. From the last data point that follows the linear loading trend of Figure 20, the incremental increases in the x-ray stress were plotted to compute the dashed curve of Figure 43. The following procedure was used to determine the dashed curve after linear agreement was not met in previously displayed Figure 20. First, enter the monotonic tensile curve at the correct value of net section stress. At the corresponding strain determined for the net section stress, plot a surface stress value using the incremental changes in x-ray stress. After the last x-ray data point in Figure 43, the dashed curve was extrapolated by assuming that the surface becomes perfectly plastic.



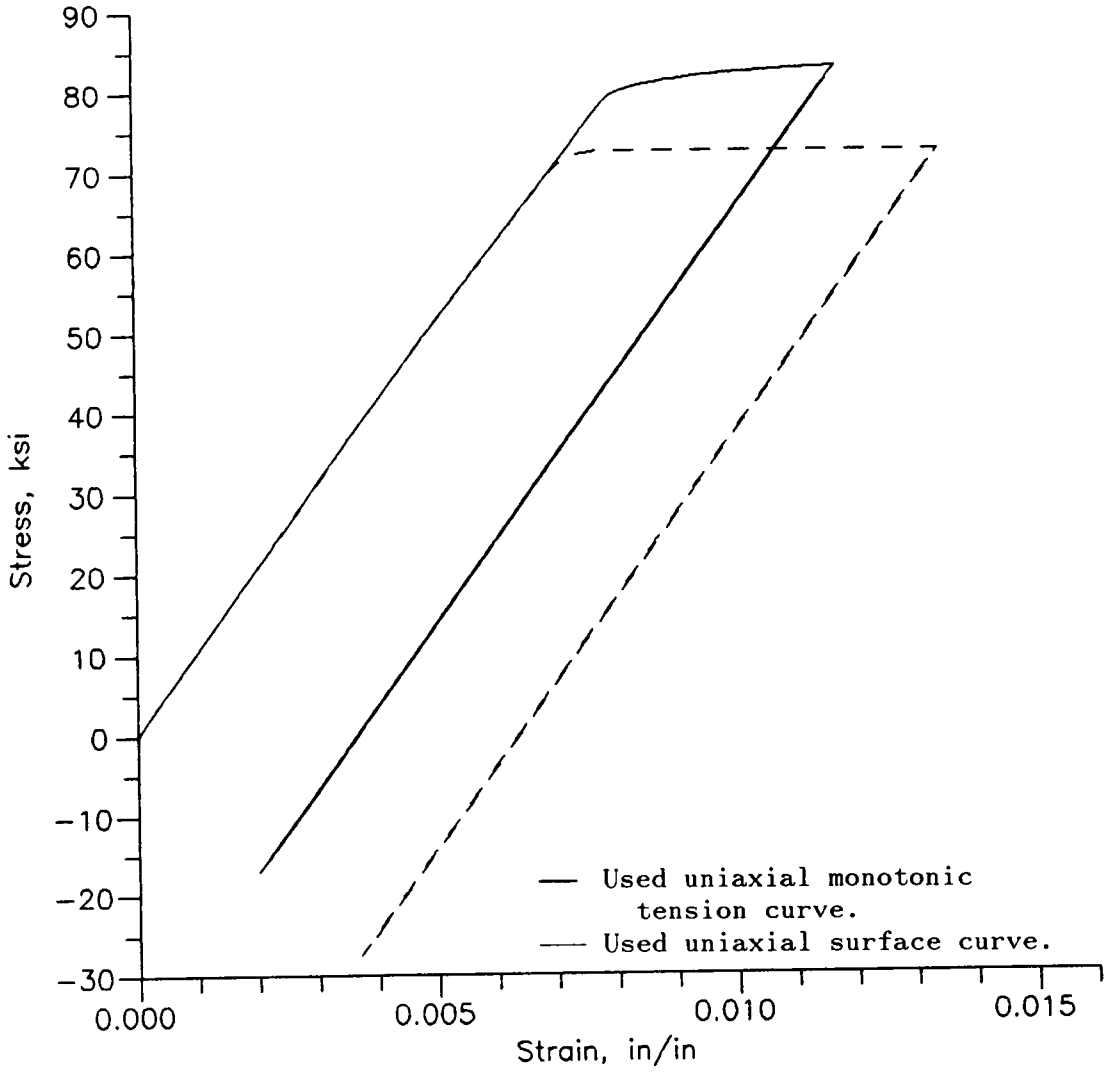
Parameter	Monotonic Curve	Surface Stress Curve
Specimen	A2A08B	AS00
E, ksi	10.3×10^3	10.3×10^3
Waveform	Ramp	Stepped
Control	Strain	Load
Loading Rate	$\dot{\epsilon} = 1 \times 10^{-4} \frac{1}{s}$	manually slow

Figure 43. 7475-T651 Al monotonic tensile curve and proposed surface stress-strain curve.

Figure 44 is a plot of the computed local stress-strain response for the test conditions of the prior Figure 42. The net section loading was elastic, and the elastic form of Neuber's rule, Equation 4, was used as given previously. The monotonic tensile curve in Figure 43 was used to determine the solid line in Figure 44. The dashed line in Figure 44 was determined using the proposed surface stress curve of Figure 43.

The estimated Neuber stress-strain responses differed only in the loading's initial response as shown in Figure 44. The estimated surface response had a lower initial yield and a greater initial strain as expected. However, the cyclic response for both was the same because it was elastic with the same cyclic stress and strain ranges. Also, cyclic hardening was not introduced since the cyclic variations were elastic.

The results of Figure 44 were then used to determine the residual stress after one cycle. The curves of Figure 44 were first shifted vertically downward by the amount of the initial residual stress of −13.0 ksi in Figure 42. Including the initial residual stress, the monotonic response gave a value of −30.0 ksi for the residual stress after one cycle. However, the proposed surface response predicts the residual stress after one cycle to be −40.5 ksi. In this case, the net section stress-strain response seemed to be a better predictor of the initial cyclic behavior.



	Curve	
	Monotonic Tension	Surface Stress
Reversal 1:		
ϵ_a, %	1.17	1.34
σ_a, ksi	82.8	72.3
Cycling:		
$\Delta\epsilon$, %	0.971	0.971
$\Delta\sigma$, ksi	99.8	99.8

Parameter	Value
k_t	2.02
$R = \frac{S_{min}}{S_{max}}$	0
S_{max}	49.5 ksi
E	10.3×10^3 ksi

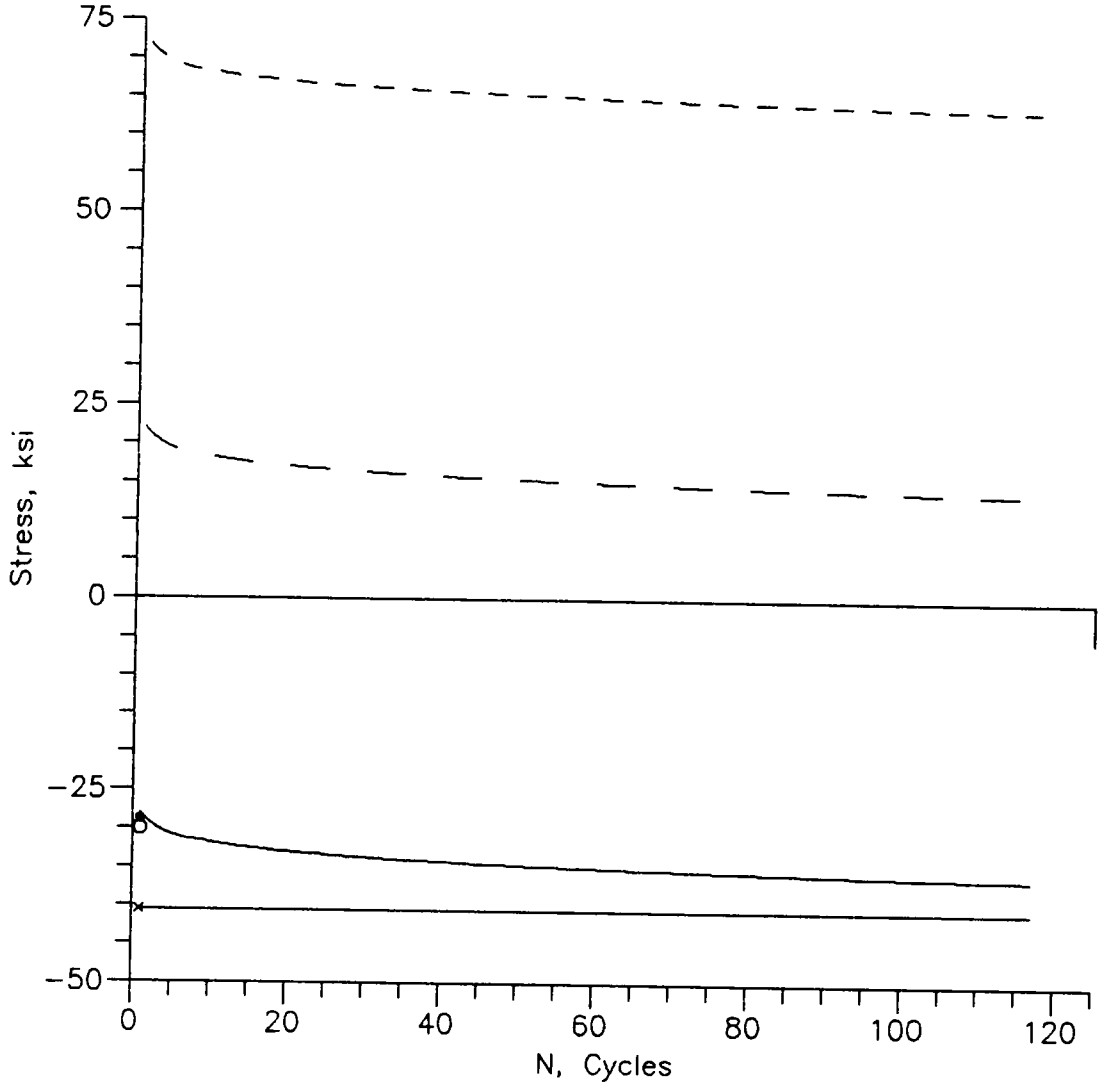
Figure 44. 7475-T651 Al hysteresis loops estimated by Neuber's rule.

On the other hand, if the initial residual stress was not used to shift the results of Figure 44, the surface response could predict the x-ray residual stress after one cycle. The shift due to the initial residual stress will be assumed to be appropriate in the following discussions.

Figure 45 is a plot of the variation of σ_{max} and σ_0 computed by shifting the residual stress variation in Figure 42 by an amount based on the stress range determined in Figure 44. Also, the computed residual stress after one cycle was plotted from Figure 44 shifted by the initial residual stress along with the measured x-ray residual stress after one cycle. The plotted variation of σ_0 was seen to drift downward toward zero stress and level off in Figure 45 based on the computed stress range and the measured x-ray residual stress variation. Also, the residual stress seems to approach asymptotically to the value of the residual stress computed using the surface stress response for one cycle; however, more work must be done to determine if there is a connection between them.

The results of Figure 41 when compared to Figures 44 and 45 were inconsistent with regard to using surface stress variations to predict the resulting residual stress at the notch root. Beyond the use of two different materials, the loading rate differs. In Figure 41, the loading may be considered as quasi-static, because the specimen was manually loaded at a slow rate. Also, there were pauses for a few minutes when x-ray stress measurements were made. In

Figures 44 and 45, the loading rate was rapid, and the only pauses were at zero load when the residual stress was measured. The difference may then be due to the materials used or a time dependent effect when the results in Figure 41 are compared with those in Figures 44 and 45.



---Computed σ_{max}, — X-ray residual stress, $\sigma_r = \sigma_{min}$, – –Computed σ_0

Second Reversal Points:
- • Measured by x-ray
- o Computed with monotonic curve
- × Computed with surface stress curve

Figure 45. Estimated variation of σ_{max} and σ_0 with cycling of 7475-T651 Al.

7.0 CONCLUSIONS

Cycle and time dependent stress-strain transients are found to be prevalent in metals at room temperature. These transients cause local variations in stress and strain amplitude and mean stress during cycling of notched members.

The viscoplastic models include cycle and time dependent stress-strain transients. The recursive integral form of Walker's model was found to be more effective in time step distribution than the differential forms using Euler's integration, Runge-Kutta, and similar methods when large excursions in strain are required based on the reduced number of time steps needed to achieve satisfactory solutions. A combination of Neuber's rule and Walker's model allowed the determination of the local notch stresses and strains while avoiding the use of 2-D finite element computation. The resulting local notch stress-strain response included transient variations in stress and strain amplitude and mean stress for cyclic load histories.

The viscoplastic model material constants were difficult to obtain because they required companion tests at varying strain rates along with nonlinear optimization procedures for their determination.

Cycle and time dependent transients influence fatigue damage by altering strain amplitude and mean stress excluding the effects of creep damage. The use of prevalent strain-life data limits damage determination to strain amplitude and mean stress determination. Fully reversed strain control fatigue test results are valid since the resulting strain amplitude was unaffected by material transients. Once local notch strain amplitudes and mean stresses were determined and included the effects of transients; strain-life data, modified Morrow's parameter, and Miner's rule may be used to determine damage, as defined herein, for variable amplitude load histories.

The results and accomplishments of this investigation are summarized as:

1) Time and cycle dependent mechanical stress-strain behavior from Walker's viscoplasticity constitutive law combined with Neuber's rule modeled the local cyclic variations of mean stress and strain amplitude in a notch and required only a one-dimensional stress-strain analysis.

2) Time and cycle dependent mechanical transients were small for the Ti-6Al-4V and 7475-T651 Al alloys chosen at room temperature from the results of both cyclic rate and static testing.

3) Rainflow cycle counting was done on the local notch strain history to compute mean stress and strain amplitude which would maintain consistency with prevalent strain-life data which are obtained ignoring transients to avoid fatigue damage modeling and testing to incorporate the effects of transients on fatigue damage.

4) Due to the small amounts of time dependent mechanical response observed for the Ti-6Al-4V and 7475-T651 Al alloys used, constants for the viscoplasticity model were not obtained for the alloys at room temperature since the viscoplasticity models do not reduce to the time independent case resulting in nonconvergence of the nonlinear optimization to obtain the constants.

5) Advanced x-ray stress instrumentation allowed the measurement of the stress in a notch under load during brief pauses over a cycle.

6) An anomalous surface behavior was observed and attributed to a lower-than-bulk yield point at the surface which was time dependent as observed with the x-ray stress instrumentation.

7) Texture was found to be extreme for both alloys chosen, but x-ray measurements still followed the linear d-spacing versus $sin^2\psi$ assumption, and satisfactory x-ray stress values were obtained.

8) Finally, x-ray stress data of residual stress variations with cycling in a notched member agreed favorably with the assumption of relaxation of mean stress and discounted the validity of relaxation of residual stress.

8.0 REFERENCES

1. Kurath, Peter, "Extension of the Local Strain Fatigue Analysis Concepts to Incorporate Time Dependent Deformation in Ti-6Al-4v at Room Temperature," T.&A.M. Report No. 464, Department of Theoretical and applied Mechanics, University of Illinois, Urbana-Champaign, Illinois, Feb. 1984.

2. Morrow, JoDean, Ross, A. S., and Sinclair, G. M., "Relaxation of Residual Stresses Due to Fatigue Loading," SAE Transactions, Vol. 68, 1960, pp. 40-48.

3. Brose, W. R., "Fatigue Life Predictions for a Notched Plate with Analysis of Mean Stress and Overstrain Effects," Report No. 402, Department of Theoretical and Applied Mechanics, University of Illinois, Urbana, Illinois, Sept. 1975. See also Fatigue under Complex Loading: Analyses and Experiments, Vol. AE-6, The Society of Automotive Engineers, 1977, pp. 117-135.

4. Wetzel, R. M., "Smooth Specimen Simulation of Fatigue Behavior of Notches," Journal of Materials, ASTM, Vol. 3, No. 3, Sept. 1968, pp. 646-657.

5. Walker, K. P., "Research and Development Program for Nonlinear Structural Modeling With Advanced Time-Temperature Dependent Constitutive Relationships," NASA Report #CR-165533, NASA Lewis Research Center, Cleveland, Ohio, November 25, 1981.

6. Wetzel, R. M., A Method of Fatigue Damage Analysis, Ph.D. Thesis, Department of Civil Engineering, University of Waterloo, Ontario, Canada, 1971. See also Technical Report No. SR 71-107, Scientific Research Staff, Ford Motor Co., Dearborn, Michigan, August 1971.

7. Potter, J. M., "The Effect of Temperature and Load Cycling on the Relaxation of Residual Stresses," Advances in X-ray Analysis, McMurdie, H. F., et. al., eds., Plenum Publishing Corporation, New York, New York, Vol. 20, 1977, pp. 309-319.

8. Wang, Zhong-Ghuang, et. al., "Cyclic Creep Acceleration and Retardation in Cr-Mo-V Rotor Steel at Ambient and Elevated Temperature Respectively," Fatigue and Fracture of Engineering Materials and Structures, Vol. 9, No. 3, 1986, pp. 219-230.

9. Kujawski, D., et. al., "An Experimental Study of Uniaxial Creep, Cyclic Creep, and Relaxation of AISI Type 304 Stainless Steel at Room Temperature," Journal Mechanics and Physics of Solids, Vol. 28, 1980, pp. 129-148.

10. Chan, K. S., et. al., "A Survey of Unified Constitutive Theories," Second Symposium on Nonlinear Constitutive Relations for High Temperature Applications, Cleveland, Ohio, June 1984.

11. Chulya, A. and Walker, K. P., "A New Uniformly Valid Asymptotic Integration Algorithm for Elasto-Plastic-Creep and Unified Viscoplastic Theories Including Continuum Damage," NASA Technical Memorandum 102344 ICOMP-89-22, National Aeronautics and Space Administration, Lewis Research Center, Cleveland, Ohio, December 1989.

12. Freed, A. D. and Walker, K. P., "Refinements in a Viscoplastic Model," NASA Technical Memorandum 102338, National Aeronautics and Space Administration, Lewis Research Center, Cleveland, Ohio, prepared for the winter annual meeting of the American Society of Mechanical Engineers, San Francisco, California, December 10-15, 1989.

13. Walker, K. P., "A Uniformly Valid Asymptotic Integration Algorithm for Unified Viscoplastic Constitutive Models," Advances in Inelastic Analysis, AMD-Vol. 88 and PED-Vol. 28, The American Society of Mechanical Engineers, presented at The Winter Annual Meeting of the American Society of Mechanical Engineers, Boston, Massachusetts, December 13-18, 1987, pp. 13-27.

14. Boyce, W. E. and DiPrima, R. C., Elementary Differential Equations and Boundary Value Problems, Third Edition, John Wiley and Sons, Inc., New York, 1977, pp. 15-16.

15. Neuber, H., "Theory of Stress Concentration for Shear-Strained Prismatical Bodies with Arbitrary Nonlinear Stress-Strain Law," Journal of Applied Mechanics, Vol. 28, No. 4, December 1961, pp. 544-550.

16. Seeger, T. and Heuler, P., "Generalized Application of Neuber's Rule," Journal of Testing and Evaluation, Vol. 8, No. 4, ASTM, July 1980, pp. 199-204.

17. Dowling, N. E., "Fatigue Life Prediction for Complex Load Versus Time Histories," Journal Engineering Material and Testing, ASME, July 1983, pp. 487-498.

18. Dowling, N. E., Brose, W. R., and Wilson, W. K., "Notched Member Fatigue Life Prediction by the Local Strain Approach," Fatigue under Complex Loading Analysis and Experiments, AE-6, Society of Automotive Engineers, 400 Commonwealth Drive, Warrendale, PA, September 1977, pp. 55-84.

19. Wilson, W. K., "Elastic-Plastic Analysis of Blunt Notched CT Specimens and Applications," Journal of Pressure Vessel Technology, Transactions of the ASME, Vol. 96, No. 4, November 1974, pp. 293-298.

20. Dowling, N. E., "A Discussion of Methods for Estimating Fatigue Life," SAE Technical Paper Series 820691, SAE, Warrendale, PA, reprinted from P-109, Proceedings of the SAE Fatigue Conference, Dearborn, Michigan, April 14-16, 1982, pp. 161-174.

21. Dowling, N. E., "A Review of Fatigue Life Prediction Methods," SAE/DOC Symposium on Durability by Design, SAE 1987 Passenger Car Meeting and Exposition, Dearborn, MI, 20-21 October, 1987.

22. Dafalias, Y. F. and Popov, E. P., "Plastic Internal Variables Formalism of Cyclic Plasticity," Journal of Applied Mechanics, Vol. 43, December 1976, pp. 645-651.

23. Dafalias, Y. F. and Popov, E. P., "A Model of Nonlinearly Hardening Materials for Complex Loading," Acta Mechanica, Vol. 21, 1975, pp. 173-192.

24. Dafalias, Y. F., "A Novel Bounding Surface Constitutive Law for the Monotonic and Cyclic Hardening Response of Metals," presented at Structural Mechanics in Reactor Technology, paper L-3/4, 1981, Paris, France.

25. Chaboche, J. L., et. al., "Problems of Describing Ratchetting Effects in Cyclic Plasticity and Viscoplasticity," La Recherche Aerospatiale No. 1, 1989, pp. 63-79.

26. Chaboche, J. L., "A New Constitutive Framework to Describe Limited Ratchetting Effects," Advances in Plasticity 1989, proceedings of Plasticity '89, the Second International Symposium on Plasticity and Its Current Applications, Pergamon Press, New York, pp. 211-214.

27. ASTM, "Standard Practices for Cycle Counting in Fatigue Analysis," 1986 Annual Book of ASTM Standards, Section 3, Metals Test Methods and Analytical Procedures, ASTM, pp. 836-847.

28. Downing, S. S. and Socie, D. F., "Simple Rain-Flow Counting Algorithms," International Journal Fatigue, Vol 4, No. 1, January 1982, pp. 31-40.

29. Rychlik, I., "A New Definition of the Rainflow Cycle Counting Method," International Journal of Fatigue, Vol. 9, No. 2, 1987, pp. 119-121.

30. Dowling, N. E. and Khosrovaneh, A. K., "Simplified Analysis of Helicopter Fatigue Loading Spectra," Development of Fatigue Loading Spectra, Potter, J. M. and Watamabe, R. T., eds., American Society for Testing and Materials, Philadelphia, ASTM STP 1006, 1989, pp. 150-171.

31. Martin, J. F., Topper, T. H., and Sinclair, G. M., "Computer Based Simulation of Cyclic Stress-Strain Behavior with Applications to Fatigue," Materials Research and Standards, ASTM, Vol. 11, No. 2, Feb. 1971, pp. 23-29.

32. Khosrovaneh, A. K. and Dowling, N. E., "Fatigue Loading History Reconstruction Based on the Rain Flow Technique," International Journal of Fatigue, Vol. 12, No. 2, 1990, pp. 99-106.

33. Conle, A. and Landgraf, R. W., "Fatigue Analysis Program for Ground Vehicle Components," Proceedings International Conference on Digital Techniques in Fatigue (SEECO' 83), Society of Environmental Engineers, London, March 1983, pp. 1-28.

34. Socie, D., Shifflet, G., and Berns, H., "A Field Recording System with Applications to Fatigue Analysis," International Journal of Fatigue, Vol. 1, No. 2, April 1979, pp. 103-111.

35. TEC, TEC Model 1600 X-ray Stress Analysis System Operation and Maintenance Manuals, Technology for Energy Corporation, Knoxville, TN, 1985.

36. Noyan, I. C. and Cohen, J. B., Residual Stress Measurement by Diffraction and Interpretation, Springer-Verlag, New York, 1987.

37. Dowling, N. E., Hendricks, R. W., and Ranganathan, K., "X-ray Residual Stress Measurements in Notched Test Specimens," Journal of Testing and Evaluation, JTEVA, Vol. 16, No. 5, September 1988, pp. 456 460.

38. Cullity, B. D., Elements of X-ray Diffraction, Second Edition, Addison-Wesley Publishing Company, Reading, Massachusetts, 1978.

39. Ranganathan, K., "Residual Stress Changes in Fatigue Volume II - A Simulation Model for Stress Measurements in Notched Test Specimens by X-ray Diffraction," Naval Air Development Center, Warminster, PA, Report No. NADC-88141-60, Vol. II, March 1, 1989.

40. Dowling, N. E. and Dunn, D. O., "Residual Stress Changes in Fatigue, Volume I- Residual Stress Measurements by X-ray Diffraction in Notched Test Specimens," Naval Air Development Center, Warminster, PA, Report NADC-88141-60, Vol. I, March 1, 1989.

41. Hirsch, Th., et. al., "Optimizing of the Bending Fatigue Behavior of Ti-Al6-V4 by Shot Peening," 5th International Conference on Fatigue.

42. Taira, Shuji, "X-ray approach for the Study on Mechanical Behavior of Metals," International Conference on Mechanical Behavior of Materials, August 20, 1971.

43. Vohringer, O., et. al., "Relaxation of Shot Peening Induced Residual Stresses of Ti-Al6-V4 by Annealing or Mechanical Treatment," 5th International Conference on Fatigue.

44. Taira, Shuji and Kitagawa, Masaki, "Study of Low Cycle Fatigue Using X-Ray Diffraction, (The effect of residual stress on low cycle fatigue life)," The 5th symposium on x-ray materials strength, The Society of Material Science Japan, Kyoto, Japan, June 1966, pp. 84-87.

45. Landgraf, R. W., "Cyclic Stress-Strain Behavior," Pressure Vessels and Piping: Design Technology -1982- A Decade of Progress, ASME, 1982, pp. 481-485.

46. Freed, A. D., National Aeronautics and Space Administration, Lewis Research Center, Cleveland, Ohio, private communication.

47. Walker, K. P., Pole 65 Log Rd., Smithfield, Road Island, private communication.

48. Khosrovaneh, A. K., et. al., "Fatigue Life Estimates For Helicopter Loading Spectra," NASA, Langley Research Center, Hampton, VA, NASA Contractor Report 181941, December, 1989. See also Journal of the American Helicopter Society, Vol. 35, No. 3, July 1990, pp. 59-67.

49. Walpole, R. E. and Myers, R., H., Probability and Statistics for Engineers and Scientists, Third Edition, Macmillan, New York, 1985, pp. 318–326.

50. Lambda Research Incorporated, "X-Ray Diffraction Pole Figure Determinations on One Ti-6Al-4V and One 7475-T651 Al Sample," Lambda Research Incorporated, 1111 Harrison Avenue, Cincinnati, Ohio. Job No. 0262-0005, December 23, 1987.

51. Laurent, M. P., "A Continuum Surface Layer Effect in Polycrystalline Aggregates," M.S. thesis, Virginia Tech, to be completed Dec. 1991.

APPENDIX A.0 DERIVATIONS FOR COMPUTER ALGORITHMS

A.1.0 ITERATIVE FORM FOR STRAIN CONTROL

Walker, in the appendix of Reference [5], outlined the procedures needed to obtain the equations for numerical solution of his model which was used in this study. The discussion was limited to the strain control form. His procedures were followed and the details are given as follows for the uniaxial case.

The mechanical stress was given by the following equation in Table 2 as

$$\sigma(t) = \Omega(t) + \int_0^t e^{-[q(t) - q(\zeta)]} \left(E\frac{\partial \epsilon}{\partial \zeta} - \frac{\partial \Omega}{\partial \zeta} \right) d\zeta \quad .$$

Let

$$t = t_{n+1}$$

and

$$\sigma(t_{n+1}) = \sigma_{n+1} \quad \Omega(t_{n+1}) = \Omega_{n+1} \quad q(t_{n+1}) = q_{n+1}$$

Substituting the above into the previous equation and rearranging gives:

$$\sigma_{n+1} - \Omega_{n+1} = \int_0^{t_{n+1}} e^{-[q_{n+1} - q(\zeta)]} \left(E\frac{\partial \epsilon}{\partial \zeta} - \frac{\partial \Omega}{\partial \zeta} \right) d\zeta$$

Notice that q_{n+1} is constant so one may factor the appropriate term outside of the integral

$$\sigma_{n+1} - \Omega_{n+1} = e^{-q_{n+1}} \int_0^{t_{n+1}} e^{q(\zeta)} \left(E\frac{\partial \epsilon}{\partial \zeta} - \frac{\partial \Omega}{\partial \zeta} \right) d\zeta \tag{A.1}$$

Breaking the above integral into two integrals over the time intervals $(0, t_n)$ and (t_n, t_{n+1}) gives

$$\sigma_{n+1} - \Omega_{n+1} = e^{-q_{n+1}} \int_{t_n}^{t_{n+1}} e^{q(\zeta)} \left(E\frac{\partial \epsilon}{\partial \zeta} - \frac{\partial \Omega}{\partial \zeta} \right) d\zeta$$

$$+ e^{-q_{n+1}} \int_0^{t_n} e^{q(\zeta)} \left(E\frac{\partial \epsilon}{\partial \zeta} - \frac{\partial \Omega}{\partial \zeta} \right) d\zeta \tag{A.2}$$

Now rewrite equation A.1 for the time interval $(0, t_n)$ as

$$\sigma_n - \Omega_n = e^{-q_n} \int_0^{t_n} e^{q(\zeta)} \left(E\frac{\partial \epsilon}{\partial \zeta} - \frac{\partial \Omega}{\partial \zeta} \right) d\zeta$$

so

$$\int_0^{t_n} e^{q(\zeta)}\left(E\frac{\partial\epsilon}{\partial\zeta}-\frac{\partial\Omega}{\partial\zeta}\right)d\zeta=\frac{\sigma_n-\Omega_n}{e^{-q_n}}$$

Substitution of the above equation into A.2 gives

$$\sigma_{n+1}-\Omega_{n+1}=e^{-q_{n+1}}\int_{t_n}^{t_{n+1}} e^{q(\zeta)}\left(E\frac{\partial\epsilon}{\partial\zeta}-\frac{\partial\Omega}{\partial\zeta}\right)d\zeta+\frac{e^{-q_{n+1}}(\sigma_n-\Omega_n)}{e^{-q_n}}$$

and

$$\sigma_{n+1}-\Omega_{n+1}=e^{-q_{n+1}}\int_{t_n}^{t_{n+1}} e^{q(\zeta)}\left(E\frac{\partial\epsilon}{\partial\zeta}-\frac{\partial\Omega}{\partial\zeta}\right)d\zeta+(\sigma_n-\Omega_n)e^{-(q_{n+1}-q_n)}$$

Now assume that $\left(E\frac{\partial\epsilon}{\partial\zeta}-\frac{\partial\Omega}{\partial\zeta}\right)$ is constant over the time interval. Also, approximate the derivatives by finite differences such as $\delta t=t_{n+1}-t_n$. The above equation becomes

$$\sigma_{n+1}-\Omega_{n+1}=e^{-q_{n+1}}\left(E\frac{\delta\epsilon}{\delta t}-\frac{\delta\Omega}{\delta t}\right)\int_{t_n}^{t_{n+1}} e^{q(\zeta)}\,d\zeta+(\sigma_n-\Omega_n)e^{-\delta q}$$

$$=e^{-q_{n+1}}\left(\frac{E\delta\epsilon-\delta\Omega}{\delta t}\right)\int_{t_n}^{t_{n+1}} e^{q(\zeta)}\,d\zeta+(\sigma_n-\Omega_n)e^{-\delta q}$$

Now change the variable of integration from ζ to $q(\zeta)$ so

$$\sigma_{n+1}-\Omega_{n+1}=e^{-q_{n+1}}\left(\frac{E\delta\epsilon-\delta\Omega}{\delta t}\right)\int_{t_n}^{t_{n+1}} \frac{e^{q(\zeta)}}{\frac{dq(\zeta)}{d\zeta}}\,dq(\zeta)+(\sigma_n-\Omega_n)e^{-\delta q}$$

Find an expression for $\frac{dq(\zeta)}{d\zeta}$ in terms of finite differences and factor

out of the integration under the assumption of it being constant

$$\sigma_{n+1} - \Omega_{n+1} = \frac{e^{-q_{n+1}\left(\frac{E\delta\epsilon - \delta\Omega}{\delta t}\right)}}{\frac{\delta q}{\delta t}} \int_{t_n}^{t_{n+1}} e^{q(\zeta)}\, dq(\zeta) + (\sigma_n - \Omega_n)e^{-\delta q}$$

Integration gives the result

$$\sigma_{n+1} - \Omega_{n+1} = \frac{e^{-q_{n+1}\left(\frac{E\delta\epsilon - \delta\Omega}{\delta t}\right)}\left(e^{q_{n+1}} - e^{q_n}\right)}{\frac{\delta q}{\delta t}} + (\sigma_n - \Omega_n)e^{-\delta q} \quad .$$

By combining exponentials by multiplication, rearranging and simplifying for δt gives the desired result

$$\sigma_{n+1} = \Omega_{n+1} + \frac{\left(E\delta\epsilon - \delta\Omega\right)\left(1 - e^{-\delta q}\right)}{\delta q} + (\sigma_n - \Omega_n)e^{-\delta q} \tag{A.3}$$

when δq was nonzero.

Before yielding it may be seen from Table 2 that $q(t)$ and, therefore, δq may be equal to zero. This was especially possible during numerical calculations due to underflow. Equation A.9 must then be evaluated in the limit when δq approaches zero. The limit is taken as follows:

$$\sigma_{n+1} - \Omega_{n+1} = (E\delta\epsilon - \delta\Omega) \lim_{\delta q \to 0}\left(\frac{1 - e^{-\delta q}}{\delta q}\right) + (\sigma_n - \Omega_n) \lim_{\delta q \to 0}\left(e^{-\delta q}\right)$$

Applying l'Hospital's rule and simplifying gives

$$\sigma_{n+1} = E\delta\epsilon + \sigma_n \tag{A.4}$$

the expected elastic solution.

The plastic strain equation from Table 2 is now to be numerically approximated. It will be seen that the change in the plastic strain over the time interval is needed for substitution into the some of the other equations. Starting with

$$c(t) = \int_0^t \left(\frac{\partial\epsilon}{\partial\zeta} - \frac{1}{E}\frac{\partial\sigma}{\partial\zeta}\right) d\zeta$$

and taking the derivative, one obtains

$$\frac{dc}{dt} = \frac{\partial\epsilon}{\partial t} - \frac{1}{E}\frac{\partial\sigma}{\partial t} \quad .$$

Now approximate the derivatives by finite differences

$$\frac{\delta c}{\delta t} = \frac{\delta\epsilon}{\delta t} - \frac{1}{E}\frac{\delta\sigma}{\delta t} \quad .$$

Simplifying gives

$$\delta c = \delta\epsilon - \frac{1}{E}\delta\sigma \tag{A.5}$$

the desired result.

The equilibrium stress equation is now to be approximated. The derivation follows that of the mechanical stress. The equilibrium stress is given in Table 2 as

$$\Omega(t) = \overset{8}{\Omega} + N_1 c(t) + N_2 \int_0^t e^{-[g(t) - g(\zeta)]} \frac{\partial c}{\partial \zeta} \, d\zeta \quad .$$

Let $t = t_{n+1}$, $\Omega(t) = \Omega(t_{n+1}) = \Omega_{n+1}$, etc., and factor constant terms giving

$$\Omega_{n+1} = \overset{8}{\Omega} + N_1 c_{n+1} + N_2 \int_0^{t_{n+1}} e^{-[g_{n+1} - g(\zeta)]} \frac{\partial c}{\partial \zeta} \, d\zeta$$

$$= \overset{8}{\Omega} + N_1 c_{n+1} + N_2 e^{-g_{n+1}} \int_0^{t_{n+1}} e^{g(\zeta)} \frac{\partial c}{\partial \zeta} \, d\zeta$$

$$= \overset{8}{\Omega} + N_1 c_{n+1} + N_2 e^{-g_{n+1}} \int_{t_n}^{t_{n+1}} e^{g(\zeta)} \frac{\partial c}{\partial \zeta} \, d\zeta + N_2 e^{-g_{n+1}} \int_0^{t_n} e^{g(\zeta)} \frac{\partial c}{\partial \zeta} \, d\zeta \quad .$$

Now, determine the expression for the integral over the time interval $(0, t_n)$ as

$$\Omega_n = \overset{8}{\Omega} + N_1 c_n + N_2 \int_0^{t_n} e^{-[g_n - g(\zeta)]} \frac{\partial c}{\partial \zeta} \, d\zeta$$

$$= \overset{8}{\Omega} + N_1 c_n + N_2 e^{-g_n} \int_0^{t_n} e^{g(\zeta)} \frac{\partial c}{\partial \zeta} \, d\zeta$$

Therefore,

$$\int_0^{t_n} e^{g(\zeta)}\frac{\partial c}{\partial \zeta}\, d\zeta = \frac{\Omega_n - \overset{\circ}{\Omega} - N_1 c_n}{N_2 e^{-g_n}} \ .$$

Next, perform the appropriate substitution

$$\Omega_{n+1} = \overset{\circ}{\Omega} + N_1 c_{n+1} + N_2 e^{-g_{n+1}} \int_{t_n}^{t_{n+1}} e^{g(\zeta)}\frac{\partial c}{\partial \zeta}\, d\zeta + N_2 e^{-g_{n+1}}\left(\frac{\Omega_n - \overset{\circ}{\Omega} - N_1 c_n}{N_2 e^{-g_n}}\right) .$$

Assume appropriate terms are constant for integration, change the variable of integration, approximate derivatives with finite differences, and finally integrate and simplify

$$\Omega_{n+1} = \overset{\circ}{\Omega} + N_1 c_{n+1} + N_2 e^{-g_{n+1}}\left(\frac{\delta c}{\delta t}\right) \int_{t_n}^{t_{n+1}} \frac{e^{g(\zeta)}}{\frac{dg(\zeta)}{d\zeta}}\, dg(\zeta)$$

$$+ N_2 e^{-g_{n+1}}\left(\frac{\Omega_n - \overset{\circ}{\Omega} - N_1 c_n}{N_2 e^{-g_n}}\right)$$

$$= \overset{\circ}{\Omega} + N_1 c_{n+1} + N_2 e^{-g_{n+1}}\left(\frac{\frac{\delta c}{\delta t}}{\frac{\delta g}{\delta t}}\right) \int_{t_n}^{t_{n+1}} e^{g(\zeta)}\, d(\zeta)$$

$$+ N_2 e^{-g_{n+1}}\left(\frac{\Omega_n - \overset{\circ}{\Omega} - N_1 c_n}{N_2 e^{-g_n}}\right)$$

$$= \overset{8}{\Omega} + N_1 c_{n+1} + N_2 e^{-g_n+1}\left(\frac{\delta c}{\delta g}\right)\left(e^{g_n+1} - e^{g_n}\right)$$

$$+ N_2 e^{-g_n+1}\left(\frac{\Omega_n - \overset{8}{\Omega} - N_1 c_n}{N_2 e^{-g_n}}\right)$$

$$= \overset{8}{\Omega} + N_1 c_{n+1} + \frac{N_2 \delta c}{\delta g}\left(1 - e^{-\delta g}\right) + (\Omega_n - \overset{8}{\Omega} - N_1 c_n) e^{-\delta g} \qquad \text{(A.6)}$$

giving the desired result when δg is nonzero.

During elastic loadings, δg may be zero, and the limit of the above equation must be taken as it approaches zero. This results in the following expression:

$$\Omega_{n+1} = N_2 \delta c + \Omega_n \qquad \text{(A.7)}$$

to be used when δg is zero.

The equation for the isotropic hardening does not need approximation. It may be rewritten using the proper nomenclature for finite time steps. Beginning with the equation in Table 2

$$k(t) = K_1 - K_2 e^{-N_7 r(t)}$$

and letting $t = t_{n+1}$ and $k(t) = k(t_{n+1}) = k_{n+1}$ etc., we find

$$k_{n+1} = K_1 - K_2 e^{-N_7 r_{n+1}} \tag{A.8}$$

the desired result.

The expressions $q(t)$ and $g(t)$ in the integrating factors must be developed for the piecewise integration approximation procedure. The approximation for $q(t)$ proceeded as previously described to obtain the change in $q(t)$ over a time interval. Starting from the equation in Table 2

$$q(t) = \int_0^t \frac{E}{k(\zeta)}\left(\frac{\partial r}{\partial \zeta}\right)^{1-\frac{1}{N}} d\zeta$$

we find

$$\frac{dq}{dt} = \frac{E}{k(t)}\left(\frac{\partial r}{\partial t}\right)^{1-\frac{1}{N}} ,$$

$$\frac{\delta q}{\delta t} = \frac{E}{k_{n+1}}\left(\frac{\delta r}{\delta t}\right)^{1-\frac{1}{N}} , \text{ and}$$

$$\delta q = \frac{E\delta t}{k_{n+1}}\left(\frac{\delta r}{\delta t}\right)^{1-\frac{1}{N}} \tag{A.9}$$

.

the desired result. Continuing to the expression for $g(t)$, starting with the expression in Table 2 as

$$g(t) = \int_0^t \left\{\left(N_3 + N_4 e^{-N_5 r(\zeta)}\right)\frac{\partial r}{\partial \zeta} + N_6 \left|\Omega(\zeta)\right|^{M-1}\right\} d\zeta$$

we find

$$\frac{dg}{dt} = \left(N_3 + N_4 e^{-N_5 r(t)}\right)\frac{\partial r}{\partial t} + N_6 \left|\Omega(t)\right|^{M-1} ,$$

$$\frac{\delta g}{\delta t} = \left(N_3 + N_4 e^{-N_5 r_{n+1}}\right)\frac{\delta r}{\delta t} + N_6 \left|\Omega_{n+1}\right|^{M-1} , \text{ and}$$

$$\delta g = \left(N_3 + N_4 e^{-N_5 r_{n+1}}\right)\delta r + N_6 \delta t \left|\Omega_{n+1}\right|^{M-1} \qquad \text{(A.10)}$$

the needed expression.

Since the equations are iteratively solved and δq and δg depended on σ and Ω, respectively, a Newton-Raphson iteration is performed to improve their values as suggested in Reference [5]. This is done when the variable of interest is nonzero. If the δq or δg is zero, the associated Newton-Raphson iteration is skipped. The following equations are needed to evaluate this iteration. Newton's iterative method is as follows: if an x is desired such that $f(x) = 0$ given an initial estimate x_{old} an improved estimate x_{new} is given by

$$x_{new} = x_{old} - \frac{f(x_{old})}{\dfrac{df(x_{old})}{dx}}$$

so that $f(\delta g)_1$, $\dfrac{df(\delta g)_1}{d(\delta g)}$, $f(\delta q)_2$, and $\dfrac{df(\delta q)_2}{d(\delta q)}$ were needed. Starting from equation A.10, the appropriate equations for δg were

$$\delta g = \left(N_3 + N_4 e^{-N_5 r_{n+1}}\right)\delta r + N_6 \delta t \left|\Omega_{n+1}\right|^{M-1}$$

$$f(\delta g)_1 = \left(N_3 + N_4 e^{-N_5 r_{n+1}}\right)\delta r + N_6 \delta t \left|\Omega_{n+1}\right|^{M-1} - \delta g \rightarrow 0 \qquad \text{(A.11)}$$

$$\frac{df(\delta g)_1}{d(\delta g)} = N_6 \delta t (M-1)\left|\Omega_{n+1}\right|^{M-2}\frac{\partial\left|\Omega_{n+1}\right|}{\partial(\delta g)} - 1 \qquad \text{(A.12)}$$

$$\Omega_{n+1} = \mathring{\Omega} + N_1 c_{n+1} + \frac{N_2 \delta c}{\delta g}\left(1 - e^{-\delta g}\right) + (\Omega_n - \mathring{\Omega} - N_1 c_n)e^{-\delta g}$$

$$\frac{\partial \Omega_{n+1}}{\partial(\delta g)} = -\frac{N_2 \delta c}{(\delta g)^2}(1 - e^{-\delta g}) + \frac{N_2 \delta c}{\delta g} e^{-\delta g} - (\Omega_n - \mathring{\Omega} - N_1 c_n)e^{-\delta g} \tag{A.13}$$

$$\frac{\partial |\Omega_{n+1}|}{\partial(\delta g)} = \begin{cases} \dfrac{\partial \Omega_{n+1}}{\partial(\delta g)} & \text{if } \Omega_{n+1} \geq 0 \\ -\dfrac{\partial \Omega_{n+1}}{\partial(\delta g)} & \text{if } \Omega_{n+1} < 0 \end{cases} \tag{A.14}$$

$$\delta g_{new} = \delta g_{old} - \frac{f(\delta g_{old})_1}{\dfrac{df(\delta g_{old})_1}{d(\delta g)}} \tag{A.15}$$

when δg_{old} is nonzero or else these equations are skipped. Starting from equation A.9, the appropriate equations for δq are

$$q_{n+1} = \frac{E\delta t}{k_{n+1}}\left(\frac{\delta r}{\delta t}\right)^{1 - \frac{1}{N}} + q_n$$

$$\delta q = \frac{E\delta t}{k_{n+1}}\left(\frac{\delta r}{\delta t}\right)^{1 - \frac{1}{N}}$$

$$f(\delta q)_2 = \delta q - \frac{E\delta t}{k_{n+1}}\left(\frac{\delta r}{\delta t}\right)^{1 - \frac{1}{N}} \rightarrow 0 \tag{A.16}$$

$$\frac{f(\delta q)_2}{d(\delta q)} = 1 - \frac{E\delta t}{k_{n+1}}\left(1 - \frac{1}{N}\right)\left(\frac{\delta r}{\delta t}\right)^{1 - \frac{1}{N}} \frac{\partial(\delta r)}{\partial(\delta q)} \tag{A.17}$$

$$\frac{\partial(\delta r)}{\partial(\delta q)} = \frac{\partial |\delta c|}{\partial(\delta q)}$$

$$\delta c = \delta \epsilon - \frac{\delta \sigma}{E}$$

$$\frac{\partial(\delta c)}{\partial(\delta q)} = -\frac{1}{E}\frac{\partial(\delta \sigma)}{\partial(\delta q)} = -\frac{1}{E}\frac{\partial(\sigma_{n+1} - \sigma_n)}{\partial(\delta q)} = -\frac{1}{E}\frac{\partial \sigma_{n+1}}{\partial(\delta q)} \tag{A.18}$$

$$\frac{\partial \sigma_{n+1}}{\partial(\delta q)} = \frac{\partial \Omega_{n+1}}{\partial(\delta q)} - \frac{(E\delta \epsilon - \delta \Omega)(1 - e^{-\delta q})}{(\delta q)^2} + \frac{(E\delta \epsilon - \delta \Omega)e^{-\delta q}}{\delta q} - (\sigma_n - \Omega_n)e^{-\delta q}$$

$$\frac{\partial \Omega_{n+1}}{\partial(\delta q)} = 0$$

$$\frac{\partial \sigma_{n+1}}{\partial(\delta q)} = -\frac{(E\delta\epsilon - \delta\Omega)(1-e^{-\delta q})}{(\delta q)^2} + \frac{(E\delta\epsilon - \delta\Omega)e^{-\delta q}}{\delta q} - (\sigma_n - \Omega_n)e^{-\delta q}$$

$$\frac{\partial \sigma_{n+1}}{\partial(\delta q)} = \frac{(E\delta\epsilon - \delta\Omega)}{(\delta q)^2}\left[(1+\delta q)e^{-\delta q} - 1\right] - (\sigma_n - \Omega_n)e^{-\delta q} \tag{A.19}$$

$$\frac{\partial(\delta r)}{\partial(\delta q)} = \begin{cases} \dfrac{\partial(\delta c)}{\partial(\delta q)} & \text{if } \delta c \geq 0 \\ -\dfrac{\partial(\delta c)}{\partial(\delta q)} & \text{if } \delta c < 0 \end{cases} \tag{A.20}$$

$$\delta q_{new} = \delta q_{old} - \frac{f(\delta q_{old})_2}{\dfrac{df(\delta q_{old})_2}{d(\delta q)}} \tag{A.21}$$

when δq_{old} is nonzero or these equations are skipped. The final expression to be determined for the strain control form is the one for the cumulative plastic strain, $r(t)$. Starting from the expression in Table 2 the derivations proceeds along the following lines:

$$r(t) = \int_0^t \left|\frac{\partial c}{\partial \zeta}\right| d\zeta$$

$$\frac{dr}{dt} = \left|\frac{\partial c}{\partial t}\right|$$

$$\frac{\delta r}{\delta t} = \left|\frac{\delta c}{\delta t}\right| = \frac{|\delta c|}{|\delta t|} = \frac{|\delta c|}{\delta t}$$

$$\delta r = |\delta c| \tag{A.22}$$

which are the needed expressions. The equations for the strain control iterations are summarized in the following table. Also, the finite difference expressions are given.

TABLE A.1. Iterative Equations for Strain Control.

$$\sigma_{n+1} = \begin{cases} \Omega_{n+1} + \dfrac{\left(E\delta\epsilon - \delta\Omega\right)\left(1 - e^{-\delta q}\right)}{\delta q} + (\sigma_n - \Omega_n)e^{-\delta q} & \text{if } \delta q \neq 0 \quad \text{(A.3)} \\ E\delta\epsilon + \sigma_n & \text{if } \delta q = 0 \quad \text{(A.4)} \end{cases}$$

$$\delta c = \delta\epsilon - \frac{1}{E}\delta\sigma \quad \text{(A.5)}$$

$$\Omega_{n+1} = \begin{cases} \overset{\circ}{\Omega} + N_1 c_{n+1} + \dfrac{N_2\delta c}{\delta g}\left(1 - e^{-\delta g}\right) + (\Omega_n - \overset{\circ}{\Omega} - N_1 c_n)e^{-\delta g} & \text{if } \delta g \neq 0 \quad \text{(A.6)} \\ N_2\delta c + \Omega_n & \text{if } \delta g = 0 \quad \text{(A.7)} \end{cases}$$

$$k_{n+1} = K_1 - K_2 e^{-N_7 r_{n+1}} \quad \text{(A.8)}$$

$$\delta q = \frac{E\delta t}{k_{n+1}}\left(\frac{\delta r}{\delta t}\right)^{1-\frac{1}{N}} \quad \text{(A.9)}$$

$$\delta g = \left(N_3 + N_4 e^{-N_5 r_{n+1}}\right)\delta r + N_6\delta t\left|\Omega_{n+1}\right|^{M-1} \quad \text{(A.10)}$$

$$f(\delta g)_1 = \left(N_3 + N_4 e^{-N_5 r_{n+1}}\right)\delta r + N_6\delta t\left|\Omega_{n+1}\right|^{M-1} - \delta g \to 0 \quad \text{(A.11)}$$

$$\frac{df(\delta g)_1}{d(\delta g)} = N_6\delta t(M-1)\left|\Omega_{n+1}\right|^{M-2}\frac{\partial\left|\Omega_{n+1}\right|}{\partial(\delta g)} - 1 \quad \text{(A.12)}$$

$$\frac{\partial\Omega_{n+1}}{\partial(\delta g)} = -\frac{N_2\delta c}{(\delta g)^2}(1 - e^{-\delta g}) + \frac{N_2\delta c}{\delta g}e^{-\delta g} - (\Omega_n - \overset{\circ}{\Omega} - N_1 c_n)e^{-\delta g} \quad \text{(A.13)}$$

$$\frac{\partial\left|\Omega_{n+1}\right|}{\partial(\delta g)} = \begin{cases} \dfrac{\partial\Omega_{n+1}}{\partial(\delta g)} & \text{if } \Omega_{n+1} \geq 0 \\ -\dfrac{\partial\Omega_{n+1}}{\partial(\delta g)} & \text{if } \Omega_{n+1} < 0 \end{cases} \quad \text{(A.14)}$$

$$\delta g_{new} = \delta g_{old} - \frac{f(\delta g_{old})_1}{\dfrac{df(\delta g_{old})_1}{d(\delta g)}} \quad \text{(A.15)}$$

$$f(\delta q)_2 = \delta q - \frac{E\delta t}{k_{n+1}}\left(\frac{\delta r}{\delta t}\right)^{1-\frac{1}{N}} \to 0 \quad \text{(A.16)}$$

TABLE A.1 CONTINUED. Iterative Equations for Strain Control.

$$\frac{f(\delta q)_2}{d(\delta q)} = 1 - \frac{E\delta t}{k_{n+1}}\left(1 - \frac{1}{N}\right)\left(\frac{\delta r}{\delta t}\right)^{1-\frac{1}{N}}\frac{\partial(\delta r)}{\partial(\delta q)} \tag{A.17}$$

$$\frac{\partial(\delta c)}{\partial(\delta q)} = -\frac{1}{E}\frac{\partial(\delta\sigma)}{\partial(\delta q)} = -\frac{1}{E}\frac{\partial(\sigma_{n+1}-\sigma_n)}{\partial(\delta q)} = -\frac{1}{E}\frac{\partial\sigma_{n+1}}{\partial(\delta q)} \tag{A.18}$$

$$\frac{\partial\sigma_{n+1}}{\partial(\delta q)} = \frac{(E\delta\epsilon - \delta\Omega)}{(\delta q)^2}\left[(1+\delta q)e^{-\delta q} - 1\right] - (\sigma_n - \Omega_n)e^{-\delta q} \tag{A.19}$$

$$\frac{\partial(\delta r)}{\partial(\delta q)} = \begin{cases} \dfrac{\partial(\delta c)}{\partial(\delta q)} & \text{if } \delta c \geq 0 \\ -\dfrac{\partial(\delta c)}{\partial(\delta q)} & \text{if } \delta c < 0 \end{cases} \tag{A.20}$$

$$\delta q_{new} = \delta q_{old} - \frac{f(\delta q_{old})_2}{\dfrac{df(\delta q_{old})_2}{d(\delta q)}} \tag{A.21}$$

$$\delta r = |\delta c| \tag{A.22}$$

TABLE A.2. Finite Difference Expressions.

$$\delta r = r_{n+1} - r_n$$

$$\delta\Omega = \Omega_{n+1} - \Omega_n$$

$$\delta\sigma = \sigma_{n+1} - \sigma_n$$

$$\delta\epsilon = \epsilon_{n+1} - \epsilon_n$$

$$\delta t = t_{n+1} - t_n$$

$$\delta c = c_{n+1} - c_n$$

$$\delta g = g_{n+1} - g_n$$

$$\delta q = q_{n+1} - q_n$$

TABLE A.3. Variables Incorporated to Improve Efficiency.

Always Constant	$Y_{M1} = M - 1$ $\quad Y_5 = -N_5$ $\quad Y_{N1} = 1 - \frac{1}{N}$ $\quad Y_{K_2} = -K_2$ $Y_{M2} = M - 2$ $\quad Y_7 = -N_7$ $\quad Y_{N2} = -\frac{1}{N}$
Constant For Each Time Step	$z_1 = N_6 \times \delta t$ $\quad z_4 = E \times \delta t$ $\quad z_7 = \delta t \times Y_{N1}$ $z_2 = \Omega_n - (\overset{8}{\Omega} + N_1 \times c_n)$ $\quad z_5 = E \times \delta\epsilon$ $\quad z_8 = \frac{\delta\sigma}{E}$ $z_3 = N_6 \times \delta t \times Y_{M1} = z_1 \times Y_{M1}$ $\quad z_6 = \sigma_n - \Omega_n$
Constant For Several Equations During An Iteration	$x_1 = \left(N_3 + N_4 \times e^{(Y_5 \times r_n + 1)}\right) \times \delta r$ $\quad x_{13} = (x_{12})^{Y_{N2}}$ $x_2 = e^{-\delta g}$ $\quad x_6 = \overset{8}{\Omega} + N_1 \times c_{n+1}$ $\quad x_{14} = \frac{1 - x_9}{\delta q}$ $x_3 = N_2 \times \delta c$ $\quad x_7 = \lvert\Omega_{n+1}\rvert$ $\quad x_{10} = z_5 - \delta\Omega$ $\quad x_{16} = \frac{1 - x_9}{\delta q}$ $x_4 = \frac{x_3 \times (1 - x_2)}{\delta g}$ $\quad x_8 = (x_7)^{Y_{M2}}$ $\quad x_{11} = z_6 \times x_9$ $\quad x_{17} = \sigma_{n+1} - \Omega_{n+1}$ $x_5 = x_2 \times z_2$ $\quad x_9 = e^{-\delta q}$ $\quad x_{12} = \frac{\delta r}{\delta t}$

TABLE A.4. Final Formulation for Strain Control.

$$k_{n+1} = K_1 + Y_{K_2} \times e^{Y_7 \times r_{n+1}}$$

$$\delta g = x_1 + z_1 \left| \Omega_{n+1} \right|^{Y_{M1}}$$

$$f(\delta g)_1 = x_1 + z_1 \times x_7 \times x_8 - \delta g$$

$$\frac{df(\delta g)_1}{d(\delta g)} = \begin{cases} z_3 \times x_8 \left(\dfrac{x_2 \times x_3 - x_4}{\delta g} - x_5 \right) - 1 & \text{if } \Omega_{n+1} \geq 0 \\ z_3 \times x_8 \left(x_5 + \dfrac{x_4 - x_2 \times x_3}{\delta g} \right) - 1 & \text{if } \Omega_{n+1} < 0 \end{cases}$$

$$\Omega_{n+1} = \begin{cases} x_4 + x_5 + x_6 \\ \qquad or \\ x_2 \times z_2 + x_6 + \dfrac{x_3(1 - x_2)}{\delta g} \end{cases} \quad \text{if } \delta g \neq 0$$

$$\Omega_{n+1} = x_3 + \Omega_n \quad \text{if } \delta g = 0$$

$$\delta g_{new} = \delta g_{old} - \frac{f(\delta g_{old})_1}{\dfrac{df(\delta g_{old})_1}{d(\delta g)}} \quad \text{if } \delta g \neq 0$$

TABLE A.4 CONTINUED. Final Formulation for Strain Control.

$$\delta q = \frac{z_4\left(\frac{\delta r}{\delta t}\right)^Y N1}{k_{n+1}}$$

$$\left.\begin{aligned}
\sigma_{n+1} &= \Omega_{n+1} + x_{11} + \frac{x_{10}(1-x_9)}{\delta q} \\
\delta c &= \delta\epsilon - \frac{\delta\sigma}{E} \\
\delta r &= |\delta c| \\
f(\delta q)_2 &= \delta q - \frac{z_4 \times x_{12} \times x_{13}}{k_{n+1}} \\
\frac{df(\delta q)_2}{d(\delta q)} &= \begin{cases} 1 + \frac{x_{13} \times z_7}{k_{n+1}}\left(\frac{x_{10}((\delta q+1)x_9 - 1)}{(\delta q)^2} - x_{11}\right) & \text{if } \delta c \geq 0 \\ 1 + \frac{x_{13} \times z_7}{k_{n+1}}\left(x_{11} + \frac{x_{10}(1-(\delta q+1)x_9)}{(\delta q)^2}\right) & \text{if } \delta c < 0 \end{cases} \\
\delta q_{new} &= \delta q_{old} - \frac{f(\delta q_{old})_2}{\frac{df(\delta q_{old})_2}{d(\delta q)}} \\
\sigma_{n+1} &= \Omega_{n+1} + \frac{x_{10}(1-x_9)}{\delta q} + z_6 \times x_9
\end{aligned}\right\} \text{if } \delta q \neq 0$$

$$\left.\begin{aligned}
\delta\sigma &= z_5 \\
\delta c &= \delta r = 0
\end{aligned}\right\} \text{if } \delta q = 0$$

Some of the quantities common to several of the above equations may be grouped together and calculated as intermediate values. This was done to improve computational efficiency. The above equations are combined as needed to accomplish this. New variables are defined to eliminate redundant calculations. These variables and the reformulated iterative equations are given in the following tables.

A.2.0 ITERATIVE FORM FOR STRESS CONTROL

The derivations for obtaining the iterative equations for stress control are outlined in the following text. Similar procedures as those used in section A.1 of Appendix A for the strain control case are used. A detailed discussion is not given for the stress control equations, and only the equation derivations are given. The equations again are for the uniaxial case.

Since the variation in stress is assumed to be known, Equation A.3 is rewritten. This is necessary to compute the strain increment. Beginning with Equation A.3 and solving for $\delta\epsilon$ we find

$$\delta\epsilon = \frac{1}{E}\left\{\frac{\delta q\left(\sigma_{n+1} - \Omega_{n+1} - (\sigma_n - \Omega_n)e^{-\delta q}\right)}{1 - e^{-\delta q}} + \delta\Omega\right\}$$

if δq is nonzero. If δq is equal to zero, the following results

$$\delta\epsilon = \frac{\delta\sigma}{E}$$

for the elastic case. Incorporating variables for computational efficiency gives

$$\delta\epsilon = \frac{1}{E}\left\{\frac{x_{17} - x_{11}}{x_{16}} + \delta\Omega\right\} \tag{A.23}$$

or

$$\delta\epsilon = \frac{1}{E}\left\{\frac{\delta q(x_{17} - z_6 \times x_9)}{1 - x_9} + \delta\Omega\right\} \tag{A.24}$$

if δq is nonzero, and

$$\delta\epsilon = z_8 \tag{A.25}$$

if δq is zero. Listed in the following table, expressions are given that are slightly different in arrangement from those in Table A.4 but compute equivalent values.

In summary, the stress control calculations started with the suitable modification of Table A.4. First, ignore all equations that involve calculating σ_{n+1} and $\delta\sigma$ since they are known and constant for the stress control case. Next, include equations A.23, A.24, and A.25 as needed to iterate for the unknown strain quantities. Finally, replace the expressions in Table A.4 with those in Table A.5 as appropriate.

TABLE A.5. Equations Modified for Stress Control.

$$\delta\Omega = x_3 \quad \text{if } \delta g = 0$$

$$\delta c = \delta\epsilon - z_8$$

$$\frac{df(\delta q)_2}{d(\delta q)} = \left\{ \begin{array}{ll} 1 - \dfrac{z_7 \times x_{13}}{k_{n+1}} \left\{ \dfrac{(E \times \delta\epsilon - \delta\Omega)(x_{16} - x_9)}{\delta q} + x_{11} \right\} & \text{if } \delta c \geq 0 \\ 1 + \dfrac{z_7 \times x_{13}}{k_{n+1}} \left\{ \dfrac{(E \times \delta\epsilon - \delta\Omega)(x_{16} - x_9)}{\delta q} + x_{11} \right\} & \text{if } \delta c < 0 \end{array} \right\} \quad \text{if } \delta q \neq 0$$

A.3.0 ITERATIVE FORM FOR NEUBER'S RULE

For the previous cases of stress or strain control, the time variation of either the stress or strain is known, and the unknown strain or stress variation is to be determined. For the notch analysis, the time variation of the stress and strain is unknown, but the time variation of either the nominal load or the net section stress is known. Neuber's rule is then used to relate the time variation of the net section stress to the time variation of the local notch stresses and strains. In this case, the values of both the stress and strain must be iteratively determined at each time step as the net section stress varies. Along with the equations in Table A.4 for strain control we must now include a method for iteratively modifying the strain. This is done by iteratively correcting the strain as determined by Neuber's rule. The correction to the strain is made by a Newton-Raphson approach using the subincrement Neuber's rule. This is necessary since if the strain is determined from the subincrement Neuber's rule directly, it is iteratively unstable. This is true since Neuber's rule involved terms with the product of stress and strain. The current value of the stress could be larger or smaller than the correct value of stress, and would be input into the subincrement Neuber's rule. This would result in a corresponding smaller or larger value of the strain, respectively, than the correct value at this iterative stage. This stain value would then be used to compute an updated stress value which would now be smaller or larger,

respectively, if the original stress at the previous stage is larger or smaller. The values of stress and strain would switch from being too small to too large, accordingly, due to the fact that Neuber's rule involves the product of the stresses and strains in its terms. Therefore, a Newton-Raphson approach using the subincrement Neuber's rule is used to overcome this iterative difficulty. The details of the Newton-Raphson equation using the subincrement Neuber's rule are described in the following. This equation is included with those in Table A.4 to iteratively determine the uniaxial stress and strain time variation for a notched member given the time variation of the net section stress.

We are interested in iteratively correcting the strain using the Newton-Raphson technique. This is described symbolically as

$$\delta\epsilon_{new} = \delta\epsilon_{old} - \frac{f(\delta\epsilon_{old})}{\frac{df(\delta\epsilon_{old})}{d(\delta\epsilon)}} \tag{A.26}$$

The functional forms of $f(\delta\epsilon)$ and its derivative are determined from the subincrement Neuber's rule. The subincrement Neuber's rule is

$$c = (\delta\sigma + \delta\sigma')(\delta\epsilon + \delta\epsilon') = \frac{\left(k_f(\delta S + \delta S')\right)^2}{E}$$

giving

$$f(\delta\epsilon) = (\delta\sigma + \delta\sigma')(\delta\epsilon + \delta\epsilon') - c \rightarrow 0 \quad . \tag{A.27}$$

Now, determine the derivative,

$$\frac{df(\delta\epsilon)}{d(\delta\epsilon)} = \frac{\partial f}{\partial(\delta\sigma)}\frac{d(\delta\sigma)}{d(\delta\epsilon)} + \frac{\partial f}{\partial(\delta\epsilon)}\frac{d(\delta\epsilon)}{d(\delta\epsilon)}$$

$$= \frac{\partial f}{\partial(\delta\sigma)}\frac{d(\delta\sigma)}{d(\delta\epsilon)} + \frac{\partial f}{\partial(\delta\epsilon)} \tag{A.28}$$

where

$$\frac{\partial f}{\partial(\delta\sigma)} = (\delta\epsilon + \delta\epsilon') \tag{A.29}$$

$$\frac{\partial f}{\partial(\delta\epsilon)} = (\delta\sigma + \delta\sigma') \ . \tag{A.30}$$

Now determine $d(\delta\sigma)/d(\delta\epsilon)$. We must consider two cases depending on whether δq is zero or nonzero. For the case where δq is nonzero, we start with equation A.3 of Table A.1,

$$\sigma_{n+1} = \Omega_{n+1} + \frac{\left(E\delta\epsilon - \delta\Omega\right)\left(1 - e^{-\delta q}\right)}{\delta q} + (\sigma_n - \Omega_n)e^{-\delta q}$$

continuing

$$\delta\sigma = \frac{\left(E\delta\epsilon - \delta\Omega\right)\left(1 - e^{-\delta q}\right)}{\delta q} + (\sigma_n - \Omega_n)e^{-\delta q} + \Omega_{n+1} - \sigma_n$$

$$= \left(E\delta\epsilon - \Omega_{n+1} + \Omega_n\right)\frac{\left(1 - e^{-\delta q}\right)}{\delta q} + (\sigma_n - \Omega_n)e^{-\delta q} + \Omega_{n+1} - \sigma_n$$

therefore,

$$\frac{d(\delta\sigma)}{d(\delta\epsilon)} = \frac{\partial(\delta\sigma)}{\partial(\delta\epsilon)} + \frac{\partial(\delta\sigma)}{\partial(\delta q)}\frac{d(\delta q)}{d(\delta\epsilon)} + \frac{\partial(\delta\sigma)}{\partial(\Omega_{n+1})}\frac{d(\Omega_{n+1})}{d(\delta\epsilon)}$$

and

$$\frac{d(\Omega_{n+1})}{d(\delta\epsilon)} = 0$$

$$\frac{d(\delta q)}{d(\delta\epsilon)} = 0$$

$$\frac{\partial(\delta\sigma)}{\partial(\delta\epsilon)} = E\frac{\left(1 - e^{-\delta q}\right)}{\delta q}$$

so

$$\frac{d(\delta\sigma)}{d(\delta\epsilon)} = E\frac{\left(1 - e^{-\delta q}\right)}{\delta q} \quad \text{if } \delta q \neq 0$$

the desired equation. For the case where δq is zero we start with equation A.4 of Table A.1,

$$\sigma_{n+1} = E\delta\epsilon + \sigma_n$$

continuing

$$\delta\sigma = E\delta\epsilon$$

so

$$\frac{d(\delta\sigma)}{d(\delta\epsilon)} = E \quad \text{if } \delta q = 0$$

for the other case. Summarizing the results immediately above gives

$$\frac{d(\delta\sigma)}{d(\delta\epsilon)} = \begin{cases} E\dfrac{\left(1 - e^{-\delta q}\right)}{\delta q} & \text{if } \delta q \neq 0 \\ E & \text{if } \delta q = 0 \end{cases} \tag{A.31}$$

Combining equations A.28, A.29, A.30, and A.31 gives

$$\frac{df(\delta\epsilon)}{d(\delta\epsilon)} = \begin{cases} (\delta\epsilon + \delta\epsilon')\,E\dfrac{\left(1 - e^{-\delta q}\right)}{\delta q} + (\delta\sigma + \delta\sigma') & \text{if } \delta q \neq 0 \\ (\delta\epsilon + \delta\epsilon')\,E + (\delta\sigma + \delta\sigma') & \text{if } \delta q = 0 \end{cases}$$

Substitution of the above into Equation A.26 along with equation A.27 results in

$$\delta\epsilon_{new} = \begin{cases} \delta\epsilon_{old} - \dfrac{(\delta\sigma + \delta\sigma')(\delta\epsilon_{old} + \delta\epsilon') - c}{\dfrac{E(\delta\epsilon_{old} + \delta\epsilon')\left(1 - e^{-\delta q}\right)}{\delta q} + (\delta\sigma + \delta\sigma')} & \text{if } \delta q \neq 0 \\ \delta\epsilon_{old} - \dfrac{(\delta\sigma + \delta\sigma')(\delta\epsilon_{old} + \delta\epsilon') - c}{E(\delta\epsilon_{old} + \delta\epsilon') + (\delta\sigma + \delta\sigma')} & \text{if } \delta q = 0 \end{cases}$$

and including terms to improve computational efficiency and simplifying the subtraction gives

$$\delta\epsilon_{new} = \begin{cases} \delta\epsilon_{old} + \dfrac{c - (\delta\sigma + \delta\sigma')(\delta\epsilon_{old} + \delta\epsilon')}{E(\delta\epsilon_{old} + \delta\epsilon')x_{14} + (\delta\sigma + \delta\sigma')} & \text{if } \delta q \neq 0 \\ \delta\epsilon_{old} + \dfrac{c - (\delta\sigma + \delta\sigma')(\delta\epsilon_{old} + \delta\epsilon')}{E(\delta\epsilon_{old} + \delta\epsilon') + (\delta\sigma + \delta\sigma')} & \text{if } \delta q = 0 \end{cases} \qquad \text{(A.32)}$$

In summary, Equation A.32 is used in combination with those in Table A.4 to perform the needed iterative calculations for Neuber control. Also, logic is included to give the correct sign for the stress and strain values. This is needed since Neuber's rule involves terms that are the products of the stress and the strain. There are, therefore, two possible solutions corresponding to loading or unloading analogous graphically to results in the first and third quadrants for a coordinate system centered at the reversal point. Once logic is included to obtain the correct result for loading or unloading,

Equation A.32 and those in Table A.4 can be used to perform the iterative calculations needed to approximate the uniaxial stress-strain response for a notched member.

APPENDIX B.0 FATIGUE HISTORY PROGRAM

B.1.0 INTRODUCTION

The program MFATHS generated the stress-strain response for a material given its net section stress or strain history. The computer code for MFATHS was written using the FORTRAN77 language ANSI standard. The program will simulate three uniaxial load cases. The cases were strain, stress, and Neuber control. Neuber control involved predicting local notch response for a stress raiser given, k_t, the stress concentration factor. All floating point values were computed with double precision arithmetic. Two input files were needed. They were "INPUT" and a file that contains the load history. "INPUT" contained information on the material, type simulation, convergence parameters, and loading rate to list a few. The file that contained the load history contained the peak and valley history. The load history does not have to be normalized as the program will normalize the history to any maximum excursion level from zero that was desired. In the area of convergence parameters, default values were assigned within the main program by PARAMETER, DATA, and assignment statements.

The default values may be permanently changed in the main program. Also, the program default values may be redefined during program execution. In the main program, the PARAMETER real and integer numeric constants MINIT1, MAXIT2, and PERERR along with FSNINC, FSSINC, FNEINC, CSNINC, CSSINC, CNEINC, IFNINC, ICNINC specify the control of the convergence and integration. The above values may be redefined when the program was executed.

The PARAMETER integer constants NUMLET, INNUM, IOTNUM, INHNU, IOTHNU, IOTGNU, IRORNU and the PARAMETER character constants INNAM, OUTNAM were assigned default values in the main program. These values may only be changed in the source code of the main program. The following list in Table B.1 describes the variables.

The program was controlled through the input file. The input file contained the following information as listed in Table B.2, and the input is given line by line.

TABLE B.1. Main Program PARAMETER Variables.

PARAMETER numeric constants	
MINIT1	-minimum number of iterations during integration subroutine call
MAXIT2	-maximum number of iterations during integration subroutine call
PERERR	-percent error specified for convergence
FSNINC	-fine strain increment to be used during integration
FSSINC	-fine stress increment to be used during integration
FNEINC	-fine Neuber increment to be used during integration
CSNINC	-coarse strain increment to be used during integration
CSSINC	-coarse stress increment to be used during integration
CNEINC	-coarse Neuber increment to be used during integration
IFNINC	-fine number of increments to be used during integration
ICNINC	-coarse number of increments to be used during integration
NUMLET	-maximum number of letters in a file name
INNUM	-unit number of input data file
IOTNUM	-unit number of output message file giving verification of input and error messages
INHNU	-unit number of input load history file
IOTHNU	-unit number of output local stress-strain history reversal points file
IOTGNU	-unit number of outputted complete local stress-strain history which may be used for later plotting
IRORNU	-unit number of reordered peak-valley load history at the maximum peak, minimum valley, or the largest excursion from zero
PARAMETER character constants	
INNAM	-name of input data file
OUTNAM	-name of output message file

TABLE B.2. Input for File "INPUT".

1. Problem Heading -user input description
2. Output File -name of file, enclosed in single quotes, to contain the computed stress-strain reversal points
3. Type of control -one of the following values 1, 2, or 3
   1 =strain control, do not include line 3a
   2 =stress control, do not include line 3a
   3 =Neuber control, also include line 3a

3a. Stress concentration factor k_t -included only if 3 for Neuber control is inputted on line 3

4. Specimen/Material Name -user supplied description enclosed in single quotes
5. Unit System -one of the two values 1 or 2
   1 =kip, ksi, in/in, seconds
   2 =MN, MPa, in/in, seconds
6. Modulus of Elasticity, E
7. Walker's Model Constants -$\overset{8}{\Omega}$, N, M
8. Walker's Model Constants -N_1, N_2, N_3
9. Walker's Model Constants -N_4, N_5, N_6
10. Walker's Model Constants -N_7, K_1, K_2
11. Title of History -user specified description enclosed in single quotes
12. Filename of Input Peak/Valley Load history enclosed in single quotes
13. Magnitude to scale largest excursion from zero for a corresponding peak or valley in the load history
14. Automated scan of history for magnitude of original maximum excursion from zero -one of two values 0 or 1
   0 = NO, do not scan, and include a line 14a with the value
   1 = YES, scan, and do not include a line 14a

14a. User supplied maximum excursion from zero -include this line only if 0 is supplied on line 14

15. Loading Rate
16. Create data file for later plotting of hysteresis loops -one of two values 0 or 1
   0 = NO, do not create the file, and do not include line 16a
   1 = YES, create the file, and include line 16a

16a. Filename for plotting data enclosed in single quotes - this line is not included if a 0 is given on line 16

TABLE B.2 CONTINUED. Input for File "INPUT".

17. Selection of convergence parameters -one of the values 0, 1, 2, 10, 11, or 12 for the operations tabulated below

	Constant Number of Increments	Constant Step Length
User Specified	0	10
Default Fine	1	11
Default coarse	2	12

If 0 used, include line 17a
If 10 used, include line 17b
Else, do not put a line 17a or 17b in the input file

17a. If line 17 is 0 then include the following 3 lines
- 17a1. MINIT, MAXIT -minimum and maximum number of iterations
- 17a2. NUMINC, PERERR -number of increments(intervals) between reversals, and the percent error indicating the maximum amount of change in prescribed values allowed before convergence is assumed
- 17a3. STNINC, STSINC, ANEINC -increments for strain, stress, and Neuber calculations to be used during integrating between reversals: The Neuber increment is to be the increment in the net section stress

17b. if line 17 is 10 then include the following 3 lines
- 17b1. MINIT, MAXIT -same as 17a1
- 17b2. PERERR -same as PERERR in line 17a2
- 17b3. STNINC, STSINC, ANEINC -same as 17a3

18. Reorder input load history -one of the three values 0, 1, 2
- 0 =do not rearrange
- 1 =rearrange starting with lowest valley
- 2 =rearrange starting with highest peak

B.2.0 SAMPLE RUNS

B.2.1 Introduction

The program computed the local stress-strain history for uniaxial loading. The program accessed two input files, and may create up to three output files. There was no interactive input/output with the program. All information to/from the program was through files.

The two input files were both required for the program to execute. The first was named "INPUT" and contains the parameters that control the programs execution described above. The second was the file that contained the normalized stress, or strain peak/valley sequence loading history. The name of this file was assigned in the file "INPUT". If either of these files was missing, an error condition was reported in a file named "OUTPUT".

Of the three output files, two were always produced by the program. The third output file was produced as an option by the user. The first default output file was named "OUTPUT". The input parameters, from file "INPUT", were echoed to this file. Also, any error messages appeared in this file. The second default file contained the reversals for the complete local stress-strain peak/valley history in sequence. The second default output file was assigned its name in the file "INPUT" during execution by the user. Failure to supply a name resulted in an error condition which was

reported in the file "OUTPUT". Also, in the second output file, a summary of the computational progress of the program from reversal to reversal was given. Also, if the numerical routines do not converge, an error code was included at the current stress-strain value where the error occurred as the last reported values in this file. The results of the second output file were useful for cycle counting and fatigue computations. The third output file was optional and included the values of each stress-strain value computed in sequence. This file's information was useful for plotting the complete local stress-strain history.

B.2.2 Example Problems and Input Files

Three example problems were given. The first was a stress control simulation. The other two were one each of a strain and a Neuber control simulation. Each example contained 10 full cycles plus two half cycles. The first half cycle was to the first reversal, and the last half cycle was to the final reversal. The loading rates in the three examples were picked so that the stress and strain rates during elastic loading were nearly identical. Also, the input load history reversal points were picked to produce similar values for the initial reversal stress/strain level and subsequent cyclic stress/strain ranges where possible. The input files for the three problems are given below.

```
'Stress control example.' Item1:Problem heading
'stsr03'                  Item2:Local peak/valley file out
2                         Item3:Stress control
'Walker constants in ksi' Item4:Material identification
1                         Item5:Unit system,kip,ksi,in/in,s
25920.3                   Item6:Elastic modulus, ksi
0.0      16.95    1.16    Item7:Omega0, ksi; N; M
0.0      300000.0 8000.0  Item8:N1; N2, ksi; N3
0.0      1.0e-14  0.0     Item9:N4, N5, N6
1.0e-14 50.931    0.0     Item10:N7; K1, ksi; K2
'R=0.183 stress history'  Item11:History title
'stsh03'                  Item12:peak/valley history in
75.8                      Item13:Max excursion from 0, ksi
1                         Item14:Find input max excursion
207                       Item15:loading rate, ksi/s
1                         Item16:All of stress-strain out
'stsg03'                  Item16A:File complete history out
11                        Item17:Default fine step length
0                         Item18:do not reorder history in.
Comments may be placed after the last required entry on a
line, or below the last required line.
These areas are not read by the program.
```

Figure B.1. Stress control: parameter input file "INPUT".

```
1000
-183
1000
-183
1000
-183
1000
-183
1000
-183
1000
-183
1000
-183
1000
-183
1000
-183
1000
-183
1000
-183
```

Figure B.2. Stress control: history input file 'STSH03'.

```
'Strain control example.'  Item1:Problem heading
'stnr02'                   Item2:Local peak/valley file out
1                          Item3:Strain control
'Walker constants in ksi'  Item4:Material identification
1                          Item5:Unit system,kip,ksi,in/in,s
25920.3                    Item6:Elastic modulus, ksi
0.0      16.95    1.16     Item7:Omega0, ksi; N; M
0.0      300000.0 8000.0   Item8:N1; N2, ksi; N3
0.0      1.0e-14  0.0      Item9:N4, N5, N6
1.0e-14  50.931   0.0      Item10:N7; K1, ksi; K2
'R=0.5 strain history'     Item11:History title
'stnh02'                   Item12:peak/valley history in
0.007                      Item13:Max excursion from 0
1                          Item14:Find input max excursion
0.008                      Item15:loading rate,in/in/s
1                          Item16:All of stress-strain out
'stng02'                   Item16A:File complete history out
11                         Item17:Default fine step length
0                          Item18:do not reorder history in
```

Comments may be placed after the last required entry on a line or below the last required line of the file Input file. These areas are not read by the program.

Figure B.3. Strain control: parameter input file "INPUT".

```
2
1
2
1
2
1
2
1
2
1
2
1
2
1
2
1
2
1
2
1
2
1
```

Figure B.4. Strain control: history input file 'STNH02'.

```
'Neuber control example.'  Item1:Problem heading
'neur04'                   Item2:Local peak/valley file out
3                          Item3:Neuber control
2.0                        Item3a:kt
'Walker constants in ksi'  Item4:Material identification
1                          Item5:Unit system,kip,ksi,in/in,s
25920.3                    Item6:Elastic modulus, ksi
0.0      16.95     1.16    Item7:Omega0, ksi; N; M
0.0      300000.0 8000.0   Item8:N1; N2, ksi; N3
0.0      1.0e-14   0.0     Item9:N4, N5, N6
1.0e-14 50.931     0.0     Item10:N7; K1, ksi; K2
'R=0.2355 Load history'    Item11:History title
'lodh04'                   Item12:peak/valley history in
58.6                       Item13:Max excursion from 0, ksi
1                          Item14:Find input max excursion
104                        Item15:loading rate, ksi/s
1                          Item16:All of stress-strain out
'neug04'                   Item16A:File complete history out
11                         Item17:Default fine step length
0                          Item18:do not reorder history in.
Comments placed after last required line entry, or below the
last required line. These areas are not read by the program.
```

Figure B.5. Neuber control: parameter input file "INPUT".

```
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
500.00
117.75
```

Figure B.6. Neuber control: history input file 'LODH04'.

Also, comments were included in the input files. Comments may be included anywhere after the required input on a line or following the last required line in the input file "INPUT". In the file "Input", parameters that were character strings such as file names and descriptive headings must be enclosed in single quotes. In both of the input files, the values were read using list directed free format. The file that contained the input loading history must contain one value per line in sequence. The output for the example problems were given in the next section.

B.2.3 Example's Outputs, Results, and Discussion

The following nine figures and three tables detail the results of the three example computer runs for the three test cases described above. The nine figures contain the actual information as outputted by the computer. The three tables explain the arrangement of the output files columns. The outputs for the three examples of stress, strain, and Neuber control were given in order. For each test case, the two default output files, "OUTPUT" followed by the local stress-strain reversal file, were given. Next, for the same test case, the optional output file giving the values of every stress-strain value computed in sequence was listed.

Figures B.7, B.8, and B.9 are for the stress control case and will be described in detail. The first, Figure B.7, displays the contents of the file "OUTPUT" which was produced by default. This file

echoed the information given in the input file "INPUT". As this information was echoed, a description of what the program had assigned the value was given. Also, a description of any errors were given in this file that relate to the information read from the file "INPUT" or involve the opening of the discussed input/output files. Figure B.8 contains the values of the local stress-strain reversal points computed which was produced by default. Since the input load history was designed to have 10 full cycles and 2 half cycles, this file had 22 computed reversals. This file also listed a zero, "0", reversal where all values were assumed to begin from zero. The reversal file had 9 columns. The quantities in each column are described in Table B.3.

There were a total of 8 possible error codes that may appear in the reversal file encoded as a two digit number. The error code encoding was given in Table B.4. The reversal file was designed to be used as input into a cycle counting and fatigue damage routine.

The final output file for the stress control example was the optional output file which was listed in Figure B.9. This file, using the assumption that all values start at zero, listed an initial zero, "0", for each item tabulated. This optional file contained the value of each stress-strain value computed and listed sequentially in time. Therefore, the details of the complete stress-strain path traversed was made available for uses such as plotting. This optional file may be very large.

```
0 PROBLEM HEADING:
      Stress control example.
0 FILENAME FOR OUTPUT OF RESULTING
      STRESS-STRAIN HISTORY:=stsr03
0 PARAMETER INDICATING THE TYPE OF CONTROL;
      STRAIN, STRESS, OR NEUBER:=           2
0 SPECIMEN TITLE:
      Walker constants in ksi
0 PARAMETER INDICATING THE SYSTEM OF
      UNITS:=           1
0 MODULUS OF ELASTICITY,E:=   25920.30000000000
0 WALKER'S CONSTANTS:
      EQUILIBRIUM STRESS CONSTANT=  0.0000000000000000E+00
      N                          =   16.95000000000000
      M                          =   1.160000000000000
0 WALKER'S CONSTANTS:
      N1                         =  0.0000000000000000E+00
      N2                         =   300000.0000000000
      N3                         =   8000.000000000000
0 WALKER'S CONSTANTS:
      N4                         =  0.0000000000000000E+00
      N5                         =  1.0000000000000002E-14
      N6                         =  0.0000000000000000E+00
0 WALKER'S CONSTANTS:
      N7                         =  1.0000000000000002E-14
      K1                         =   50.93100000000000
      K2                         =  0.0000000000000000E+00
```

Figure B.7. Stress control: default input echo and error message output file "OUTPUT".

```
0 HISTORY TITLE:
      R=0.183 stress history
0 FILENAME WITH NORMALIZED INPUT
      HISTORY:=stsh03
0 MAGNITUDE OF MAXIMUM AMPLITUDE FROM ZERO TO
      SCALE HISTORY TO:=   75.80000000000000
0 PARAMETER CONTROLLING THE AUTOMATED SCANNING
      OF HISTORY TO DETERMINE LARGEST
      AMPLITUDE MAGNITUDE FROM ZERO OF A PEAK
      OR VALLEY NO/YES(0/1):=           1
0 LOADING RATE:=   207.0000000000000
0 PARAMETER INDICATING GRAPHIC OUTPUT DATA
      YES/NO:=          1
0 NAME OF GRAPHIC OUTPUT DATA
      FILE:=stsg03
0 PARAMETER CONTROLLING THE SELECTION OF
      CONVERGENCE PARAMETERS; NUMBER OF
      ITERATIONS, NUMBER OF STEPS, STEP SIZE,
      ETC.:=          11
0 PARAMETER CONTROLLLING REARRANGEMENT OF
      HISTORY:=           0
```

Figure B.7 CONTINUED. Stress control: default input echo and error message output file "OUTPUT".

0	0.0000000000000D+00	0.0000000	0.000	0	0			
1	0.3661835748792D+00	0.0033894	75.800	38	4	33	73	0
2	0.7993787439614D+00	0.0000014	-13.871	45	3	17	51	0
3	0.1232573913043D+01	0.0038466	75.800	45	3	18	34	0
4	0.1665769082126D+01	0.0004597	-13.871	45	3	16	47	0
5	0.2098964251208D+01	0.0043054	75.800	45	3	18	34	0
6	0.2532159420290D+01	0.0009195	-13.871	45	3	16	44	0
7	0.2965354589372D+01	0.0047657	75.800	45	3	18	34	0
8	0.3398549758454D+01	0.0013803	-13.871	45	3	15	42	0
9	0.3831744927536D+01	0.0052267	75.800	45	3	18	34	0
10	0.4264940096618D+01	0.0018427	-13.871	45	3	15	40	0
11	0.4698135265701D+01	0.0056896	75.800	45	3	17	34	0
12	0.5131330434783D+01	0.0023063	-13.871	45	3	15	39	0
13	0.5564525603865D+01	0.0061534	75.800	45	3	17	34	0
14	0.5997720772947D+01	0.0027709	-13.871	45	3	15	38	0
15	0.6430915942029D+01	0.0066184	75.800	45	3	17	34	0
16	0.6864111111111D+01	0.0032363	-13.871	45	3	15	37	0
17	0.7297306280193D+01	0.0070841	75.800	45	3	17	34	0
18	0.7730501449275D+01	0.0037024	-13.871	45	3	15	36	0
19	0.8163696618358D+01	0.0075502	75.800	45	3	17	34	0
20	0.8596891787440D+01	0.0041694	-13.871	45	2	14	35	0
21	0.9030086956522D+01	0.0080173	75.800	45	3	17	34	0
22	0.9463282125604D+01	0.0046376	-13.871	45	2	14	34	0

Figure B.8. Stress control: default local reversal output file 'STSR03'.

0.0000000000000D+00	0.0000000	0.000
0.9636409865243D-02	0.0000770	1.995
0.1927281973049D-01	0.0001539	3.989
0.2890922959573D-01	0.0002309	5.984
0.3854563946097D-01	0.0003078	7.979
0.4818204932621D-01	0.0003848	9.974
0.5781845919146D-01	0.0004617	11.968
0.6745486905670D-01	0.0005387	13.963
0.7709127892194D-01	0.0006157	15.958
0.8672768878719D-01	0.0006926	17.953
0.9636409865243D-01	0.0007696	19.947
0.1060005085177D+00	0.0008465	21.942
0.1156369183829D+00	0.0009235	23.937
0.1252733282482D+00	0.0010005	25.932
0.1349097381134D+00	0.0010777	27.926
0.1445461479786D+00	0.0011552	29.921
0.1541825578439D+00	0.0012337	31.916
0.1638189677091D+00	0.0013139	33.911
0.1734553775744D+00	0.0013958	35.905
0.1830917874396D+00	0.0014790	37.900
0.1927281973049D+00	0.0015630	39.895

Figure B.9. Stress control: optional local path of stress-strain output file 'STSG03'.

TABLE B.3. Listing of Values in Reversal File.

Column	Description
1	-Reversal number.
2	-Current time.
3	-Local strain.
4	-Local stress.
5	-Number of computational steps between reversals.
6	-Minimum number of iterations for any step between the last reversal and the current reversal.
7	-Average number of iterations for any step between the last reversal and the current reversal.
8	-Maximum number of iterations for any step between the last reversal and the current reversal.
9	-Error code for nonconvergence of iterations. It's zero if convergence is achieved, and nonzero if convergence does not take place. If the numerical routines do not converge, the current values are given in each column and the computation stops. Further, column 5 gives the number of computational steps that would have been needed to reach the next reversal from the prior reversal.

For this reason, only the beginning of this file was listed for each of the three test cases. This optional file for the stress control case had 3 columns. The quantities in each column, for the optional file, were described in Table B.5.

The results for the strain control example were given in Figures B.10, B.11, and B.12. The results of the Neuber control example were given in Figures B.13, B.14, and B.15. The same discussions for the stress control case applies to the corresponding output files for both the strain and Neuber control examples. The only exception for Neuber control produced a fourth column of values in the optional file which is listed in Figure B.15. The extra column that was produced in this file for Neuber control calculations was also described in Table B.5.

TABLE B.4. Error Code Encoding for Reversal File[1].

	Values	Error Description
A	3	Stress did not converge
	4	Plastic strain did not converge
	5	Equilibrium stress did not converge
	6	Total strain did not converge
B	1	Mostly elastic behavior
	2	Mostly plastic behavior

[1]Note: Error Code $= A \times 10^1 + B \times 10^0$
Example: 32 = The stress did not converge and mostly plastic behavior.

TABLE B.5. Listing of Values in Optional File.

Column	Description
1	-Current time.
2	-Local strain.
3	-Local stress.
4	-Net section nominal stress. Only appears for the case of Neuber control.

```
0 PROBLEM HEADING:
      Strain control example.
0 FILENAME FOR OUTPUT OF RESULTING
      STRESS-STRAIN HISTORY:=stnr02
0 PARAMETER INDICATING THE TYPE OF CONTROL;
      STRAIN, STRESS, OR NEUBER:=           1
0 SPECIMEN TITLE:
      Walker constants in ksi
0 PARAMETER INDICATING THE SYSTEM OF
      UNITS:=           1
0 MODULUS OF ELASTICITY,E:=   25920.30000000000
0 WALKER'S CONSTANTS:
      EQUILIBRIUM STRESS CONSTANT=  0.0000000000000000E+00
      N                          =   16.95000000000000
      M                          =   1.160000000000000
0 WALKER'S CONSTANTS:
      N1                         =  0.0000000000000000E+00
      N2                         =   300000.0000000000
      N3                         =   8000.000000000000
0 WALKER'S CONSTANTS:
      N4                         =  0.0000000000000000E+00
      N5                         =  1.0000000000000002E-14
      N6                         =  0.0000000000000000E+00
0 WALKER'S CONSTANTS:
      N7                         =  1.0000000000000002E-14
      K1                         =   50.93100000000000
      K2                         =  0.0000000000000000E+00
```

Figure B.10. Strain control: default input echo and error message output file "OUTPUT".

```
0 HISTORY TITLE:
      R=0.5 strain history
0 FILENAME WITH NORMALIZED INPUT
      HISTORY:=stnh02
0 MAGNITUDE OF MAXIMUM AMPLITUDE FROM ZERO TO
      SCALE HISTORY TO:=  6.9999999999999993E-03
0 PARAMETER CONTROLLING THE AUTOMATED SCANNING
      OF HISTORY TO DETERMINE LARGEST
      AMPLITUDE MAGNITUDE FROM ZERO OF A PEAK
      OR VALLEY NO/YES(0/1):=           1
0 LOADING RATE:=  8.0000000000000002E-03
0 PARAMETER INDICATING GRAPHIC OUTPUT DATA
      YES/NO:=          1
0 NAME OF GRAPHIC OUTPUT DATA
      FILE:=stng02
0 PARAMETER CONTROLLING THE SELECTION OF
      CONVERGENCE PARAMETERS; NUMBER OF
      ITERATIONS, NUMBER OF STEPS, STEP SIZE,
      ETC.:=          11
0 PARAMETER CONTROLLING REARRANGEMENT OF
      HISTORY:=           0
```

Figure B.10 CONTINUED. Strain control: default input echo and error message output file "OUTPUT".

0	0.0000000000000D+00	0.0000000	0.000	0	0			
1	0.8750000000000D+00	0.0070000	75.806	70	2	17	73	0
2	0.1312500000000D+01	0.0035000	-15.449	36	2	15	30	0
3	0.1750000000000D+01	0.0070000	69.788	36	3	18	35	0
4	0.2187500000000D+01	0.0035000	-20.340	36	2	16	32	0
5	0.2625000000000D+01	0.0070000	66.105	36	4	19	38	0
6	0.3062500000000D+01	0.0035000	-23.740	36	4	17	41	0
7	0.3500000000000D+01	0.0070000	63.297	36	4	20	40	0
8	0.3937500000000D+01	0.0035000	-26.393	36	4	18	59	0
9	0.4375000000000D+01	0.0070000	61.012	36	4	21	42	0
10	0.4812500000000D+01	0.0035000	-28.569	36	4	18	37	0
11	0.5250000000000D+01	0.0070000	59.093	36	4	23	46	0
12	0.5687500000000D+01	0.0035000	-30.403	36	4	19	40	0
13	0.6125000000000D+01	0.0070000	57.449	36	3	24	51	0
14	0.6562500000000D+01	0.0035000	-31.975	36	4	19	45	0
15	0.7000000000000D+01	0.0070000	56.024	36	4	30	80	0
16	0.7437500000000D+01	0.0035000	-33.340	36	4	20	49	0
17	0.7875000000000D+01	0.0070000	54.777	36	4	26	70	0
18	0.8312500000000D+01	0.0035000	-34.535	36	2	20	46	0
19	0.8750000000000D+01	0.0070000	53.679	36	4	24	51	0
20	0.9187500000000D+01	0.0035000	-35.586	36	3	20	50	0
21	0.9625000000000D+01	0.0070000	52.708	36	4	24	53	0
22	0.1006250000000D+02	0.0035000	-36.516	36	3	20	44	0

Figure B.11. Strain control: default local reversal output file 'STNR02'.

0.0000000000000D+00	0.0000000	0.000
0.1250000000000D-01	0.0001000	2.592
0.2500000000000D-01	0.0002000	5.184
0.3750000000000D-01	0.0003000	7.776
0.5000000000000D-01	0.0004000	10.368
0.6250000000000D-01	0.0005000	12.960
0.7500000000000D-01	0.0006000	15.552
0.8750000000000D-01	0.0007000	18.144
0.1000000000000D+00	0.0008000	20.736
0.1125000000000D+00	0.0009000	23.328
0.1250000000000D+00	0.0010000	25.919
0.1375000000000D+00	0.0011000	28.503
0.1500000000000D+00	0.0012000	31.064
0.1625000000000D+00	0.0013000	33.569
0.1750000000000D+00	0.0014000	36.003
0.1875000000000D+00	0.0015000	38.390
0.2000000000000D+00	0.0016000	40.752
0.2125000000000D+00	0.0017000	43.097
0.2250000000000D+00	0.0018000	45.423
0.2375000000000D+00	0.0019000	47.730
0.2500000000000D+00	0.0020000	50.014

Figure B.12. Strain control: optional local path of stress-strain output file 'STNG02'.

```
0 PROBLEM HEADING:
      Neuber control example.
0 FILENAME FOR OUTPUT OF RESULTING
      STRESS-STRAIN HISTORY:=neur04
0 PARAMETER INDICATING THE TYPE OF CONTROL;
      STRAIN, STRESS, OR NEUBER:=           3
0 ELASTIC STRESS CONCENTRATION
      FACTOR:=   2.000000000000000
0 SPECIMEN TITLE:
      Walker constants in ksi
0 PARAMETER INDICATING THE SYSTEM OF
      UNITS:=           1
0 MODULUS OF ELASTICITY,E:=   25920.30000000000
0 WALKER'S CONSTANTS:
      EQUILIBRIUM STRESS CONSTANT=  0.0000000000000000E+00
      N                          =   16.95000000000000
      M                          =   1.160000000000000
0 WALKER'S CONSTANTS:
      N1                         =  0.0000000000000000E+00
      N2                         =   300000.0000000000
      N3                         =   8000.000000000000
0 WALKER'S CONSTANTS:
      N4                         =  0.0000000000000000E+00
      N5                         =  1.0000000000000002E-14
      N6                         =  0.0000000000000000E+00
0 WALKER'S CONSTANTS:
      N7                         =  1.0000000000000002E-14
      K1                         =   50.93100000000000
      K2                         =  0.0000000000000000E+00
```

Figure B.13. Neuber control: default input echo and error message output file "OUTPUT".

```
0 HISTORY TITLE:
      R=0.2355 Load history
0 FILENAME WITH NORMALIZED INPUT
      HISTORY:=lodh04
0 MAGNITUDE OF MAXIMUM AMPLITUDE FROM ZERO TO
      SCALE HISTORY TO:=   58.60000000000000
0 PARAMETER CONTROLLING THE AUTOMATED SCANNING
      OF HISTORY TO DETERMINE LARGEST
      AMPLITUDE MAGNITUDE FROM ZERO OF A PEAK
      OR VALLEY NO/YES(0/1):=           1
0 LOADING RATE:=   104.0000000000000
0 PARAMETER INDICATING GRAPHIC OUTPUT DATA
      YES/NO:=          1
0 NAME OF GRAPHIC OUTPUT DATA
      FILE:=neug04
0 PARAMETER CONTROLLING THE SELECTION OF
      CONVERGENCE PARAMETERS; NUMBER OF
      ITERATIONS, NUMBER OF STEPS, STEP SIZE,
      ETC.:=          11
0 PARAMETER CONTROLLING REARRANGEMENT OF
      HISTORY:=           0
```

Figure B.13 CONTINUED. Neuber control: default input echo and error message output file "OUTPUT".

0	0.0000000000000D+00	0.0000000	0.000	0	0			
1	0.5634615384615D+00	0.0067679	78.304	118	2	16	48	0
2	0.9942278846154D+00	0.0033655	-12.726	90	5	12	24	0
3	0.1424994230769D+01	0.0069700	73.201	90	9	16	29	0
4	0.1855760576923D+01	0.0035131	-16.393	90	4	15	29	0
5	0.2286526923077D+01	0.0070822	70.385	90	7	17	24	0
6	0.2717293269231D+01	0.0036156	-18.960	90	4	15	27	0
7	0.3148059615385D+01	0.0071679	68.230	90	10	17	27	0
8	0.3578825961538D+01	0.0036985	-21.041	90	8	16	29	0
9	0.4009592307692D+01	0.0072403	66.405	90	11	18	29	0
10	0.4440358653846D+01	0.0037688	-22.812	90	9	16	28	0
11	0.4871125000000D+01	0.0073032	64.819	90	9	18	29	0
12	0.5301891346154D+01	0.0038295	-24.344	90	8	16	27	0
13	0.5732657692308D+01	0.0073586	63.420	90	10	18	30	0
14	0.6163424038461D+01	0.0038831	-25.696	90	9	16	28	0
15	0.6594190384615D+01	0.0074079	62.172	90	10	18	27	0
16	0.7024956730769D+01	0.0039318	-26.926	90	10	17	29	0
17	0.7455723076923D+01	0.0074529	61.034	90	13	18	32	0
18	0.7886489423077D+01	0.0039750	-28.019	90	10	17	29	0
19	0.8317255769231D+01	0.0074933	60.013	90	12	19	33	0
20	0.8748022115384D+01	0.0040149	-29.029	90	8	17	27	0
21	0.9178788461538D+01	0.0075308	59.063	90	11	19	32	0
22	0.9609554807692D+01	0.0040517	-29.963	90	9	17	31	0

Figure B.14. Neuber control: default local reversal output file 'NEUR04'.

0.0000000000000D+00	0.0000000	0.000	0.000
0.4775097783572D-02	0.0000383	0.993	0.497
0.9550195567145D-02	0.0000766	1.986	0.993
0.1432529335072D-01	0.0001150	2.980	1.490
0.1910039113429D-01	0.0001533	3.973	1.986
0.2387548891786D-01	0.0001916	4.966	2.483
0.2865058670143D-01	0.0002299	5.959	2.980
0.3342568448501D-01	0.0002682	6.953	3.476
0.3820078226858D-01	0.0003065	7.946	3.973
0.4297588005215D-01	0.0003449	8.939	4.469
0.4775097783572D-01	0.0003832	9.932	4.966
0.5252607561930D-01	0.0004215	10.925	5.463
0.5730117340287D-01	0.0004598	11.919	5.959
0.6207627118644D-01	0.0004981	12.912	6.456
0.6685136897001D-01	0.0005365	13.905	6.953
0.7162646675359D-01	0.0005748	14.898	7.449
0.7640156453716D-01	0.0006131	15.892	7.946
0.8117666232073D-01	0.0006514	16.885	8.442
0.8595176010430D-01	0.0006897	17.878	8.939
0.9072685788787D-01	0.0007280	18.871	9.436
0.9550195567145D-01	0.0007664	19.864	9.932

Figure B.15. Neuber control: optional local path of stress-strain output file 'NEUG04'.

B.3.0 PROGRAM LISTING

```
      PROGRAM MFATHS
C23456789+123456789+223456789+323456789+423456789+523456789+623456789+7
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(NUMLET=6)
      CHARACTER*(NUMLET) INNAM,OUTNAM
      PARAMETER(INNAM='INPUT',INNUM=5)
      PARAMETER(OUTNAM='OUTPUT',IOTNUM=6)
      PARAMETER(INHNU=20,IOTHNU=21)
      PARAMETER(IOTGNU=22)
      PARAMETER(IRORNU=23)
      PARAMETER(MINIT1=2,MAXIT1=2000)
      PARAMETER(PERERR=0.01D0,ABSERR=PERERR/100.0D0)
C STRAIN INCREMENTS MUST BE DIMENSIONLESS;
C STRESS INCREMENTS MUST BE IN KSI.
      PARAMETER(FSNINC=0.0001D0,FSSINC=2.0D0,FNEINC=FSSINC/2.0D0)
      PARAMETER(CSNINC=0.001D0,CSSINC=20.0D0,CNEINC=FNEINC)
      PARAMETER(IFNINC=24,ICNINC=8)
      PARAMETER(NAMDIM=5)
      DIMENSION AINCLN(2,3),INCCNT(2)
      CHARACTER*(NUMLET) NAMFIL(NAMDIM)
      DIMENSION          NUMFIL(NAMDIM)
      COMMON /CONVER/MINIT2,MAXIT2,CONERR,ELSCON,DCCON,DMCON
      COMMON /WALCON/EOSTS,AN1,AN2,AN3,AN4,AK1,E,Y5,Y7,YK2,YM1,YM2,
     $               YN1,YN2
      COMMON /DTCONS/Z1,Z3,Z4,Z5,Z7,AN6,Z8
      COMMON /GEO/AKT,HISCLE,RATELD,AMPMAG,NEUTYP,IUNITS,IORDER,ISCAN
      COMMON /STPCON/IALTER,NUMINC,STNINC,STSINC,ANEINC
      COMMON /ODRPOS/LMIN,LMAX
      DATA (AINCLN(1,I),I=1,3)/FSNINC,FSSINC,FNEINC/
      DATA (AINCLN(2,I),I=1,3)/CSNINC,CSSINC,CNEINC/
      DATA (INCCNT(I),I=1,2)/IFNINC,ICNINC/
      DATA (NUMFIL(I),I=1,5)/INNUM,IOTNUM,INHNU,IOTHNU,IOTGNU/
      DATA (NAMFIL(I),I=1,5)/INNAM,OUTNAM,3*' '/
      MINIT2=MINIT1
      MAXIT2=MAXIT1
      CONERR=ABSERR
      CALL VALUES(NUMFIL(1),NAMFIL(1),NAMDIM,AINCLN(1,1),INCCNT(1))
      CALL SCAN(NUMFIL(3),AMAX,AMIN,AMAG)
      CALL SCLFAC(HISCLE,AMAG,FACTOR)
      CALL REORDR(NUMFIL(3),IRORNU,FACTOR)
      IF(NEUTYP.EQ.1)THEN
           CALL STRAIN(IRORNU,NUMFIL(4),NAMFIL(5))
      ELSEIF(NEUTYP.EQ.2)THEN
           CALL STRESS(IRORNU,NUMFIL(4),NAMFIL(5))
      ELSEIF(NEUTYP.EQ.3)THEN
           CALL NEUBER(IRORNU,NUMFIL(4),NAMFIL(5))
      ENDIF
      STOP
```

```
      END
C=======================================================================
      SUBROUTINE VALUES(NUMFIL,NAMFIL,NAMDIM,AINCLN,INCCNT)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER(ITEXT=70)
C
C*** FOR THE NADC COMPUTER, REMOVE THE COMMENT CHARACTER TO INCLUDE
C*** THE FOLLOWING TWO LINES.  FOR ALL OTHER COMPUTERS,
C*** THESE TWO LINES SHOULD COMMENTED OUT!
C
C     PARAMETER(NUMLET=6)
C     CHARACTER*NUMLET    NAMFIL(NAMDIM)
C
C
C*** FOR THE NADC COMPUTER, COMMENT OUT THE FOLLOWING LINE. FOR ALL
C*** OTHER COMPUTERS, THE FOLLOWING LINE SHOULD BE INCLUDED BY
C*** REMOVING THE COMMENT CHARACTER!
C
      CHARACTER*(*)       NAMFIL(NAMDIM)
C
      DIMENSION AINCLN(2,*),INCCNT(*)
      DIMENSION           NUMFIL(NAMDIM)
      COMMON /CONVER/MINIT1,MAXIT1,CONERR,ELSCON,DCCON,DMCON
      COMMON /WALCON/EOSTS,AN1,AN2,AN3,AN4,AK1,E,Y5,Y7,YK2,YM1,YM2,
     $               YN1,YN2
      COMMON /DTCONS/Z1,Z3,Z4,Z5,Z7,AN6,Z8
      COMMON /GEO/AKT,HISCLE,RATELD,AMPMAG,NEUTYP,IUNITS,IORDER,ISCAN
      COMMON /STPCON/IALTER,NUMINC,STNINC,STSINC,ANEINC
      CHARACTER*(ITEXT) HEAD,HISNAM,SPCNAM
      OPEN(UNIT=NUMFIL(2),FILE=NAMFIL(2),ERR=2600)
      OPEN(UNIT=NUMFIL(1),FILE=NAMFIL(1),ERR=2500,STATUS='OLD')
      NAMFIL(5)=' '
      IVAR=0
C
C*** ITEM 1
C
      LINE=1
      READ(NUMFIL(1),*,ERR=100)HEAD
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 PROBLEM HEADING:'
      WRITE(NUMFIL(2),*)'        ',HEAD
C
C*** ITEM 2
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=200)NAMFIL(4)
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 FILENAME FOR OUTPUT OF RESULTING'
      WRITE(NUMFIL(2),*)'        STRESS-STRAIN HISTORY:=',NAMFIL(4)
      OPEN(UNIT=NUMFIL(4),FILE=NAMFIL(4),ERR=300)
```

```
C      WRITE(NUMFIL(4),*)' ''',HEAD,''''
C
      WRITE(NUMFIL(4),*)' ''',NAMFIL(4),''''
C
C*** ITEM 3
C
       LINE=LINE+1
       READ(NUMFIL(1),*,ERR=400)NEUTYP
       IVAR=IVAR+1
       WRITE(NUMFIL(2),*)'0 PARAMETER INDICATING THE TYPE OF CONTROL;'
       WRITE(NUMFIL(2),*)'       STRAIN, STRESS, OR NEUBER:=',NEUTYP
C      WRITE(NUMFIL(4),*)NEUTYP
       IF((NEUTYP.LT.1).OR.(NEUTYP.GT.3)) GOTO 500
       IF(NEUTYP.EQ.3)THEN
C
C*** ITEM 3A
C
            LINE=LINE+1
            READ(NUMFIL(1),*,ERR=600)AKT
            WRITE(NUMFIL(2),*)'0 ELASTIC STRESS CONCENTRATION'
            WRITE(NUMFIL(2),*)'       FACTOR:=',AKT
            IVAR=IVAR+1
            IF(AKT.LT.1.0D0) GOTO 610
C           WRITE(NUMFIL(4),*)AKT
       ENDIF
C
C*** ITEM 4
C
       LINE=LINE+1
       READ(NUMFIL(1),*,ERR=900)SPCNAM
       IVAR=IVAR+1
       WRITE(NUMFIL(2),*)'0 SPECIMEN TITLE:'
       WRITE(NUMFIL(2),*)'       ',SPCNAM
C      WRITE(NUMFIL(4),*)' ''',SPCNAM,''''
C
C*** ITEM 5
C
       LINE=LINE+1
       READ(NUMFIL(1),*,ERR=1000)IUNITS
       IVAR=IVAR+1
       WRITE(NUMFIL(2),*)'0 PARAMETER INDICATING THE SYSTEM OF'
       WRITE(NUMFIL(2),*)'       UNITS:=',IUNITS
       IF((IUNITS.LT.1).OR.(IUNITS.GT.2)) GOTO 1100
C      WRITE(NUMFIL(4),*)IUNITS
C
C*** ITEM 6
C
       LINE=LINE+1
       READ(NUMFIL(1),*,ERR=1200)E
       IVAR=IVAR+1
```

```
      WRITE(NUMFIL(2),*)'0 MODULUS OF ELASTICITY,E:=',E
      IF(E.LE.0.0D0) GOTO 1210
C     WRITE(NUMFIL(4),*)E
C
C*** ITEM 7
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1300)EOSTS,AN,AM
      IVAR=IVAR+3
      WRITE(NUMFIL(2),*)'0 WALKER''S CONSTANTS:'
      WRITE(NUMFIL(2),*)'       EQUILIBRIUM STRESS CONSTANT=',EOSTS
      WRITE(NUMFIL(2),*)'       N                          =',AN
      WRITE(NUMFIL(2),*)'       M                          =',AM
      IF(AN.LT.1.0D0) GOTO 1310
      IF(AM.LT.1.0D0) GOTO 1320
C     WRITE(NUMFIL(4),*)EOSTS,AN,AM
C
C*** ITEM 8
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1400)AN1,AN2,AN3
      IVAR=IVAR+3
      WRITE(NUMFIL(2),*)'0 WALKER''S CONSTANTS:'
      WRITE(NUMFIL(2),*)'       N1                         =',AN1
      WRITE(NUMFIL(2),*)'       N2                         =',AN2
      WRITE(NUMFIL(2),*)'       N3                         =',AN3
      IF(AN1.LT.0.0D0) GOTO 1410
      IF(AN2.LT.0.0D0) GOTO 1420
      IF(AN3.LT.0.0D0) GOTO 1430
C     WRITE(NUMFIL(4),*)AN1,AN2,AN3
C
C*** ITEM 9
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1500)AN4,AN5,AN6
      IVAR=IVAR+3
      WRITE(NUMFIL(2),*)'0 WALKER''S CONSTANTS:'
      WRITE(NUMFIL(2),*)'       N4                         =',AN4
      WRITE(NUMFIL(2),*)'       N5                         =',AN5
      WRITE(NUMFIL(2),*)'       N6                         =',AN6
      IF(AN4.LT.0.0D0) GOTO 1510
      IF(AN5.LE.0.0D0) GOTO 1520
      IF(AN6.LT.0.0D0) GOTO 1530
C     WRITE(NUMFIL(4),*)AN4,AN5,AN6
C
C*** ITEM 10
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1600)AN7,AK1,AK2
      IVAR=IVAR+3
```

```
      WRITE(NUMFIL(2),*)'0 WALKER''S CONSTANTS:'
      WRITE(NUMFIL(2),*)'       N7                              =',AN7
      WRITE(NUMFIL(2),*)'       K1                              =',AK1
      WRITE(NUMFIL(2),*)'       K2                              =',AK2
      IF(AN7.LE.0.0D0) GOTO 1610
      IF(AK2.LT.0.0D0) GOTO 1630
      IF(AK1.LE.AK2)   GOTO 1620
      IF((EOSTS.LE.(-1.0D0*AK1)).OR.(EOSTS.GE.AK1)) GOTO 1640
C     WRITE(NUMFIL(4),*)AN7,AK1,AK2
      Y5=-1.0D0*AN5
      Y7=-1.0D0*AN7
      YK2=-1.0D0*AK2
      YM1=AM-1.0D0
      YM2=AM-2.0D0
      YN1=1.0D0-1.0D0/AN
      YN2=-1.0D0/AN
C
C*** ITEM 11
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1700)HISNAM
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 HISTORY TITLE:'
      WRITE(NUMFIL(2),*)'       ',HISNAM
C     WRITE(NUMFIL(4),*)' ''',HISNAM,''''
C
C*** ITEM 12
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1800)NAMFIL(3)
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 FILENAME WITH NORMALIZED INPUT'
      WRITE(NUMFIL(2),*)'       HISTORY:=',NAMFIL(3)
C     WRITE(NUMFIL(4),*)' ''',NAMFIL(3),''''
C
C*** ITEM 13
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1900)HISCLE
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 MAGNITUDE OF MAXIMUM AMPLITUDE FROM ZERO TO'
      WRITE(NUMFIL(2),*)'       SCALE HISTORY TO:=',HISCLE
      IF(HISCLE.LE.0.0D0) GOTO 1905
C     WRITE(NUMFIL(4),*)HISCLE
C
C*** ITEM 14
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1906)ISCAN
      IVAR=IVAR+1
```

```
      WRITE(NUMFIL(2),*)'0 PARAMETER CONTROLLING THE AUTOMATED SCANNING'
      WRITE(NUMFIL(2),*)'       OF HISTORY TO DETERMINE LARGEST'
      WRITE(NUMFIL(2),*)'       AMPLITUDE MAGNITUDE FROM ZERO OF A PEAK'
      WRITE(NUMFIL(2),*)'       OR VALLEY NO/YES(0/1):=',ISCAN
      IF((ISCAN.LT.0).OR.(ISCAN.GT.1)) GOTO 1907
C     WRITE(NUMFIL(4),*)ISCAN
      IF(ISCAN.EQ.0)THEN
C
C*** ITEM 14A
C
           LINE=LINE+1
           READ(NUMFIL(1),*,ERR=1908)AMPMAG
           IVAR=IVAR+1
           WRITE(NUMFIL(2),*)'0 VALUE OF LARGEST AMPLITUDE MAGNITUDE'
           WRITE(NUMFIL(2),*)'       FROM ZERO IN HISTORY OF A PEAK OR'
           WRITE(NUMFIL(2),*)'       VALLEY GIVEN AS USER INPUT'
           WRITE(NUMFIL(2),*)'       AS:=',AMPMAG
           IF(AMPMAG.LE.0.0D0) GOTO 1909
C          WRITE(NUMFIL(4),*)AMPMAG
      ENDIF
C
C*** ITEM 15
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1910)RATELD
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 LOADING RATE:=',RATELD
      IF(RATELD.LE.0.0D0) GOTO 1920
C     WRITE(NUMFIL(4),*)RATELD
C
C*** ITEM 16
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=1922)IGRAPH
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 PARAMETER INDICATING GRAPHIC OUTPUT DATA'
      WRITE(NUMFIL(2),*)'       YES/NO:=',IGRAPH
      IF(IGRAPH.LT.0.OR.IGRAPH.GT.1) GOTO 1923
C     WRITE(NUMFIL(4),*)IGRAPH
      IF(IGRAPH.EQ.1)THEN
C
C*** ITEM 16A
C
           LINE=LINE+1
           READ(NUMFIL(1),*,ERR=1925)NAMFIL(5)
           IVAR=IVAR+1
           WRITE(NUMFIL(2),*)'0 NAME OF GRAPHIC OUTPUT DATA'
           WRITE(NUMFIL(2),*)'       FILE:=',NAMFIL(5)
C          WRITE(NUMFIL(4),*)' ''',NAMFIL(5),''''
      ENDIF
```

```
C
C*** ITEM 17
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=2000)IALTER
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 PARAMETER CONTROLLING THE SELECTION OF'
      WRITE(NUMFIL(2),*)'       CONVERGENCE PARAMETERS; NUMBER OF'
      WRITE(NUMFIL(2),*)'       ITERATIONS, NUMBER OF STEPS, STEP SIZE,'
      WRITE(NUMFIL(2),*)'       ETC.:=',IALTER
C     WRITE(NUMFIL(4),*)IALTER
      IF(((IALTER.LT.0).OR. (IALTER.GT.12)) .OR.
     $   ((IALTER.GT.2).AND.(IALTER.LT.10))) GOTO 2100
      IF(IALTER.EQ.0)THEN
C
C*** ITEM 17A1
C
           LINE=LINE+1
           READ(NUMFIL(1),*,ERR=2200)MINIT2,MAXIT2
           IVAR=IVAR+2
           WRITE(NUMFIL(2),*)'0 MINIMUM NUMBER OF ITERATIONS:=',MINIT2
           WRITE(NUMFIL(2),*)'       MAXIMUM NUMBER OF'
           WRITE(NUMFIL(2),*)'       ITERATIONS:=',MAXIT2
           IF((MINIT2.GE.MAXIT2).OR.(MINIT2.LT.MINIT1)) GOTO 2210
           MINIT1=MINIT2
           MAXIT1=MAXIT2
C          WRITE(NUMFIL(4),*)MINIT1,MAXIT1
C
C*** ITEM 17A2
C
           LINE=LINE+1
           READ(NUMFIL(1),*,ERR=2300)NUMINC,PERERR
           IVAR=IVAR+2
           WRITE(NUMFIL(2),*)'0 NUMBER OF INCREMENTS:=',NUMINC
           WRITE(NUMFIL(2),*)'       PERCENT ERROR FOR'
           WRITE(NUMFIL(2),*)'       CONVERGENCE:=',PERERR,'%'
           IF((NUMINC.LE.0).OR.(PERERR.LE.0)) GOTO 2310
C          WRITE(NUMFIL(4),*)NUMINC,PERERR
           CONERR=PERERR/100.0D0
C
C*** ITEM 17A3
C
           LINE=LINE+1
           READ(NUMFIL(1),*,ERR=2320)STNINC,STSINC,ANEINC
           IVAR=IVAR+3
           WRITE(NUMFIL(2),*)'0 STRAIN INCREMENT:=',STNINC
           WRITE(NUMFIL(2),*)'       STRESS INCREMENT:=',STSINC
           WRITE(NUMFIL(2),*)'       NEUBER INCREMENT:=',ANEINC
           IF((STNINC.LE.0).OR.(STSINC.LE.0).OR.(ANEINC.LE.0)) GOTO 2330
C          WRITE(NUMFIL(4),*)STNINC,STSINC,ANEINC
```

```
      ELSEIF(IALTER.EQ.1)THEN
C           WRITE(NUMFIL(4),*)MINIT1,MAXIT1
            NUMINC=INCCNT(1)
C           WRITE(NUMFIL(4),*)NUMINC,CONERR*100.0
            STNINC=AINCLN(1,1)
            STSINC=AINCLN(1,2)
            ANEINC=AINCLN(1,3)
C           WRITE(NUMFIL(4),*)STNINC,STSINC,ANEINC
      ELSEIF(IALTER.EQ.2)THEN
C           WRITE(NUMFIL(4),*)MINIT1,MAXIT1
            NUMINC=INCCNT(2)
C           WRITE(NUMFIL(4),*)NUMINC,CONERR*100.0
            STNINC=AINCLN(1,1)
            STSINC=AINCLN(1,2)
            ANEINC=AINCLN(1,3)
C           WRITE(NUMFIL(4),*)STNINC,STSINC,ANEINC
      ELSEIF(IALTER.EQ.10)THEN
C
C*** ITEM 17B1
C
            LINE=LINE+1
            READ(NUMFIL(1),*,ERR=2200)MINIT2,MAXIT2
            IVAR=IVAR+2
            WRITE(NUMFIL(2),*)'0 MINIMUM NUMBER OF ITERATIONS:=',MINIT2
            WRITE(NUMFIL(2),*)'        MAXIMUM NUMBER OF'
            WRITE(NUMFIL(2),*)'       ITERATIONS:=',MAXIT2
            IF((MINIT2.GE.MAXIT2).OR.(MINIT2.LT.MINIT1)) GOTO 2210
            MINIT1=MINIT2
            MAXIT1=MAXIT2
C           WRITE(NUMFIL(4),*)MINIT1,MAXIT1
C
C*** ITEM 17B2
C
            LINE=LINE+1
            READ(NUMFIL(1),*,ERR=2305)PERERR
            IVAR=IVAR+1
            WRITE(NUMFIL(2),*)'0 PERCENT ERROR FOR'
            WRITE(NUMFIL(2),*)'        CONVERGENCE:=',PERERR,'%'
            IF(PERERR.LE.0) GOTO 2315
C           WRITE(NUMFIL(4),*)PERERR
            CONERR=PERERR/100.0D0
C
C*** ITEM 17B3
C
            LINE=LINE+1
            READ(NUMFIL(1),*,ERR=2320)STNINC,STSINC,ANEINC
            IVAR=IVAR+3
            WRITE(NUMFIL(2),*)'0 STRAIN INCREMENT:=',STNINC
            WRITE(NUMFIL(2),*)'        STRESS INCREMENT:=',STSINC
            WRITE(NUMFIL(2),*)'        NEUBER INCREMENT:=',ANEINC
```

```
            IF((STNINC.LE.0).OR.(STSINC.LE.0).OR.(ANEINC.LE.0)) GOTO 2330
C           WRITE(NUMFIL(4),*)STNINC,STSINC,ANEINC
      ELSEIF(IALTER.EQ.11)THEN
C           WRITE(NUMFIL(4),*)MINIT1,MAXIT1
C           WRITE(NUMFIL(4),*)CONERR*100.0
            STNINC=AINCLN(1,1)
            STSINC=AINCLN(1,2)
            ANEINC=AINCLN(1,3)
C           WRITE(NUMFIL(4),*)STNINC,STSINC,ANEINC
      ELSEIF(IALTER.EQ.12)THEN
C           WRITE(NUMFIL(4),*)MINIT1,MAXIT1
C           WRITE(NUMFIL(4),*)CONERR*100.0
            STNINC=AINCLN(2,1)
            STSINC=AINCLN(2,2)
            ANEINC=AINCLN(2,3)
C           WRITE(NUMFIL(4),*)STNINC,STSINC,ANEINC
      ENDIF
      OPEN(UNIT=NUMFIL(3),FILE=NAMFIL(3),ERR=2400,STATUS='OLD')
C
C*** ITEM 18
C
      LINE=LINE+1
      READ(NUMFIL(1),*,ERR=2700)IORDER
      IVAR=IVAR+1
      WRITE(NUMFIL(2),*)'0 PARAMETER CONTROLLING REARRANGEMENT OF'
      WRITE(NUMFIL(2),*)'       HISTORY:=',IORDER
      IF((IORDER.LT.0).OR.(IORDER.GT.2)) GOTO 2800
C     WRITE(NUMFIL(4),*)IORDER
      IF(NAMFIL(5).NE.' ')THEN
           OPEN(UNIT=NUMFIL(5),FILE=NAMFIL(5),ERR=2900)
      ENDIF
      RETURN
  100 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' PROBLEM GENERAL HEADING. ASCII TEXT OF ',ITEXT
      WRITE(NUMFIL(2),*)' CHARACTERS. READ USING FREE FORMAT. NOTE THAT'
      WRITE(NUMFIL(2),*)' THE TEXT STRING MUST BE ENCLOSED IN SINGLE'
      WRITE(NUMFIL(2),*)' QUOTES.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
  200 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' FILENAME FOR STRESS-STRAIN DATA OUTPUT, ASCII'
      WRITE(NUMFIL(2),*)' TEXT OF ',NUMLET,' CHARACTERS. READ IN FREE'
      WRITE(NUMFIL(2),*)' FORMAT. NOTE THAT THE TEXT MUST BE ENCLOSED'
      WRITE(NUMFIL(2),*)' IN SINGLE QUOTES.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
  300 WRITE(NUMFIL(2),*)'0ERROR IN SUBROUTINE VALUES.'
      WRITE(NUMFIL(2),*)' UNABLE TO SUCCESSFULLY OPEN DATA FILE FOR'
```

```
      WRITE(NUMFIL(2),*)' STRESS-STRAIN OUTPUT WITH FILE'
      WRITE(NUMFIL(2),*)' NAME=',NAMFIL(4),'.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
  400 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' EXPECT A VALUE FOR FOR THE TYPE OF CONTROL'
      WRITE(NUMFIL(2),*)' CONDITIONS; STRESS, STRAIN, OR NEUBER.'
      WRITE(NUMFIL(2),*)' EXPECTING AN INTEGER VALUE READ USING FREE'
      WRITE(NUMFIL(2),*)' FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
  500 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' VALUE FOR THE TYPE OF ANALYSIS SIMULATION'
      WRITE(NUMFIL(2),*)' SHOULD BE:'
      WRITE(NUMFIL(2),*)' 1=STRAIN CONTROL SIMULATION'
      WRITE(NUMFIL(2),*)' 2=STRESS CONTROL SIMULATION'
      WRITE(NUMFIL(2),*)' 3=NOTCH SPECIMEN NEUBER ANALYSIS ELASTIC'
      WRITE(NUMFIL(2),*)'   NOMINAL STRESSES ONLY.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',NEUTYP
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
  600 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR ELASTIC STRESS'
      WRITE(NUMFIL(2),*)' CONCENTRATION FACTOR KT. EXPECTING A REAL'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE READ USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
  610 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AKT FOR ELASTIC STRESS'
      WRITE(NUMFIL(2),*)' CONCENTRATION FACTOR IS NOT .GE. 1.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AKT
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
  900 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' SPECIMEN/MATERIAL NAME. ASCII TEXT OF ',ITEXT
      WRITE(NUMFIL(2),*)' CHARACTERS. READ USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1000 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR INDICATING THE UNIT'
      WRITE(NUMFIL(2),*)' SYSTEM. EXPECTING AN INTEGER NUMERICAL VALUE'
      WRITE(NUMFIL(2),*)' READ USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
```

```
      STOP
 1100 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' VALUE INDICATING THE SYSTEM OF UNITS DESIRED'
      WRITE(NUMFIL(2),*)' SHOULD BE:'
      WRITE(NUMFIL(2),*)' 1=KIP, KSI, IN/IN(DIMENSIONLESS), SEC'
      WRITE(NUMFIL(2),*)' 2=MN,  MPA, IN/IN(DIMENSIONLESS), SEC'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',IUNITS
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1200 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR WALKER CONSTANT E.'
      WRITE(NUMFIL(2),*)' EXPECTING A REAL NUMERICAL VALUE READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1210 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF E FOR WALKER''S CONSTANT IS NOT'
      WRITE(NUMFIL(2),*)' .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',E
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1300 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR WALKER CONSTANTS'
      WRITE(NUMFIL(2),*)' EOSTS, AN, AND AM.'
      WRITE(NUMFIL(2),*)' EXPECTING REAL NUMERICAL VALUES READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1310 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN FOR WALKER''S CONSTANT IS NOT'
      WRITE(NUMFIL(2),*)' .GE. 1.0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1320 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AM FOR WALKER''S CONSTANT IS NOT'
      WRITE(NUMFIL(2),*)' .GE. 1.0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AM
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1400 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR WALKER CONSTANTS'
      WRITE(NUMFIL(2),*)' AN1, AN2, AND AN3.'
      WRITE(NUMFIL(2),*)' EXPECTING REAL NUMERICAL VALUES READ USING'
```

```
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1410 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN1 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GE. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN1
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1420 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN2 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GE. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN2
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1430 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN3 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GE. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN3
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1500 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR WALKER CONSTANTS'
      WRITE(NUMFIL(2),*)' AN4, AN5, AND AN6.'
      WRITE(NUMFIL(2),*)' EXPECTING REAL NUMERICAL VALUES READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1510 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN4 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GE. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN4
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1520 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN5 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN5
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1530 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN6 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GE. 0.'
```

```
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN6
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1600 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR WALKER CONSTANTS'
      WRITE(NUMFIL(2),*)' AN7, AK1, AND AK2.'
      WRITE(NUMFIL(2),*)' EXPECTING REAL NUMERICAL VALUES READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1610 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AN7 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AN7
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1620 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AK1 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GT. AK2=',AK2
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AK1
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1630 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AK2 FOR WALKER''S CONSTANT IS'
      WRITE(NUMFIL(2),*)' NOT .GE. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AK2
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1640 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF EOSTS FOR WALKER''S CONSTANT'
      WRITE(NUMFIL(2),*)' IS NOT BETWEEN',-1.0*AK1,' AND',AK1
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',EOSTS
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1700 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' HISTORY NAME. ASCII TEXT OF ',ITEXT
      WRITE(NUMFIL(2),*)' CHARACTERS. READ USING FREE FORMAT. TEXT MUST'
      WRITE(NUMFIL(2),*)' BE ENCLOSED IN SINGLE QUOTES.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1800 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' FILE NAME OF FILE THAT CONTAINS PEAK/VALLEY'
      WRITE(NUMFIL(2),*)' LOAD HISTORY TO ANALYZE. ASCII TEXT OF'
```

```
      WRITE(NUMFIL(2),*)' ',NUMLET,' CHARACTERS. READ USING FREE'
      WRITE(NUMFIL(2),*)' FORMAT.'
      WRITE(NUMFIL(2),*)' TEXT MUST BE ENCLOSED IN SINGLE QUOTES.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1900 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE TO SCALE THE MAGNITUDE IN THE'
      WRITE(NUMFIL(2),*)' HISTORY OF THE MOST EXTREME PEAK OR VALLEY.'
      WRITE(NUMFIL(2),*)' EXPECTING A REAL NUMERICAL VALUE READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1905 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF HISCLE TO SCALE THE AMPLITUDE'
      WRITE(NUMFIL(2),*)' MAGNITUDE FROM ZERO OF THE HISTORY IS NOT'
      WRITE(NUMFIL(2),*)' .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',HISCLE
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1906 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE CONTROLLING THE AUTOMATED'
      WRITE(NUMFIL(2),*)' SCANNING OF THE HISTORY FOR THE LARGEST'
      WRITE(NUMFIL(2),*)' AMPLITUDE MAGNITUDE FROM ZERO OF A PEAK OR'
      WRITE(NUMFIL(2),*)' VALLEY.'
      WRITE(NUMFIL(2),*)' EXPECTING AN INTEGER NUMERICAL VALUE READ'
      WRITE(NUMFIL(2),*)' USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1907 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF ISCAN, CONTROLLING THE AUTOMATED'
      WRITE(NUMFIL(2),*)' SCANNING OF THE HISTORY FOR THE LARGEST'
      WRITE(NUMFIL(2),*)' AMPLITUDE MAGNITUDE FROM ZERO OF A PEAK OR'
      WRITE(NUMFIL(2),*)' VALLEY, IS INCORRECT AND SHOULD BE:'
      WRITE(NUMFIL(2),*)' 0=NO AUTOMATED SCANNING. USER GIVEN MAXIMUM'
      WRITE(NUMFIL(2),*)'   AMPLITUDE MAGNITUDE FROM ZERO FOLLOWS.'
      WRITE(NUMFIL(2),*)' 1=AUTOMATED SCANNING DONE.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',ISCAN
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1908 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' VALUE OF THE USER SUPPLIED LARGEST AMPLITUDE'
      WRITE(NUMFIL(2),*)' MAGNITUDE FROM ZERO OF A PEAK OR VALLEY.'
      WRITE(NUMFIL(2),*)' EXPECTING A REAL NUMERICAL VALUE READ'
      WRITE(NUMFIL(2),*)' USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
```

```
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1909 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF AMPMAG, FOR THE LARGEST'
      WRITE(NUMFIL(2),*)' AMPLITUDE MAGNITUDE FROM ZERO OF A PEAK OR'
      WRITE(NUMFIL(2),*)' VALLEY, IS NOT .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',AMPMAG
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1910 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE FOR HISTORY LOADING RATE.'
      WRITE(NUMFIL(2),*)' EXPECTING A REAL NUMERICAL VALUE READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1920 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF RATELD THE RATE OF LOADING IS'
      WRITE(NUMFIL(2),*)' NOT .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',RATELD
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1922 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE TO CONTROL OUTPUT OF GRAPHIC'
      WRITE(NUMFIL(2),*)' DATA FILE.'
      WRITE(NUMFIL(2),*)' EXPECTING AN INTEGER NUMERICAL VALUE READ'
      WRITE(NUMFIL(2),*)' USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1923 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' VALUE CONTROLLING OUTPUT OF GRAPHIC DATA'
      WRITE(NUMFIL(2),*)' FILE. PARAMETERS SHOULD BE:'
      WRITE(NUMFIL(2),*)' 0=DO NOT OUTPUT GRAPHIC DATA FILE.'
      WRITE(NUMFIL(2),*)' 1=OUTPUT GRAPHIC DATA FILE.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',IGRAPH
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 1925 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' FILENAME FOR GRAPHIC DATA OUTPUT, ASCII TEXT'
      WRITE(NUMFIL(2),*)' OF',NUMLET,' CHARACTERS. READ IN FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NOTE THAT THE TEXT MUST BE ENCLOSED IN SINGLE'
      WRITE(NUMFIL(2),*)' QUOTES.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2000 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
```

```
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE TO CONTROL SELECTION OF USER'
      WRITE(NUMFIL(2),*)' DESIRED INPUT OF CONVERGENCE PARAMETERS.'
      WRITE(NUMFIL(2),*)' EXPECTING AN INTEGER NUMERICAL VALUE READ'
      WRITE(NUMFIL(2),*)' USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2100 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' VALUE CONTROLLING INPUT OF CONVERGENCE'
      WRITE(NUMFIL(2),*)' PARAMETERS SHOULD BE:'
      WRITE(NUMFIL(2),*)' 0=USER VALUES CONSTANT NUMBER OF INCREMENTS.'
      WRITE(NUMFIL(2),*)' 1=PROGRAM DEFAULT CONSTANT NUMBER OF'
      WRITE(NUMFIL(2),*)'   INCREMENTS. FINE INCREMENTS.'
      WRITE(NUMFIL(2),*)' 2=PROGRAM DEFAULT CONSTANT NUMBER OF'
      WRITE(NUMFIL(2),*)'   INCREMENTS. COURSE INCREMENTS.'
      WRITE(NUMFIL(2),*)' 10=USER VALUES CONSTANT STEP LENGTH.'
      WRITE(NUMFIL(2),*)' 11=PROGRAM DEFAULT CONSTANT STEP LENGTH.'
      WRITE(NUMFIL(2),*)'    FINE STEP LENGTH.'
      WRITE(NUMFIL(2),*)' 12=PROGRAM DEFAULT CONSTANT STEP LENGTH.'
      WRITE(NUMFIL(2),*)'    COURSE STEP LENGTH.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',IALTER
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2200 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUES CONTROLLING THE NUMBER OF'
      WRITE(NUMFIL(2),*)' ITERATIONS MINIMUM AND MAXIMUM.'
      WRITE(NUMFIL(2),*)' EXPECTING INTEGER NUMERICAL VALUES READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2210 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF MINIT IS NOT .LT. MAXIT OR'
      WRITE(NUMFIL(2),*)' MINIT IS NOT .GE.',MINIT1
      WRITE(NUMFIL(2),*)' VALUE GIVEN MINIT=',MINIT2
      WRITE(NUMFIL(2),*)' MAXIT=',MAXIT2,'NUMBER OF SUCCESSFULLY READ'
      WRITE(NUMFIL(2),*)' NUMERICAL AND ASCII VARIABLES FROM DATA'
      WRITE(NUMFIL(2),*)' FILE=',IVAR
      STOP
 2300 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUES CONTROLLING THE CONVERGENCE'
      WRITE(NUMFIL(2),*)' THE NUMBER OF STEPS TO A REVERSAL,'
      WRITE(NUMFIL(2),*)' THE CONVERGENCE ERROR IN PERCENT.'
      WRITE(NUMFIL(2),*)' EXPECTING AN INTEGER VALUE FOLLOWED BY A REAL'
      WRITE(NUMFIL(2),*)' VALUE READ USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2305 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
```

```
      WRITE(NUMFIL(2),*)' NUMERICAL VALUE CONTROLLING THE CONVERGENCE'
      WRITE(NUMFIL(2),*)' THE CONVERGENCE ERROR IN PERCENT.'
      WRITE(NUMFIL(2),*)' EXPECTING REAL NUMERICAL VALUE READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2310 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF NUMINC, OR PERERR IS NOT'
      WRITE(NUMFIL(2),*)' .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN NUMINC=',NUMINC,' PERERR=',PERERR
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2315 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF PERERR IS NOT'
      WRITE(NUMFIL(2),*)' .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN PERERR=',PERERR
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2320 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' NUMERICAL VALUES CONTROLLING THE CONVERGENCE'
      WRITE(NUMFIL(2),*)' THE 3 STEP INCREMENT LENGTHS USED TO TRAVERSE'
      WRITE(NUMFIL(2),*)' TO A REVERSAL.'
      WRITE(NUMFIL(2),*)' THE 3 VALUES ARE: THE STRAIN, STRESS AND'
      WRITE(NUMFIL(2),*)' NEUBER INCREMENTS.'
      WRITE(NUMFIL(2),*)' EXPECTING REAL NUMERICAL VALUES READ USING'
      WRITE(NUMFIL(2),*)' FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2330 WRITE(NUMFIL(2),*)'OERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF STNINC, STSINC, OR ANEINC IS NOT'
      WRITE(NUMFIL(2),*)' .GT. 0.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN STNINC=',STNINC,' STSINC=',STSINC
      WRITE(NUMFIL(2),*)' AND ANEINC=',ANEINC
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2400 WRITE(NUMFIL(2),*)'OERROR IN SUBROUTINE VALUES.'
      WRITE(NUMFIL(2),*)' UNABLE TO SUCCESSFULLY OPEN OLD DATA FILE FOR'
      WRITE(NUMFIL(2),*)' INPUT OF HISTORY WITH FILE'
      WRITE(NUMFIL(2),*)' NAME=',NAMFIL(3),'.'
      STOP
 2500 WRITE(NUMFIL(2),*)'OERROR IN SUBROUTINE VALUES.'
      WRITE(NUMFIL(2),*)' UNABLE TO SUCCESSFULLY OPEN OLD DATA FILE FOR'
      WRITE(NUMFIL(2),*)' INPUT OF CALCULATION PARAMETERS WITH UNIT'
      WRITE(NUMFIL(2),*)' NUMBER=',NUMFIL(1),' AND FILE'
      WRITE(NUMFIL(2),*)' NAME=',NAMFIL(1),'.'
```

```
      STOP
 2600 WRITE(*,*)'0ERROR IN SUBROUTINE VALUES.'
      WRITE(*,*)' UNABLE TO SUCCESSFULLY OPEN DATA FILE FOR'
      WRITE(*,*)' OUTPUT OF PROGRAM MESSAGES WITH UNIT'
      WRITE(*,*)' NUMBER=',NUMFIL(2),' AND FILE'
      WRITE(*,*)' NAME=',NAMFIL(2),'.'
      STOP
 2700 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' ORDERING OF HISTORY PARAMETER. AN INTEGER'
      WRITE(NUMFIL(2),*)' VALUE READ USING FREE FORMAT.'
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2800 WRITE(NUMFIL(2),*)'0ERROR READING RECORD=',LINE,' OF DATA FILE.'
      WRITE(NUMFIL(2),*)' THE VALUE OF IORDER INDICATING THE ORDERING'
      WRITE(NUMFIL(2),*)' OF THE HISTORY IS NOT ONE OF THE FOLLOWING:'
      WRITE(NUMFIL(2),*)' 0=HISTORY NOT REORDERED USED AS GIVEN.'
      WRITE(NUMFIL(2),*)' 1=HISTORY REORDERED STARTING WITH LOWEST'
      WRITE(NUMFIL(2),*)'   VALLEY.'
      WRITE(NUMFIL(2),*)' 2=HISTORY REORDERED STARTING WITH HIGHEST'
      WRITE(NUMFIL(2),*)'   PEAK.'
      WRITE(NUMFIL(2),*)' VALUE GIVEN=',IORDER
      WRITE(NUMFIL(2),*)' NUMBER OF SUCCESSFULLY READ NUMERICAL AND'
      WRITE(NUMFIL(2),*)' ASCII VARIABLES FROM DATA FILE=',IVAR
      STOP
 2900 WRITE(NUMFIL(2),*)'0ERROR IN SUBROUTINE VALUES.'
      WRITE(NUMFIL(2),*)' UNABLE TO SUCCESSFULLY OPEN DATA FILE FOR'
      WRITE(NUMFIL(2),*)' OUTPUT OF GRAPHIC DATA.'
      WRITE(NUMFIL(2),*)' FILE NAME=',NAMFIL(5),'.'
      STOP
      END
C=====================================================================
      SUBROUTINE SCAN(IHISIN,AMAX,AMIN,AMAG)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /GEO/P2,P4,P5,AMPMAG,IP1,IP3,IORDER,ISCAN
      COMMON /ODRPOS/LMIN,LMAX
      IF((ISCAN.EQ.1).OR.(IORDER.NE.0))THEN
           REWIND(IHISIN)
           LINE=0
           READ(IHISIN,*)VALUE
           LINE=LINE+1
           AMAX=VALUE
           AMIN=VALUE
           LMIN=LINE
           LMAX=LINE
  100      READ(IHISIN,*,END=200)VALUE
                LINE=LINE+1
                IF(VALUE.GT.AMAX)THEN
                     AMAX=VALUE
                     LMAX=LINE
```

```
                ELSEIF(VALUE.LT.AMIN)THEN
                     AMIN=VALUE
                     LMIN=LINE
                ENDIF
           GOTO 100
 200       CONTINUE
      ENDIF
      IF(ISCAN.EQ.0)THEN
           AMAG=AMPMAG
      ELSE
           AAMAX=ABS(AMAX)
           AAMIN=ABS(AMIN)
           IF(AAMAX.GT.AAMIN)THEN
                AMAG=AAMAX
           ELSE
                AMAG=AAMIN
           ENDIF
      ENDIF
      RETURN
      END
C=======================================================================
      SUBROUTINE SCLFAC(HISCLE,AMAG,FACTOR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      FACTOR=HISCLE/AMAG
      RETURN
      END
C=======================================================================
      SUBROUTINE REORDR(IHISIN,NEWODR,FACTOR)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /GEO/PAD2,PAD4,PAD5,PAD8,IPAD1,IPAD3,IORDER,IPAD7
      COMMON /ODRPOS/LMIN,LMAX
      OPEN(UNIT=NEWODR,STATUS='SCRATCH')
      REWIND(IHISIN)
      IF(IORDER.EQ.0)THEN
  10       READ(IHISIN,*,END=20)VALUE
                WRITE(NEWODR,*)VALUE*FACTOR
           GOTO 10
  20       CONTINUE
      ELSEIF(IORDER.EQ.1)THEN
           IF(LMIN.GT.1)THEN
                DO 30, I=1,LMIN-1
                     READ(IHISIN)
  30            CONTINUE
           ENDIF
  40       READ(IHISIN,*,END=50)VALUE
                WRITE(NEWODR,*)VALUE*FACTOR
           GOTO 40
  50       CONTINUE
           IF(LMIN.GT.1)THEN
                REWIND(IHISIN)
```

```
                DO 60, I=1,LMIN-1
                     READ(IHISIN,*)VALUE
                     WRITE(NEWODR,*)VALUE*FACTOR
   60           CONTINUE
           ENDIF
      ELSE
           IF(LMAX.GT.1)THEN
                DO 70, I=1,LMAX-1
                     READ(IHISIN)
   70           CONTINUE
           ENDIF
   80      READ(IHISIN,*,END=90)VALUE
                WRITE(NEWODR,*)VALUE*FACTOR
           GOTO 80
   90      CONTINUE
           IF(LMAX.GT.1)THEN
                REWIND(IHISIN)
                DO 100, I=1,LMAX-1
                     READ(IHISIN,*)VALUE
                     WRITE(NEWODR,*)VALUE*FACTOR
  100           CONTINUE
           ENDIF
      ENDIF
      END
C=======================================================================
      SUBROUTINE STRAIN(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /STPCON/IALTER,IPAD1,PAD2,PAD3,PAD4
      IF(IALTER.LE.2)THEN
           CALL STNNUM(IHISIN,NUMFIL(1),NAMGPH)
      ELSE
           CALL STNLEN(IHISIN,NUMFIL(1),NAMGPH)
      ENDIF
      RETURN
      END
C=======================================================================
      SUBROUTINE STNNUM(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /WALCON/EOSTS,P2,P3,P4,P5,P6,E,P8,P9,P10,YM1,P12,YN1,P14
      COMMON /GEO/AP2,AP4,RATELD,AP8,IAP1,IUNITS,IAP6,IAP7
      COMMON /STPCON/IBP1,NUMINC,STNINC,BP4,BP5
      COMMON /DTCONS/Z1,Z3,Z4,Z5,Z7,AN6,CP7
      COMMON /CONVER/IDP1,MAXIT,CONERR,ELSCON,DCCON,DP6
C TIME, T=0 INITIALIZATION
      MAVG=0
      REWIND(IHISIN)
```

```
      T2=0.0D0
      BM2STS=0.0D0
      T2STN=0.0D0
      P2STN=0.0D0
      C2STN=0.0D0
      E2STS=E0STS
      STNBGN=0.0D0
      DMSTS=0.0D0
      IERR=0
      M=0
      I=0
      IF(IUNITS.EQ.1)THEN
           WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
           ENDIF
      ELSE
           WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
           ENDIF
      ENDIF
   10 READ(IHISIN,*,END=40)STNEND
           MIN=MAXIT
           MAX=0
           STNRNG=STNEND-STNBGN
C
C*** CHECK FOR STNRNG=0. IF TRUE, READ ANOTHER REVERSAL.
C
           IF(STNRNG.EQ.0.0D0) GOTO 10
           NUMSTP=NUMINC
           STNSTP=STNRNG/DBLE(NUMSTP)
           IF(ABS(STNSTP).LT.STNINC)THEN
                NUMSTP=INT(ABS(STNRNG/STNINC))+1
                STNSTP=STNRNG/DBLE(NUMSTP)
           ENDIF
           DTSTN=STNSTP
           DT=ABS(STNSTP)/RATELD
           Z1=AN6*DT
           Z3=Z1*YM1
           Z4=E*DT
           Z7=DT*YN1
           DCABS=ABS(DTSTN)
           DCCON=DCABS*CONERR
           Z5=E*DTSTN
           ELSCON=DCABS/2.0D0
           DESTS=0.0D0
           DO 20, II=1,NUMSTP
                DPSTN=DTSTN
                DCSTN=DCABS
```

```
                BM1STS=BM2STS
                P1STN=P2STN
                C1STN=C2STN
                E1STS=E2STS
                T1STN=T2STN
                T1=T2
                CALL STNCON(BM1STS,P1STN,C1STN,E1STS,T1STN,T1,DT,DTSTN,
     $                      DMSTS,DPSTN,DCSTN,DESTS,BM2STS,P2STN,C2STN,
     $                      E2STS,T2STN,T2,IERR,M)
                MAVG=MAVG+M
                IF(M.GT.MAX) MAX=M
                IF(M.LT.MIN) MIN=M
                IF(NAMGPH.NE.' ')THEN
                     IF(IUNITS.EQ.1)THEN
                          WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
                     ELSE
                          WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
                     ENDIF
                ENDIF
                IF(IERR.NE.0)GOTO 30
   20       CONTINUE
   30       CONTINUE
            I=I+1
            MAVG=MAVG/NUMSTP
            IF(IUNITS.EQ.1)THEN
                 WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                MAX,IERR
            ELSE
                 WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                MAX,IERR
            ENDIF
            IF(IERR.NE.0)GOTO 40
            STNBGN=T2STN
         GOTO 10
   40 CONTINUE
      RETURN
 1000 FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.3,1X,4(I4,1X),I2)
 2000 FORMAT(1X,D20.13,1X,F10.7,1X,F8.3)
 3000 FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.2,1X,4(I4,1X),I2)
 4000 FORMAT(1X,D20.13,1X,F10.7,1X,F8.2)
      END
C=======================================================================
      SUBROUTINE STNLEN(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /WALCON/EOSTS,P2,P3,P4,P5,P6,E,P8,P9,P10,YM1,P12,YN1,P14
      COMMON /GEO/AP2,AP4,RATELD,AP8,IAP1,IUNITS,IAP6,IAP7
      COMMON /STPCON/IBP1,IBP2,STNINC,BP4,BP5
      COMMON /DTCONS/Z1,Z3,Z4,Z5,Z7,AN6,CP7
```

```
      COMMON /CONVER/IDP1,MAXIT,CONERR,ELSCON,DCCON,DP6
C TIME, T=0 INITIALIZATION
      MAVG=0
      REWIND(IHISIN)
      T2=0.0D0
      BM2STS=0.0D0
      T2STN=0.0D0
      P2STN=0.0D0
      C2STN=0.0D0
      E2STS=E0STS
      STNBGN=0.0D0
      DMSTS=0.0D0
      IERR=0
      M=0
      I=0
      IF(IUNITS.EQ.1)THEN
           WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
           ENDIF
      ELSE
           WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
           ENDIF
      ENDIF
   10 READ(IHISIN,*,END=40)STNEND
           MIN=MAXIT
           MAX=0
           STNRNG=STNEND-STNBGN
C
C*** CHECK FOR STNRNG=0. IF TRUE, READ ANOTHER REVERSAL.
C
           IF(STNRNG.EQ.0.0D0) GOTO 10
           NUMSTP=INT(ABS(STNRNG/STNINC))+1
           STNSTP=STNRNG/DBLE(NUMSTP)
           DTSTN=STNSTP
           DT=ABS(STNSTP)/RATELD
           Z1=AN6*DT
           Z3=Z1*YM1
           Z4=E*DT
           Z7=DT*YN1
           DCABS=ABS(DTSTN)
           DCCON=DCABS*CONERR
           Z5=E*DTSTN
           ELSCON=DCABS/2.0D0
           DESTS=0.0D0
           DO 20, II=1,NUMSTP
                DPSTN=DTSTN
                DCSTN=DCABS
```

```
                BM1STS=BM2STS
                P1STN=P2STN
                C1STN=C2STN
                E1STS=E2STS
                T1STN=T2STN
                T1=T2
                CALL STNCON(BM1STS,P1STN,C1STN,E1STS,T1STN,T1,DT,DTSTN,
     $                      DMSTS,DPSTN,DCSTN,DESTS,BM2STS,P2STN,C2STN,
     $                      E2STS,T2STN,T2,IERR,M)
                MAVG=MAVG+M
                IF(M.GT.MAX) MAX=M
                IF(M.LT.MIN) MIN=M
                IF(NAMGPH.NE.' ')THEN
                     IF(IUNITS.EQ.1)THEN
                          WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
                     ELSE
                          WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
                     ENDIF
                ENDIF
                IF(IERR.NE.0)GOTO 30
   20       CONTINUE
   30       CONTINUE
            I=I+1
            MAVG=MAVG/NUMSTP
            IF(IUNITS.EQ.1)THEN
                 WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                MAX,IERR
            ELSE
                 WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                MAX,IERR
            ENDIF
            IF(IERR.NE.0)GOTO 40
            STNBGN=T2STN
         GOTO 10
   40 CONTINUE
         RETURN
 1000 FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.3,1X,4(I4,1X),I2)
 2000 FORMAT(1X,D20.13,1X,F10.7,1X,F8.3)
 3000 FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.2,1X,4(I4,1X),I2)
 4000 FORMAT(1X,D20.13,1X,F10.7,1X,F8.2)
      END
C=====================================================================
      SUBROUTINE STNCON(BM1STS,P1STN,C1STN,E1STS,T1STN,T1,DT,DTSTN,
     $DMSTS,DPSTN,DCSTN,DESTS,BM2STS,P2STN,C2STN,
     $E2STS,T2STN,T2,IERR,M)
C*********************************************************************
C FOLLOWING INPUT ONLY TO SUBROUTINE
C   FORTRAN CONSTANTS: MINIT,MAXIT,CONERR
C   WALKER CONSTANTS: EOSTS,AN,AM,AN1,AN2,AN3,AN4,AN5,AN6,AN7,AK1,
C                     AK2,E
```

```
C   VARIABLES: BM1STS,P1STN,C1STN,E1STS,DT,DTSTN
C FOLLOWING INPUT TO AND OUTPUT FROM SUBROUTINE
C   VARIABLES: DPSTN
C FOLLOWING OUTPUT ONLY FROM SUBROUTINE
C   VARIABLES: BM2STS,P2STN,C2STN,E2STS
C*********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /CONVER/MINIT,MAXIT,CONERR,ELSCON,DCCON,PAD1
      COMMON /WALCON/EOSTS,AN1,AN2,AN3,AN4,AK1,E,Y5,Y7,YK2,YM1,YM2,
     $               YN1,YN2
      COMMON /DTCONS/Z1,Z3,Z4,Z5,Z7,AN6,PAD2
C
C*** CHECK FOR DT=0.0 WHICH WOULD RESULT IN DIVISION BY ZERO
C
      IF(DT.EQ.0.0D0)THEN
           WRITE(*,*)'0 ERROR DT=0'
           WRITE(*,*)'  SUBROUTINE STNCON: CHECK DT CALCULATION IN'
           WRITE(*,*)'                     CALLING PROGRAM'
           IERR=70
           RETURN
      ENDIF
C
C*** THE STEP NUMBERING APPROXIMATES THAT GIVEN IN WALKER'S PAPER.
C
C*** INITIALIZE SUBROUTINE:
C***  STEP #3: PLASTIC STRAIN GUESS AT END OF INCREMENT.
C***  STEP #4: INITIAL GUESS FOR EQUILIBRIUM STRESS AT END OF INCREMENT.
C
      Z2=E1STS-(EOSTS+AN1*P1STN)
      Z6=BM1STS-E1STS
      P2STN=P1STN+DPSTN
      E2STS=E1STS+DESTS
C
C*** STEP #7: START OF INNER ITERATION LOOP. M ITERATION LOOP INDEX.
C***           INITIALIZE CONTROL PARAMETER FOR LOOP.
C
      M=0
C
C*** STEP #8: START OF REPETITIVE STATEMENTS OF LOOP.
C
    8 M=M+1
C
C*** STORE VALUES FOR CONVERGENCE TEST
C
      OLDDE=DESTS
      OLDDC=DCSTN
      OLDDMS=DMSTS
C
C*** STEP #9:  COMPUTE THE CUMULATIVE PLASTIC STRAIN AT THE END OF THE
C***           INCREMENT.
```

```
C*** STEP #10: CALCULATE DRAG STRESS (ISOTROPIC HARDENING) AT END OF
C***            INCREMENT.
C*** STEP #11: EVALUATE THE GUESS FOR DELTA G.
C***            X1-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #13: COMPUTE EQUILIBRIUM STRESS GUESS AT END OF INCREMENT.
C***            X2,X3,X4,X5,X6-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #16: COMPUTE FUNCTION OF DELTA G TO ITERATE  FOR FINDING ITS
C***            ZERO TO IMPROVE THE ESTIMATE OF DELTA G.
C***            X7,X8-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C
      C2STN=C1STN+DCSTN
      BK2=AK1+YK2*EXP(Y7*C2STN)
      X1=(AN3+AN4*EXP(Y5*C2STN))*DCSTN
      DG=X1+Z1*ABS(E2STS)**YM1
      X2=EXP(-1.0D0*DG)
      X3=AN2*DPSTN
      X5=X2*Z2
      X6=E0STS+AN1*P2STN
      IF(DG.NE.0.0D0)THEN
           X4=X3*(1.0D0-X2)/DG
           E2STS=X6+X5+X4
           X7=ABS(E2STS)
           X8=X7**YM2
           F1DG=X1+Z1*X8*X7-DG
C
C*** STEP #17: COMPUTE THE DERIVATIVE OF THE PROCEEDING FUNCTION
C
           IF(E2STS.GE.0.0D0)THEN
                DRF1DG=Z3*X8*((X2*X3-X4)/DG-X5)-1.0D0
           ELSE
                DRF1DG=Z3*X8*(X5+(X4-X2*X3)/DG)-1.0D0
           ENDIF
C
C*** STEP  #18: REFINE DELTA G BY A SINGLE NEWTON ITERATION.
C*** STEP  #19: COMPUTE REFINED VALUE OF THE EQUILIBRIUM STRESS BY
C***            DIRECT SUBSTITUTION ITERATION, SUBSTITUTING REFINED
C***            DELTA G.
C***            X2-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #19A: COMPUTE THE EQUILIBRIUM STRESS INCREMENT
C*** STEP  #20: COMPUTE GUESS FOR DELTA Q.
C*** STEP  #21: COMPUTE GUESS FOR STRESS AT END OF INCREMENT. TERMED THE
C***            MECHANICAL STRESS AT THE END OF THE TIME INCREMENT,
C***            BM2STS.
C***            X9,X10,X11-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #21A: COMPUTE STRESS INCREMENT.
C*** STEP  #22: REFINE THE PLASTIC STRAIN INCREMENT.
C*** STEP  #24: REFINE THE CUMULATIVE PLASTIC STRAIN INCREMENT.
C*** STEP #24A: COMPUTE REFINED PLASTIC STRAIN VALUE AT THE END OF THE
C***            INCREMENT.
C*** STEP #24B: COMPUTE REFINED DRAG STRESS.
```

```
C
            DG=DG-F1DG/DRF1DG
            X2=EXP(-1.0D0*DG)
            E2STS=X6+X3*(1.0D0-X2)/DG+X2*Z2
      ELSE
            E2STS=X3+E1STS
      ENDIF
      DESTS=E2STS-E1STS
      DQ=Z4/BK2*(DCSTN/DT)**YN1
      X9=EXP(-1.0D0*DQ)
      X10=Z5-DESTS
      X11=Z6*X9
      IF(DQ.NE.0.0D0)THEN
            BM2STS=E2STS+X11+X10*(1.0D0-X9)/DQ
            DMSTS=BM2STS-BM1STS
            DPSTN=DTSTN-DMSTS/E
            DCSTN=ABS(DPSTN)
            C2STN=DCSTN+C1STN
            BK2=AK1+YK2*EXP(Y7*C2STN)
C
C*** STEP  #25: COMPUTE THE FUNCTION OF DELTA Q TO ITERATE TO DETERMINE
C***            ITS ZERO SO AS TO REFINE DELTA Q IN THE PROCESS.
C***            X12,X13-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C
            X12=DCSTN/DT
            X13=X12**YN2
            F2DQ=DQ-Z4/BK2*X12*X13
C
C*** STEP #26. COMPUTE THE DERIVATIVE OF THE PROCEEDING FUNCTION.
C
            IF(DPSTN.GE.0.0)THEN
                  DRF2DQ=1.0D0+X13*Z7/BK2*(X10*((DQ+1.0D0)*X9-1.0D0)/DQ/
     $                  DQ-X11)
            ELSE
                  DRF2DQ=1.0D0+X13*Z7/BK2*(X11+X10/DQ*(1.0D0-(1.0D0+DQ)*
     $                  X9)/DQ)
            ENDIF
C
C*** STEP  #27: REFINE DELTA Q WITH A SINGLE NEWTON ITERATION.
C*** STEP  #28: REFINE THE FUTURE STRESS AT THE END OF THE INCREMENT BY
C***            DIRECT SUBSTITUTION ITERATION, SUBSTITUTING REFINED
C***            DELTA Q.
C***            X9-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #28A: COMPUTE STRESS INCREMENT.
C*** STEP  #29: REFINE THE PLASTIC STRAIN INCREMENT.
C*** STEP  #30: REFINE THE CUMULATIVE PLASTIC STRAIN INCREMENT.
C*** STEP #30A: COMPUTE THE PLASTIC STRAIN AT THE END OF THE INCREMENT.
C***            COPY OF STEP #12 PLACED HERE FOR COMPUTATIONAL
C***            EFFICIENCY.
C
```

```
            DQ=DQ-F2DQ/DRF2DQ
            X9=EXP(-1.0D0*DQ)
            BM2STS=E2STS+X10/DQ*(1.0D0-X9)+Z6*X9
            DMSTS=BM2STS-BM1STS
            DPSTN=DTSTN-DMSTS/E
            DCSTN=ABS(DPSTN)
            P2STN=P1STN+DPSTN
      ELSE
            DMSTS=Z5
            DPSTN=0.0D0
            DCSTN=0.0D0
            P2STN=P1STN
            BM2STS=BM1STS+DMSTS
      ENDIF
C
C*** CONVERGENCE CHECKS:
C
C***  1: FORCE A MINIMUM NUMBER OF ITERATIONS
C
      IF(M.LT.MINIT) GOTO 8
C
C***  2: CHECK FOR EXCEEDING ALLOWED MAXIMUM NUMBER OF ITERATIONS AND
C***    2A: FLAG NONCONVERGENCE OF A MOSTLY ELASTIC STRAIN INCREMENT
C***    2B: OR FLAG MOSTLY A PLASTIC INCREMENT WITH NONCONVERGENCE
C
      IF(M.GE.MAXIT)THEN
            IF(DCSTN.LT.ELSCON)THEN
                  IERR=1+IERR
                  GOTO 7000
            ELSE
                  IERR=2+IERR
                  GOTO 7000
            ENDIF
      ENDIF
C
C***  3: CHECK CONVERGENCE OF THE MECHANICAL STRESS.
C
      IF(BM2STS.EQ.0.0D0)THEN
            IF(ABS(OLDDMS-DMSTS).GT.CONERR)THEN
                  IERR=30
                  GOTO 8
            ENDIF
      ELSE
            IF(ABS((OLDDMS-DMSTS)/BM2STS).GT.CONERR)THEN
                  IERR=30
                  GOTO 8
            ENDIF
      ENDIF
C
C***  4: CHECK CONVERGENCE OF CUMULATIVE PLASTIC STRAIN INCREMENT AND
```

```
C***     THEREFORE ALSO THE PLASTIC STRAIN INCREMENT. THIS CHECK IS
C***     SKIPPED IF THE PLASTIC INCREMENT MAGNITUDE IS BELOW A
C***     PERCENTAGE OF THE MAGNITUDE OF THE TOTAL STRAIN INCREMENT.
C
      IF(DCSTN.GE.DCCON)THEN
           IF(P2STN.EQ.0.0D0)THEN
                IF(ABS(OLDDC-DCSTN).GT.CONERR) THEN
                     IERR=40
                     GOTO 8
                ENDIF
           ELSE
                IF(ABS((OLDDC-DCSTN)/P2STN).GT.CONERR)THEN
                     IERR=40
                     GOTO 8
                ENDIF
           ENDIF
      ENDIF
C
C***  5: CHECK CONVERGENCE OF THE EQUILIBRIUM STRESS. DONE ONLY WHEN THE
C***     MAGNITUDE OF THE EQUILIBRIUM STRESS EXCEEDS A PERCENTAGE OF THE
C***     MAGNITUDE OF THE MECHANICAL STRESS.
C
      IF(ABS(DESTS).GE.ABS(DMSTS)*CONERR)THEN
           IF(E2STS.EQ.0.0D0)THEN
                IF(ABS(OLDDE-DESTS).GT.CONERR)THEN
                     IERR=50
                     GOTO 8
                ENDIF
           ELSE
                IF(ABS((OLDDE-DESTS)/E2STS).GT.CONERR)THEN
                     IERR=50
                     GOTO 8
                ENDIF
           ENDIF
      ENDIF
C
C*** CONVERGENCE SATISFIED
C
      IERR=0
C
C*** STEP #35: ITERATION COMPLETE UPDATE STATE VARIABLES AND PRINT OUT
C***           RESULTS.
C
 7000 C2STN=C1STN+DCSTN
      T2STN=T1STN+DTSTN
      T2=DT+T1
C
C*** STEP #36. GO TO THE NEXT TIME INCREMENT.
C
      RETURN
```

```
      END
C=======================================================================
      SUBROUTINE STRESS(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /STPCON/IALTER,IPAD1,PAD2,PAD3,PAD4
      IF(IALTER.LE.2)THEN
           CALL STSNUM(IHISIN,NUMFIL(1),NAMGPH)
      ELSE
           CALL STSLEN(IHISIN,NUMFIL(1),NAMGPH)
      ENDIF
      RETURN
      END
C=======================================================================
      SUBROUTINE STSNUM(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /WALCON/EOSTS,P2,P3,P4,P5,P6,E,P8,P9,P10,YM1,P12,YN1,P14
      COMMON /GEO/AP2,AP4,RATELD,AP8,IAP1,IUNITS,IAP6,IAP7
      COMMON /STPCON/IBP1,NUMINC,BP3,STSINC,BP5
      COMMON /DTCONS/Z1,Z3,Z4,CP4,Z7,AN6,Z8
      COMMON /CONVER/IDP1,MAXIT,CONERR,ELSCON,DP5,DMCON
C FOLLOWING IS FOR STRESS CONTROL
C TIME, T=0 INITIALIZATION
      MAVG=0
      REWIND(IHISIN)
      T2=0.0D0
      BM2STS=0.0D0
      T2STN=0.0D0
      P2STN=0.0D0
      C2STN=0.0D0
      E2STS=EOSTS
      STSBGN=0.0D0
      IERR=0
      M=0
      I=0
      IF(IUNITS.EQ.1)THEN
           WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
           ENDIF
      ELSE
           WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
           ENDIF
      ENDIF
   10 READ(IHISIN,*,END=40)STSEND
```

```
            MIN=MAXIT
            MAX=0
            STSRNG=STSEND-STSBGN
C
C*** CHECK FOR STSRNG=0. IF TRUE, READ ANOTHER REVERSAL.
C
            IF(STSRNG.EQ.0.0D0) GOTO 10
            NUMSTP=NUMINC
            STSSTP=STSRNG/DBLE(NUMSTP)
            IF(ABS(STSSTP).LT.STSINC)THEN
                 NUMSTP=INT(ABS(STSRNG/STSINC))+1
                 STSSTP=STSRNG/DBLE(NUMSTP)
            ENDIF
            DMSTS=STSSTP
            DMCON=CONERR*ABS(DMSTS)
            DT=ABS(STSSTP)/RATELD
            ELSCON=ABS(DMSTS)/E
            Z1=AN6*DT
            Z3=Z1*YM1
            Z4=E*DT
            Z7=DT*YN1
            Z8=DMSTS/E
            IF(DMSTS.GE.0.0)THEN
                 DTSTN=0.001
            ELSE
                 DTSTN=-0.001
            ENDIF
            DO 20, II=1,NUMSTP
                 DPSTN=DTSTN
                 DCSTN=ABS(DPSTN)
                 DESTS=DMSTS
                 BM1STS=BM2STS
                 P1STN=P2STN
                 C1STN=C2STN
                 E1STS=E2STS
                 T1STN=T2STN
                 T1=T2
                 CALL STSCON(BM1STS,P1STN,C1STN,E1STS,T1STN,T1,DT,DTSTN,
     $                       DMSTS,DPSTN,DCSTN,DESTS,BM2STS,P2STN,C2STN,
     $                       E2STS,T2STN,T2,IERR,M)
                 MAVG=MAVG+M
                 IF(M.GT.MAX)MAX=M
                 IF(M.LT.MIN)MIN=M
                 IF(NAMGPH.NE.' ')THEN
                      IF(IUNITS.EQ.1)THEN
                           WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
                      ELSE
                           WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
                      ENDIF
                 ENDIF
```

```
                IF(IERR.NE.0)GOTO 30
   20         CONTINUE
   30         CONTINUE
              I=I+1
              MAVG=MAVG/NUMSTP
              IF(IUNITS.EQ.1)THEN
                   WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                        MAX,IERR
              ELSE
                   WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                        MAX,IERR
              ENDIF
              IF(IERR.NE.0)GOTO 40
              STSBGN=BM2STS
         GOTO 10
   40    CONTINUE
         RETURN
 1000    FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.3,1X,4(I4,1X),I2)
 2000    FORMAT(1X,D20.13,1X,F10.7,1X,F8.3)
 3000    FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.2,1X,4(I4,1X),I2)
 4000    FORMAT(1X,D20.13,1X,F10.7,1X,F8.2)
         END
C=======================================================================
         SUBROUTINE STSLEN(IHISIN,NUMFIL,NAMGPH)
         IMPLICIT DOUBLE PRECISION (A-H,O-Z)
         CHARACTER*(*) NAMGPH
         DIMENSION NUMFIL(2)
         COMMON /WALCON/EOSTS,P2,P3,P4,P5,P6,E,P8,P9,P10,YM1,P12,YN1,P14
         COMMON /GEO/AP2,AP4,RATELD,AP8,IAP1,IUNITS,IAP6,IAP7
         COMMON /STPCON/IBP1,IBP2,BP3,STSINC,BP5
         COMMON /DTCONS/Z1,Z3,Z4,CP4,Z7,AN6,Z8
         COMMON /CONVER/IDP1,MAXIT,CONERR,ELSCON,DP5,DMCON
C FOLLOWING IS FOR STRESS CONTROL
C TIME, T=0 INITIALIZATION
         MAVG=0
         REWIND(IHISIN)
         T2=0.0D0
         BM2STS=0.0D0
         T2STN=0.0D0
         P2STN=0.0D0
         C2STN=0.0D0
         E2STS=EOSTS
         STSBGN=0.0D0
         IERR=0
         M=0
         I=0
         IF(IUNITS.EQ.1)THEN
              WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,M,IERR
              IF(NAMGPH.NE.' ')THEN
                   WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
```

```
            ENDIF
         ELSE
            WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,M,IERR
            IF(NAMGPH.NE.' ')THEN
               WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
            ENDIF
         ENDIF
      10 READ(IHISIN,*,END=40)STSEND
            MIN=MAXIT
            MAX=0
            STSRNG=STSEND-STSBGN
C
C*** CHECK FOR STSRNG=0. IF TRUE, READ ANOTHER REVERSAL.
C
            IF(STSRNG.EQ.0.0D0) GOTO 10
            NUMSTP=INT(ABS(STSRNG/STSINC))+1
            STSSTP=STSRNG/DBLE(NUMSTP)
            DMSTS=STSSTP
            DMCON=CONERR*ABS(DMSTS)
            DT=ABS(STSSTP)/RATELD
            ELSCON=ABS(DMSTS)/E
            Z1=AN6*DT
            Z3=Z1*YM1
            Z4=E*DT
            Z7=DT*YN1
            Z8=DMSTS/E
            IF(DMSTS.GE.0)THEN
               DTSTN=0.001
            ELSE
               DTSTN=-0.001
            ENDIF
            DO 20, II=1,NUMSTP
               DESTS=DMSTS
               DPSTN=DTSTN
               DCSTN=ABS(DPSTN)
               BM1STS=BM2STS
               P1STN=P2STN
               C1STN=C2STN
               E1STS=E2STS
               T1STN=T2STN
               T1=T2
               CALL STSCON(BM1STS,P1STN,C1STN,E1STS,T1STN,T1,DT,DTSTN,
     $                     DMSTS,DPSTN,DCSTN,DESTS,BM2STS,P2STN,C2STN,
     $                     E2STS,T2STN,T2,IERR,M)
               MAVG=MAVG+M
               IF(M.GT.MAX) MAX=M
               IF(M.LT.MIN) MIN=M
               IF(NAMGPH.NE.' ')THEN
                  IF(IUNITS.EQ.1)THEN
                     WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS
```

```
                    ELSE
                         WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS
                    ENDIF
               ENDIF
               IF(IERR.NE.0)GOTO 30
   20      CONTINUE
   30      CONTINUE
           I=I+1
           MAVG=MAVG/NUMSTP
           IF(IUNITS.EQ.1)THEN
               WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                    MAX,IERR
           ELSE
               WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                    MAX,IERR
           ENDIF
           IF(IERR.NE.0)GOTO 40
           STSBGN=BM2STS
       GOTO 10
   40  CONTINUE
       RETURN
 1000  FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.3,1X,4(I4,1X),I2)
 2000  FORMAT(1X,D20.13,1X,F10.7,1X,F8.3)
 3000  FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.2,1X,4(I4,1X),I2)
 4000  FORMAT(1X,D20.13,1X,F10.7,1X,F8.2)
       END
C=======================================================================
       SUBROUTINE STSCON(BM1STS,P1STN,C1STN,E1STS,T1STN,T1,DT,DTSTN,
      $DMSTS,DPSTN,DCSTN,DESTS,BM2STS,P2STN,C2STN,
      $E2STS,T2STN,T2,IERR,M)
C***********************************************************************
C FOLLOWING INPUT ONLY TO SUBROUTINE
C   FORTRAN CONSTANTS: MINIT,MAXIT,CONERR
C   WALKER CONSTANTS: EOSTS,AN,AM,AN1,AN2,AN3,AN4,AN5,AN6,AN7,AK1,
C                     AK2,E
C   VARIABLES: BM1STS,P1STN,C1STN,E1STS,DT,DTSTN
C FOLLOWING INPUT TO AND OUTPUT FROM SUBROUTINE
C   VARIABLES: DPSTN
C FOLLOWING OUTPUT ONLY FROM SUBROUTINE
C   VARIABLES: BM2STS,P2STN,C2STN,E2STS
C***********************************************************************
       IMPLICIT DOUBLE PRECISION(A-H,O-Z)
       COMMON /CONVER/MINIT,MAXIT,CONERR,ELSCON,DCCON,DMCON
       COMMON /WALCON/EOSTS,AN1,AN2,AN3,AN4,AK1,E,Y5,Y7,YK2,YM1,YM2,
      $               YN1,YN2
       COMMON /DTCONS/Z1,Z3,Z4,Z5,Z7,AN6,Z8
C
C*** CHECK FOR DT=0.0 WHICH WOULD RESULT IN DIVISION BY ZERO
C
       IF(DT.EQ.0.0D0)THEN
```

```
            WRITE(*,*)'0 ERROR DT=0'
            WRITE(*,*)'  SUBROUTINE STSCON: CHECK DT CALCULATION IN'
            WRITE(*,*)'                     CALLING PROGRAM'
            IERR=70
            RETURN
         ENDIF
         Z2=E1STS-(E0STS+AN1*P1STN)
         Z6=BM1STS-E1STS
         C2STN=DCSTN+C1STN
         E2STS=DESTS+E1STS
         P2STN=DPSTN+P1STN
         BM2STS=DMSTS+BM1STS
         M=0
      10 M=M+1
C
C***    CHECK FOR EXCEEDING ALLOWED MAXIMUM NUMBER OF ITERATIONS AND
C***       A: FLAG NONCONVERGENCE OF A MOSTLY ELASTIC STRAIN INCREMENT
C***       B: OR FLAG MOSTLY A PLASTIC INCREMENT WITH NONCONVERGENCE
C
         IF(M.GT.MAXIT)THEN
            M=M-1
            IF(DCSTN.LT.ELSCON)THEN
               IERR=1+IERR
               GOTO 7000
            ELSE
               IERR=2+IERR
               GOTO 7000
            ENDIF
         ENDIF
         OLDDE=DESTS
         OLDDC=DCSTN
         OLDDT=DTSTN
         X1=(AN3+AN4*EXP(Y5*C2STN))*DCSTN
         DG=X1+Z1*ABS(E2STS)**YM1
         IF(DG.NE.0.0D0)THEN
            X2=EXP(-1.0D0*DG)
            X3=AN2*DPSTN
            X4=X3*(1.0D0-X2)/DG
            X5=Z2*X2
            X6=E0STS+AN1*P2STN
            E2STS=X6+X4+X5
            X7=ABS(E2STS)
            X8=X7**YM2
            F1DG=X1+Z1*X8*X7-DG
            IF(E2STS.GE.0.0D0)THEN
               DF1DDG=Z3*X8*((X3*X2-X4)/DG-X5)-1.0D0
            ELSE
               DF1DDG=Z3*X8*((X4-X3*X2)/DG+X5)-1.0D0
            ENDIF
            DG=DG-F1DG/DF1DDG
```

```
            X2=EXP(-1.0D0*DG)
            E2STS=X6+X3*(1.0D0-X2)/DG+Z2*X2
            DESTS=E2STS-E1STS
      ELSE
            X3=AN2*DPSTN
            E2STS=X3+E1STS
            DESTS=X3
      ENDIF
      BK2=AK1+YK2*EXP(Y7*C2STN)
      DQ=Z4/BK2*(DCSTN/DT)**YN1
      IF(DQ.NE.0.0D0)THEN
            X9=EXP(-1.0D0*DQ)
            X11=Z6*X9
            X16=(1.0D0-X9)/DQ
            X17=BM2STS-E2STS
            DTSTN=((X17-X11)/X16+DESTS)/E
            DPSTN=DTSTN-Z8
            DCSTN=ABS(DPSTN)
            C2STN=DCSTN+C1STN
            BK2=AK1+YK2*EXP(Y7*C2STN)
            X12=DCSTN/DT
            X13=X12**YN2
            F2DQ=DQ-Z4/BK2*X13*X12
            IF(DCSTN.GE.0.0D0)THEN
                  DF2DDQ=1.0D0-Z7/BK2*X13*((E*DTSTN-DESTS)/DQ*(X16-X9)+
     $                  X11)
            ELSE
                  DF2DDQ=1.0D0+Z7/BK2*X13*((E*DTSTN-DESTS)/DQ*(X16-X9)+
     $                  X11)
            ENDIF
            DQ=DQ-F2DQ/DF2DDQ
            X9=EXP(-1.0D0*DQ)
            DTSTN=(DQ/(1.0D0-X9)*(X17-Z6*X9)+DESTS)/E
            DPSTN=DTSTN-Z8
            DCSTN=ABS(DPSTN)
            C2STN=DCSTN+C1STN
            P2STN=DPSTN+P1STN
      ELSE
            DTSTN=Z8
            DPSTN=0.0D0
            DCSTN=0.0D0
            C2STN=C1STN
            P2STN=P1STN
      ENDIF
      T2STN=T1STN+DTSTN
C
C*** CONVERGENCE CHECKS:
C
C***  1: FORCE A MINIMUM NUMBER OF ITERATIONS
C
```

```
      IF(M.LT.MINIT) GOTO 10
C
C***  6: CHECK CONVERGENCE OF THE TOTAL STRAIN.
C
      IF(T2STN.EQ.0.0D0)THEN
           IF(ABS(OLDDT-DTSTN).GT.CONERR)THEN
                IERR=60
                GOTO 10
           ENDIF
      ELSE
           IF(ABS((OLDDT-DTSTN)/T2STN).GT.CONERR)THEN
                IERR=60
                GOTO 10
           ENDIF
      ENDIF
C
C***  4: CHECK CONVERGENCE OF CUMULATIVE PLASTIC STRAIN INCREMENT AND
C***     THEREFORE ALSO THE PLASTIC STRAIN INCREMENT. THIS CHECK IS
C***     SKIPPED IF THE PLASTIC INCREMENT MAGNITUDE IS BELOW A
C***     PERCENTAGE OF THE MAGNITUDE OF THE TOTAL STRAIN INCREMENT.
C
      IF(DCSTN.GE.ABS(DTSTN)*CONERR)THEN
           IF(P2STN.EQ.0.0D0)THEN
                IF(ABS(OLDDC-DCSTN).GT.CONERR) THEN
                     IERR=40
                     GOTO 10
                ENDIF
           ELSE
                IF(ABS((OLDDC-DCSTN)/P2STN).GT.CONERR)THEN
                     IERR=40
                     GOTO 10
                ENDIF
           ENDIF
      ENDIF
C
C***  5: CHECK CONVERGENCE OF THE EQUILIBRIUM STRESS. DONE ONLY WHEN THE
C***     MAGNITUDE OF THE EQUILIBRIUM STRESS EXCEEDS A PERCENTAGE OF THE
C***     MAGNITUDE OF THE MECHANICAL STRESS.
C
      IF(ABS(DESTS).GE.DMCON)THEN
           IF(E2STS.EQ.0.0D0)THEN
                IF(ABS(OLDDE-DESTS).GT.CONERR)THEN
                     IERR=50
                     GOTO 10
                ENDIF
           ELSE
                IF(ABS((OLDDE-DESTS)/E2STS).GT.CONERR)THEN
                     IERR=50
                     GOTO 10
                ENDIF
```

```
            ENDIF
         ENDIF
C
C*** CONVERGENCE SATISFIED
C
         IERR=0
C
C*** STEP #35: ITERATION COMPLETE UPDATE STATE VARIABLES AND PRINT OUT
C***           RESULTS.
C
C*** STEP #36. GO TO THE NEXT TIME INCREMENT.
C
 7000 T2=T1+DT
      RETURN
      END
C=======================================================================
      SUBROUTINE NEUBER(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /STPCON/IALTER,IPAD1,PAD2,PAD3,PAD4
      IF(IALTER.LE.2)THEN
           CALL NEUNUM(IHISIN,NUMFIL(1),NAMGPH)
      ELSE
           CALL NEULEN(IHISIN,NUMFIL(1),NAMGPH)
      ENDIF
      RETURN
      END
C=======================================================================
      SUBROUTINE NEUNUM(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /WALCON/EOSTS,P2,P3,P4,P5,P6,E,P8,P9,P10,YM1,P12,YN1,P14
      COMMON /GEO/AKT,AP4,RATELD,AP8,IAP1,IUNITS,IAP6,IAP7
      COMMON /STPCON/IBP1,NUMINC,BP3,BP4,STSINC
      COMMON /DTCONS/Z1,Z3,Z4,CP4,Z7,AN6,CP7
      COMMON /CONVER/IDP1,MAXIT,DP3,DP4,DP5,DP6
C TIME, T=0 INITIALIZATION
      MAVG=0
      REWIND(IHISIN)
      T2=0.0D0
      BM2STS=0.0D0
      T2STN=0.0D0
      P2STN=0.0D0
      C2STN=0.0D0
      E2STS=EOSTS
      DESTS=0.0D0
      SBGN=0.0D0
      IERR=0
```

```
      M=0
      I=0
      SSTPP=0.0D0
      DTSTNP=0.0D0
      DMSTSP=0.0D0
      STNBGN=0.0D0
      STSBGN=0.0D0
      IF(IUNITS.EQ.1)THEN
           WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS,SSTPP+SBGN
           ENDIF
      ELSE
           WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS,SSTPP+SBGN
           ENDIF
      ENDIF
   10 READ(IHISIN,*,END=40)SEND
           MIN=MAXIT
           MAX=0
           SRNG=SEND-SBGN
C
C*** CHECK FOR SRNG=0. IF TRUE, READ ANOTHER REVERSAL.
C
           IF(SRNG.EQ.0.0D0) GOTO 10
           NUMSTP=NUMINC
           SSTP=SRNG/DBLE(NUMSTP)
           IF(ABS(SSTP).LT.STSINC/AKT)THEN
                NUMSTP=INT(ABS(SRNG/(STSINC/AKT)))+1
                SSTP=SRNG/DBLE(NUMSTP)
           ENDIF
           IF(SRNG.LT.0.0D0)THEN
                ISGNTC=-1
           ELSE
                ISGNTC=1
           ENDIF
           DT=ABS(SSTP/RATELD)
           Z1=AN6*DT
           Z3=Z1*YM1
           Z4=E*DT
           Z7=DT*YN1
           DSSDSN=(AKT*(SSTPP+SSTP))**2/E
           DTSTN=AKT*SSTP/E
           DO 20, II=1,NUMSTP
                DMSTS=0.0D0
                DESTS=E*DTSTN
                BM1STS=BM2STS
                P1STN=P2STN
                C1STN=C2STN
```

```
                E1STS=E2STS
                T1STN=T2STN
                T1=T2
                CALL NEUSTN(ISGNTC,DSSDSN,BM1STS,P1STN,C1STN,E1STS,
     $                      T1STN,T1,DT,DTSTNP,DMSTSP,DTSTN,DMSTS,DESTS,
     $                      BM2STS,P2STN,C2STN,E2STS,T2STN,T2,IERR,M)
                MAVG=MAVG+M
                IF(M.GT.MAX) MAX=M
                IF(M.LT.MIN) MIN=M
                SSTPP=SSTPP+SSTP
                IF(NAMGPH.NE.' ')THEN
                     IF(IUNITS.EQ.1)THEN
                         WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS,SSTPP+SBGN
                     ELSE
                         WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS,SSTPP+SBGN
                     ENDIF
                ENDIF
                IF(IERR.NE.0)GOTO 30
                DSSDSN=(AKT*(SSTPP+SSTP))**2/E
                DTSTNP=T2STN-STNBGN
                DMSTSP=BM2STS-STSBGN
   20        CONTINUE
   30        CONTINUE
             I=I+1
             MAVG=MAVG/NUMSTP
             IF(IUNITS.EQ.1)THEN
                  WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                     MAX,IERR
             ELSE
                  WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                                     MAX,IERR
             ENDIF
             IF(IERR.NE.0)GOTO 40
             STNBGN=T2STN
             STSBGN=BM2STS
             SBGN=SSTPP+SBGN
             SSTPP=0.0D0
             DTSTNP=0.0D0
             DMSTSP=0.0D0
          GOTO 10
   40  CONTINUE
          RETURN
 1000  FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.3,1X,4(I4,1X),I2)
 2000  FORMAT(1X,D20.13,1X,F10.7,2(1X,F8.3))
 3000  FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.2,1X,4(I4,1X),I2)
 4000  FORMAT(1X,D20.13,1X,F10.7,2(1X,F8.2))
        END
C=======================================================================
      SUBROUTINE NEULEN(IHISIN,NUMFIL,NAMGPH)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
```

```
      CHARACTER*(*) NAMGPH
      DIMENSION NUMFIL(2)
      COMMON /WALCON/EOSTS,P2,P3,P4,P5,P6,E,P8,P9,P10,YM1,P12,YN1,P14
      COMMON /GEO/AKT,AP4,RATELD,AP8,IAP1,IUNITS,IAP6,IAP7
      COMMON /STPCON/IBP1,IBP2,BP3,BP4,STSINC
      COMMON /DTCONS/Z1,Z3,Z4,CP4,Z7,AN6,CP7
      COMMON /CONVER/IDP1,MAXIT,DP3,DP4,DP5,DP6
C TIME, T=0 INITIALIZATION
      MAVG=0
      REWIND(IHISIN)
      T2=0.0D0
      BM2STS=0.0D0
      T2STN=0.0D0
      P2STN=0.0D0
      C2STN=0.0D0
      E2STS=EOSTS
      SBGN=0.0D0
      DESTS=0.0D0
      IERR=0
      M=0
      I=0
      SSTPP=0.0D0
      DTSTNP=0.0D0
      DMSTSP=0.0D0
      STNBGN=0.0D0
      STSBGN=0.0D0
      IF(IUNITS.EQ.1)THEN
           WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS,SSTPP+SBGN
           ENDIF
      ELSE
           WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,M,IERR
           IF(NAMGPH.NE.' ')THEN
                WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS,SSTPP+SBGN
           ENDIF
      ENDIF
   10 READ(IHISIN,*,END=40)SEND
           MIN=MAXIT
           MAX=0
           SRNG=SEND-SBGN
C
C*** CHECK FOR SRNG=0. IF TRUE, READ ANOTHER REVERSAL.
C
           IF(SRNG.EQ.0.0D0) GOTO 10
           NUMSTP=INT(ABS(SRNG/(STSINC/AKT)))+1
           SSTP=SRNG/DBLE(NUMSTP)
           IF(SRNG.LT.0.0D0)THEN
                ISGNTC=-1
           ELSE
```

```
            ISGNTC=1
          ENDIF
          DT=ABS(SSTP/RATELD)
          Z1=AN6*DT
          Z3=Z1*YM1
          Z4=E*DT
          Z7=DT*YN1
          DSSDSN=(AKT*(SSTPP+SSTP))**2/E
          DTSTN=AKT*SSTP/E
          DO 20, II=1,NUMSTP
               DESTS=E*DTSTN
               DMSTS=0.0D0
               BM1STS=BM2STS
               P1STN=P2STN
               C1STN=C2STN
               E1STS=E2STS
               T1STN=T2STN
               T1=T2
               CALL NEUSTN(ISGNTC,DSSDSN,BM1STS,P1STN,C1STN,E1STS,
     $                    T1STN,T1,DT,DTSTNP,DMSTSP,DTSTN,DMSTS,DESTS,
     $                    BM2STS,P2STN,C2STN,E2STS,T2STN,T2,IERR,M)
               MAVG=MAVG+M
               IF(M.GT.MAX) MAX=M
               IF(M.LT.MIN) MIN=M
               SSTPP=SSTPP+SSTP
               IF(NAMGPH.NE.' ')THEN
                    IF(IUNITS.EQ.1)THEN
                        WRITE(NUMFIL(2),2000)T2,T2STN,BM2STS,SSTPP+SBGN
                    ELSE
                        WRITE(NUMFIL(2),4000)T2,T2STN,BM2STS,SSTPP+SBGN
                    ENDIF
               ENDIF
               IF(IERR.NE.0)GOTO 30
               DSSDSN=(AKT*(SSTPP+SSTP))**2/E
               DTSTNP=T2STN-STNBGN
               DMSTSP=BM2STS-STSBGN
20        CONTINUE
30        CONTINUE
          I=I+1
          MAVG=MAVG/NUMSTP
          IF(IUNITS.EQ.1)THEN
               WRITE(NUMFIL(1),1000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                              MAX,IERR
          ELSE
               WRITE(NUMFIL(1),3000)I,T2,T2STN,BM2STS,NUMSTP,MIN,MAVG,
     $                              MAX,IERR
          ENDIF
          IF(IERR.NE.0)GOTO 40
          STNBGN=T2STN
          STSBGN=BM2STS
```

```
            SBGN=SSTPP+SBGN
            SSTPP=0.0D0
            DTSTNP=0.0D0
            DMSTSP=0.0D0
         GOTO 10
      40 CONTINUE
         RETURN
    1000 FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.3,1X,4(I4,1X),I2)
    2000 FORMAT(1X,D20.13,1X,F10.7,2(1X,F8.3))
    3000 FORMAT(1X,I10,1X,D20.13,1X,F10.7,1X,F8.2,1X,4(I4,1X),I2)
    4000 FORMAT(1X,D20.13,1X,F10.7,2(1X,F8.2))
         END
C=======================================================================
      SUBROUTINE NEUSTN(ISGNTC,DSSDSN,BM1STS,P1STN,C1STN,E1STS,T1STN,T1,
     $                  DT,DTSTNP,DMSTSP,DTSTN,DMSTS,DESTS,
     $                  BM2STS,P2STN,C2STN,E2STS,T2STN,T2,IERR,M)
C***********************************************************************
C FOLLOWING INPUT ONLY TO SUBROUTINE
C   FORTRAN CONSTANTS: MINIT,MAXIT,CONERR
C   WALKER CONSTANTS: EOSTS,AN,AM,AN1,AN2,AN3,AN4,AN5,AN6,AN7,AK1,
C                     AK2,E
C   VARIABLES: BM1STS,P1STN,C1STN,E1STS,DT,DTSTN
C FOLLOWING INPUT TO AND OUTPUT FROM SUBROUTINE
C   VARIABLES: DPSTN
C FOLLOWING OUTPUT ONLY FROM SUBROUTINE
C   VARIABLES: BM2STS,P2STN,C2STN,E2STS
C***********************************************************************
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON /CONVER/MINIT,MAXIT,CONERR,PAD1,PAD2,PAD3
      COMMON /WALCON/EOSTS,AN1,AN2,AN3,AN4,AK1,E,Y5,Y7,YK2,YM1,YM2,
     $               YN1,YN2
      COMMON /DTCONS/Z1,Z3,Z4,PAD4,Z7,AN6,PAD5
C
C*** CHECK FOR DT=0.0 WHICH WOULD RESULT IN DIVISION BY ZERO
C
      IF(DT.EQ.0.0D0)THEN
           WRITE(*,*)'0 ERROR DT=0'
           WRITE(*,*)'  SUBROUTINE NEUSTN: CHECK DT CALCULATION IN'
           WRITE(*,*)'                     CALLING PROGRAM'
           IERR=70
           RETURN
      ENDIF
C
C*** THE STEP NUMBERING APPROXIMATES THAT GIVEN IN WALKER'S PAPER.
C
C*** INITIALIZE SUBROUTINE:
C***  STEP #3: PLASTIC STRAIN GUESS AT END OF INCREMENT.
C***  STEP #4: INITIAL GUESS FOR EQUILIBRIUM STRESS AT END OF INCREMENT.
C
      DPSTN=DTSTN-DMSTS/E
```

```
      DCSTN=ABS(DPSTN)
      Z2=E1STS-(EOSTS+AN1*P1STN)
      Z6=BM1STS-E1STS
      P2STN=P1STN+DPSTN
      E2STS=E1STS+DESTS
C
C*** STEP #7: START OF INNER ITERATION LOOP. M ITERATION LOOP INDEX.
C***          INITIALIZE CONTROL PARAMETER FOR LOOP.
C
      M=0
C
C*** STEP #8: START OF REPETITIVE STATEMENTS OF LOOP.
C
    8 M=M+1
C
C*** STORE VALUES FOR CONVERGENCE TEST
C
      OLDDTS=DTSTN
      OLDDE=DESTS
      OLDDC=DCSTN
      OLDDMS=DMSTS
C
C*** STEP #9:  COMPUTE THE CUMULATIVE PLASTIC STRAIN AT THE END OF THE
C***           INCREMENT.
C*** STEP #10: CALCULATE DRAG STRESS (ISOTROPIC HARDENING) AT END OF
C***           INCREMENT.
C*** STEP #11: EVALUATE THE GUESS FOR DELTA G.
C***           X1-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #13: COMPUTE EQUILIBRIUM STRESS GUESS AT END OF INCREMENT.
C***           X2,X3,X4,X5,X6-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #16: COMPUTE FUNCTION OF DELTA G TO ITERATE  FOR FINDING ITS
C***           ZERO TO IMPROVE THE ESTIMATE OF DELTA G.
C***           X7,X8-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C
      C2STN=C1STN+DCSTN
      BK2=AK1+YK2*EXP(Y7*C2STN)
      X1=(AN3+AN4*EXP(Y5*C2STN))*DCSTN
      DG=X1+Z1*ABS(E2STS)**YM1
      X2=EXP(-1.0D0*DG)
      X3=AN2*DPSTN
      X5=X2*Z2
      X6=EOSTS+AN1*P2STN
      IF(DG.NE.0.0D0)THEN
           X4=X3*(1.0D0-X2)/DG
           E2STS=X6+X5+X4
           X7=ABS(E2STS)
           X8=X7**YM2
           F1DG=X1+Z1*X8*X7-DG
C
C*** STEP #17: COMPUTE THE DERIVATIVE OF THE PROCEEDING FUNCTION
```

```
C
            IF(E2STS.GE.0.0D0)THEN
                 DRF1DG=Z3*X8*((X2*X3-X4)/DG-X5)-1.0D0
            ELSE
                 DRF1DG=Z3*X8*(X5+(X4-X2*X3)/DG)-1.0D0
            ENDIF
C
C*** STEP  #18: REFINE DELTA G BY A SINGLE NEWTON ITERATION.
C*** STEP  #19: COMPUTE REFINED VALUE OF THE EQUILIBRIUM STRESS BY
C***            DIRECT SUBSTITUTION ITERATION, SUBSTITUTING REFINED
C***            DELTA G.
C***            X2-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #19A: COMPUTE THE EQUILIBRIUM STRESS INCREMENT
C*** STEP  #20: COMPUTE GUESS FOR DELTA Q.
C*** STEP  #21: COMPUTE GUESS FOR STRESS AT END OF INCREMENT. TERMED THE
C***            MECHANICAL STRESS AT THE END OF THE TIME INCREMENT,
C***            BM2STS.
C***            X9,X10,X11-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #21A: COMPUTE STRESS INCREMENT.
C*** STEP  #22: REFINE THE PLASTIC STRAIN INCREMENT.
C*** STEP  #24: REFINE THE CUMULATIVE PLASTIC STRAIN INCREMENT.
C*** STEP #24A: COMPUTE REFINED PLASTIC STRAIN VALUE AT THE END OF THE
C***            INCREMENT.
C*** STEP #24B: COMPUTE REFINED DRAG STRESS.
C
            DG=DG-F1DG/DRF1DG
            X2=EXP(-1.0D0*DG)
            E2STS=X6+X3*(1.0D0-X2)/DG+X2*Z2
       ELSE
            E2STS=X3+E1STS
       ENDIF
       DESTS=E2STS-E1STS
       DQ=Z4/BK2*(DCSTN/DT)**YN1
       X9=EXP(-1.0D0*DQ)
       X10=E*DTSTN-DESTS
       X11=Z6*X9
       X14=(1.0D0-X9)/DQ
       IF(DQ.NE.0.0D0)THEN
            BM2STS=E2STS+X11+X10*X14
            DMSTS=BM2STS-BM1STS
            IF(ISGNTC.GE.0.0D0)THEN
                 IF(DMSTS.LT.0.0D0)THEN
                      DMSTS=ABS(DMSTS)
                 ENDIF
            ELSE
                 IF(DMSTS.GT.0.0D0)THEN
                      DMSTS=-DMSTS
                 ENDIF
            ENDIF
            DTSTN0=DTSTN
```

```
            DTSTN=(DTSTN+(DSSDSN-(DMSTS+DMSTSP)*(DTSTN+DTSTNP))/(E*
     $           (DTSTN+DTSTNP)*X14+(DMSTS+DMSTSP)))
            IF(ISGNTC.GE.0.0D0)THEN
                 IF(DTSTN.LT.0.0D0)THEN
                      DTSTN=0.5D0*ABS(DTSTN0)
                 ENDIF
            ELSE
                 IF(DTSTN.GT.0.0D0)THEN
                      DTSTN=-0.5D0*ABS(DTSTN0)
                 ENDIF
            ENDIF
            DPSTN=DTSTN-DMSTS/E
            DCSTN=ABS(DPSTN)
            C2STN=DCSTN+C1STN
            BK2=AK1+YK2*EXP(Y7*C2STN)
C
C*** STEP  #25: COMPUTE THE FUNCTION OF DELTA Q TO ITERATE TO DETERMINE
C***            ITS ZERO SO AS TO REFINE DELTA Q IN THE PROCESS.
C***            X12,X13-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C
            X12=DCSTN/DT
            X13=X12**YN2
            F2DQ=DQ-Z4/BK2*X12*X13
C
C*** STEP #26. COMPUTE THE DERIVATIVE OF THE PROCEEDING FUNCTION.
C
            IF(DPSTN.GE.0.0)THEN
                 DRF2DQ=1.0D0+X13*Z7/BK2*(X10*((DQ+1.0D0)*X9-1.0D0)/DQ/
     $                DQ-X11)
            ELSE
                 DRF2DQ=1.0D0+X13*Z7/BK2*(X11+X10/DQ*(1.0D0-(1.0D0+DQ)*
     $                X9)/DQ)
            ENDIF
C
C*** STEP  #27: REFINE DELTA Q WITH A SINGLE NEWTON ITERATION.
C*** STEP  #28: REFINE THE FUTURE STRESS AT THE END OF THE INCREMENT BY
C***            DIRECT SUBSTITUTION ITERATION, SUBSTITUTING REFINED
C***            DELTA Q.
C***            X9-COMPUTED FOR COMPUTATIONAL EFFICIENCY.
C*** STEP #28A: COMPUTE STRESS INCREMENT.
C*** STEP  #29: REFINE THE PLASTIC STRAIN INCREMENT.
C*** STEP  #30: REFINE THE CUMULATIVE PLASTIC STRAIN INCREMENT.
C*** STEP #30A: COMPUTE THE PLASTIC STRAIN AT THE END OF THE INCREMENT.
C***            COPY OF STEP #12 PLACED HERE FOR COMPUTATIONAL
C***            EFFICIENCY.
C
            DQ=DQ-F2DQ/DRF2DQ
            X9=EXP(-1.0D0*DQ)
            X14=(1.0D0-X9)/DQ
            BM2STS=E2STS+X10*X14+Z6*X9
```

```
      DMSTS=BM2STS-BM1STS
      IF(ISGNTC.GE.0.0D0)THEN
            IF(DMSTS.LT.0.0D0)THEN
                  DMSTS=ABS(DMSTS)
            ENDIF
      ELSE
            IF(DMSTS.GT.0.0D0)THEN
                  DMSTS=-DMSTS
            ENDIF
      ENDIF
      DTSTN0=DTSTN
      DTSTN=(DTSTN+(DSSDSN-(DMSTS+DMSTSP)*(DTSTN+DTSTNP))/(E*
     $       (DTSTN+DTSTNP)*X14+(DMSTS+DMSTSP)))
      IF(ISGNTC.GE.0)THEN
            IF(DTSTN.LT.0.0D0)THEN
                  DTSTN=0.5D0*ABS(DTSTN0)
            ENDIF
      ELSE
            IF(DTSTN.GT.0.0D0)THEN
                  DTSTN=-0.5D0*ABS(DTSTN0)
            ENDIF
      ENDIF
      DPSTN=DTSTN-DMSTS/E
      DCSTN=ABS(DPSTN)
      P2STN=P1STN+DPSTN
 ELSE
      DMSTS=E*DTSTN
      IF(ISGNTC.GE.0.0D0)THEN
            IF(DMSTS.LT.0.0D0)THEN
                  DMSTS=ABS(DMSTS)
            ENDIF
      ELSE
            IF(DMSTS.GT.0.0D0)THEN
                  DMSTS=-DMSTS
            ENDIF
      ENDIF
      DTSTN0=DTSTN
      DTSTN=(DTSTN+(DSSDSN-(DMSTS+DMSTSP)*(DTSTN+DTSTNP))/(E*
     $       (DTSTN+DTSTNP)+(DMSTS+DMSTSP)))
      IF(ISGNTC.GE.0)THEN
            IF(DTSTN.LT.0.0D0)THEN
                  DTSTN=0.5D0*ABS(DTSTN0)
            ENDIF
      ELSE
            IF(DTSTN.GT.0.0D0)THEN
                  DTSTN=-0.5D0*ABS(DTSTN0)
            ENDIF
      ENDIF
      DMSTS=E*DTSTN
      IF(ISGNTC.GE.0.0D0)THEN
```

```
                IF(DMSTS.LT.0.0D0)THEN
                    DMSTS=ABS(DMSTS)
                ENDIF
            ELSE
                IF(DMSTS.GT.0.0D0)THEN
                    DMSTS=-DMSTS
                ENDIF
            ENDIF
            DTSTN0=DTSTN
            DTSTN=(DTSTN+(DSSDSN-(DMSTS+DMSTSP)*(DTSTN+DTSTNP))/(E*
     $           (DTSTN+DTSTNP)+(DMSTS+DMSTSP)))
            IF(ISGNTC.GE.0)THEN
                IF(DTSTN.LT.0.0D0)THEN
                    DTSTN=0.5D0*ABS(DTSTN0)
                ENDIF
            ELSE
                IF(DTSTN.GT.0.0D0)THEN
                    DTSTN=-0.5D0*ABS(DTSTN0)
                ENDIF
            ENDIF
            DPSTN=0.0D0
            DCSTN=0.0D0
            P2STN=P1STN
        ENDIF
        BM2STS=BM1STS+DMSTS
        T2STN=T1STN+DTSTN
C
C*** CONVERGENCE CHECKS:
C
C***  1: FORCE A MINIMUM NUMBER OF ITERATIONS
C
        IF(M.LT.MINIT) GOTO 8
C
C***  2: CHECK FOR EXCEEDING ALLOWED MAXIMUM NUMBER OF ITERATIONS AND
C***    2A: FLAG NONCONVERGENCE OF A MOSTLY ELASTIC STRAIN INCREMENT
C***    2B: OR FLAG MOSTLY A PLASTIC INCREMENT WITH NONCONVERGENCE
C
        IF(M.GE.MAXIT)THEN
            IF(DCSTN.LT.ABS(DMSTS)/E)THEN
                IERR=1+IERR
                GOTO 7000
            ELSE
                IERR=2+IERR
                GOTO 7000
            ENDIF
        ENDIF
C
C***  3: CHECK CONVERGENCE OF THE MECHANICAL STRESS.
C
        IF(BM2STS.EQ.0.0D0)THEN
```

```
            IF(ABS(OLDDMS-DMSTS).GT.CONERR)THEN
                 IERR=30
                 GOTO 8
            ENDIF
       ELSE
            IF(ABS((OLDDMS-DMSTS)/BM2STS).GT.CONERR)THEN
                 IERR=30
                 GOTO 8
            ENDIF
       ENDIF
C
C***  6: CHECK CONVERGENCE OF THE TOTAL STRAIN
C
       IF(T2STN.EQ.0.0D0)THEN
            IF(ABS(OLDDTS-DTSTN).GT.CONERR)THEN
                 IERR=60
                 GOTO 8
            ENDIF
       ELSE
            IF(ABS((OLDDTS-DTSTN)/T2STN).GT.CONERR)THEN
                 IERR=60
                 GOTO 8
            ENDIF
       ENDIF
C
C***  4: CHECK CONVERGENCE OF CUMULATIVE PLASTIC STRAIN INCREMENT AND
C***     THEREFORE ALSO THE PLASTIC STRAIN INCREMENT. THIS CHECK IS
C***     SKIPPED IF THE PLASTIC INCREMENT MAGNITUDE IS BELOW A
C***     PERCENTAGE OF THE MAGNITUDE OF THE TOTAL STRAIN INCREMENT.
C
       IF(DCSTN.GE.ABS(DTSTN)*CONERR)THEN
            IF(P2STN.EQ.0.0D0)THEN
                 IF(ABS(OLDDC-DCSTN).GT.CONERR) THEN
                      IERR=40
                      GOTO 8
                 ENDIF
            ELSE
                 IF(ABS((OLDDC-DCSTN)/P2STN).GT.CONERR)THEN
                      IERR=40
                      GOTO 8
                 ENDIF
            ENDIF
       ENDIF
C
C***  5: CHECK CONVERGENCE OF THE EQUILIBRIUM STRESS. DONE ONLY WHEN THE
C***     MAGNITUDE OF THE EQUILIBRIUM STRESS EXCEEDS A PERCENTAGE OF THE
C***     MAGNITUDE OF THE MECHANICAL STRESS.
C
       IF(ABS(DESTS).GE.ABS(DMSTS)*CONERR)THEN
            IF(E2STS.EQ.0.0D0)THEN
```

```
                IF(ABS(OLDDE-DESTS).GT.CONERR)THEN
                     IERR=50
                     GOTO 8
                ENDIF
           ELSE
                IF(ABS((OLDDE-DESTS)/E2STS).GT.CONERR)THEN
                     IERR=50
                     GOTO 8
                ENDIF
           ENDIF
      ENDIF
C
C*** CONVERGENCE SATISFIED
C
      IERR=0
C
C*** STEP #35: ITERATION COMPLETE UPDATE STATE VARIABLES AND PRINT OUT
C***           RESULTS.
C
      C2STN=C1STN+DCSTN
C     IF((DMSTSP.EQ.0.0D0).AND.(DTSTNP.EQ.0.0D0))THEN
           IF(ISGNTC.GE.0)THEN
                IF(DTSTN.LT.0.0D0)THEN
                     DTSTN=ABS(DTSTN)
                     GOTO 8
                ENDIF
           ELSE
                IF(DTSTN.GT.0.0D0)THEN
                     DTSTN=-1.0D0*DTSTN
                     GOTO 8
                ENDIF
           ENDIF
C     ENDIF
      T2=T1+DT
C
C*** STEP #36. GO TO THE NEXT TIME INCREMENT.
C
      RETURN
 7000 C2STN=C1STN+DCSTN
      T2=T1+DT
      RETURN
      END
C=======================================================================
```

The vita has been removed from the scanned document