5-7-2005

The Influence Of Residual Stress On Fatigue Crack Growth

James E. LaRue

THE INFLUENCE OF RESIDUAL STRESS ON FATIGUE CRACK GROWTH

By

James E. LaRue

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Mechanical Engineering
in the Department of Mechanical Engineering

Mississippi State, Mississippi

May 2005

THE INFLUENCE OF RESIDUAL STRESS ON FATIGUE CRACK GROWTH

By

James E. LaRue

Approved:

S. R. Daniewicz
Professor of Mechanical Engineering
(Major Professor)

S. R. Daniewicz
Graduate Coordinator of the
Department of Mechanical Engineering

M. F. Horstemeyer
Center for Advanced Vehicular Systems Chair
in Computational Solid Mechanics
Professor of Mechanical Engineering
Department of Mechanical Engineering
(Committee Member)

J. C. Newman Jr.
Professor of Aerospace Engineering
(Committee Member)

K. H. Schulz
Dean of the College of Engineering

Name: James E. LaRue

Date of Degree: May 7, 2005

Institution: Mississippi State University

Major Field: Mechanical Engineering

Major Professor: Dr. S. R. Daniewicz

Title of Study: THE INFLUENCE OF RESIDUAL STRESS ON FATIGUE CRACK GROWTH

Pages in Study: 86

Candidate for Degree of Master of Science

This thesis discusses the analysis of fatigue crack growth in the presence of residual stresses to determine a suitable method for fatigue life predictions. In the research discussed herein, the prediction methodologies are compared to determine the most accurate prediction technique. Finite element analysis results are presented as well as laboratory test data. The validity of each methodology is addressed and future work is proposed.

ACKNOWLEDGEMENTS

I would like to extend my utmost gratitude to my advisor, Dr. Steven R. Daniewicz for his patience and support with this research project and for instilling in me a desire for higher education during my undergraduate career. His guidance has proven invaluable to me as I have worked on this research. Thanks is also due to Dr. James C. Newman Jr., for providing invaluable technical support with this research. Expressed appreciation is also due to Dr. Mark F. Horstemeyer for serving as a member of my thesis committee. Also, much appreciation is extended to Dr. Judy Schneider for her valuable help in the testing phase of this research. Finally, I would like to thank Jeffrey Skinner and Dr. Gabriel Portiniche for their valuable assistance with the finite element routines utilized in this research.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE | Page

CHAPTER I

INTRODUCTION AND LITERATURE REVIEW

Overview

Fatigue is the source of at least half of all mechanical failures (Stephens, 2001). The fatigue problem is complex and not fully understood, but it is very important in the design of mechanical systems. Fatigue is especially of interest to the aircraft industry. Many components used in aircraft are fastened together and fastener holes are prevalent. These holes are a source of high stress concentration, and therefore are a potential site for fatigue cracks.

One technique used to enhance the fatigue strength of a fastener hole is to introduce a compressive residual stress field around the hole. An applied load must overcome this residual stress before the crack can grow, thus leading to a longer fatigue life. Although it is widely recognized that compressive residual stress improves fatigue life, in many applications the benefits of compressive residual stresses are not included in the final predicted fatigue life (Ozdemir, 1993). In these cases the residual stress provides added confidence against usage uncertainty, but is not quantified, leading to conservative life predictions. If the residual stress can be included accurately in the fatigue life prediction, the decreased time in part inspection and replacement can be very beneficial to the aircraft industry.

Residual Stress

The effect of residual stress on fatigue crack propagation is of great practical significance and has been the focus of much research. This research has been reviewed in several studies (Nelson, 1982; Parker, 1982; Besuner, 1986; Leis, 1997; SAE, 1997; Stephens, 2001). There are numerous methods of introducing residual stress into mechanical components, including shot peening, interference fit fasteners, low plasticity burnishing, laser shock peening, tensile overloading, and cold expansion. The methods investigated here are tensile overloading and cold expansion.

Tensile overloading occurs when a single tensile load is applied to a component exhibiting a stress gradient, causing plastic deformation and subsequent compressive residual stress. When applied to a hole, a disadvantage of this process is that the residual stress is not uniform around the hole. The tangential residual stress changes from compressive to tensile at different locations around the hole, and this tensile residual stress can be very deleterious if the configuration of the component does not allow the loading to be in the desired direction.

Split sleeve cold expansion is a technique used frequently by the aircraft industry to improve the fatigue performance of structures. The basic split sleeve cold expansion process was developed by the Boeing Company in the late 1960's (Pavier, 1997), and Fatigue Technology Inc. has marketed an efficient method accepted as the standard practice in the United States (FTI, 1991). The process involves radially expanding a hole to create a zone of residual compressive stresses around the hole that then protects it from the effects of cyclic stresses. Using a tapered mandrel fitted with a lubricated sleeve and drawing the mandrel/sleeve combination through the hole using a hydraulic puller produces the radial expansion. The diameter of the mandrel and the sleeve is greater than the starting diameter of the hole. As the mandrel/sleeve is pulled through the hole, the material expands, allowing the mandrel to pass through the hole. The area surrounding the hole is in a subsequent state of compression that protects the hole from fatigue cracking. The function of the disposable split sleeve is to reduce mandrel pull force, ensure correct radial expansion of the hole, preclude damage to the hole, and allow one-sided processing. A finish ream is employed to diminish the effects of the damage to the hole. Unlike the tensile overloading, cold expansion leads to a uniform tangential residual stress around the hole.

Cold expansion has been investigated in numerous studies. Many analytical solutions for computation of the residual stress have been developed, but most have achieved only limited agreement with experimental results (Hsu, 1975; Rich, 1975; Chen, 1986; Zhu, 1987). Researchers have suggested that the poor results are attributed to the analytical models' assumption of two-dimensional plane stress or plane strain (Poussard, 1995; Kang, 2002). Another problem with the analytical models is the exclusion of the reaming process.

To better simulate the cold expansion operation, finite element analyses have been employed. A finite element simulation can include all processes included in cold expansion, including the reaming step. Many two-dimensional simulations have been conducted using either plane stress or plane strain, along with two-dimensional axisymmetric simulations to account for the three-dimensional effects of the mandrel removal. Kang *et al.* (Kang, 2002) have shown that the residual stress is significantly different at different sections through the thickness. Pavier *et al*. (Pavier, 1997) also used a two-dimensional axisymmetric model to simulate the cold expansion process. They concluded that residual stresses can only be estimated accurately by using a realistic simulation of cold expansion. Poussard *et al*. (Poussard, 1995) further illustrated the need for a three-dimensional simulation to account for the through-thickness variation of residual stress.

Another factor in the simulation of cold expansion is the unloading response of the material as the mandrel is pulled through the hole. Several studies have shown that the residual stress predicted in the region of reverse yielding is sensitive to the compressive yielding behavior (Kang, 2002). Neither isotropic nor kinematic hardening models used in finite element simulations adequately account for the Bauschinger effect.

Superposition

Superposition techniques are often used when assessing the effects of a known residual stress field on fatigue crack propagation. The superposition involves the computation of a stress intensity factor $(K)_R$ which is associated with the initial pre-existing residual stress field. This factor it then superposed upon the stress intensity factor that results from external loading $(K)_L$ to give the total resultant stress intensity factor for the maximum and minimum loads:

$$K_{max} = \left(K_{max}\right)_L + \left(K\right)_R \quad (1)$$

$$K_{\min} = (K_{\min})_L + (K)_R \quad (2)$$

The stress intensity factor range and stress ratio is then calculated as the following:

$$\Delta K = K_{\max} - K_{\min} \tag{3}$$

$$R = \frac{K_{\min}}{K_{\max}} \tag{4}$$

The stress intensity factor range does not change since the stress intensity factor from the residual stress is negated, and the stress ratio holds the dependence of the stress intensity factor from the residual stress. Fatigue crack growth is predicted using a correlation of the form:

$$\frac{da}{dN} = f(\Delta K, R) \tag{5}$$

The superposition method described by Equation 5 is widely used, but the dependence of R results in more rigorous calculations. To remove the stress ratio from the function and simplify the calculations for this work, the following superposition method was employed. Maximum and minimum values of the total resultant stress intensity factor K are computed for the cyclic loading, and negative resultant K values are set to zero. A total resultant stress intensity factor range ΔK is then calculated. This resultant stress intensity factor range may then be used to compute the predicted fatigue crack growth rate da/dN in the residual stress field using a correlation of the form:

$$\frac{da}{dN} = f(\Delta K, R = 0) \tag{6}$$

The superposition technique is used extensively because of its simplicity. It has been criticized by some researchers because it considers only the initial residual stress field that exists in the uncracked structure, with no acknowledgement of the redistribution of residual stress that occurs as the propagating fatigue crack penetrates the residual stress field with its free or partially free surfaces (Underwood, 1977; Fukuda, 1978; Chandawanich, 1979; Nelson, 1982; Lam, 1989; Wilks, 1993; Kiciak, 1996; Lee, 1998). Other researchers have argued that the redistribution of residual stress is of no consequence (Parker, 1982; Todoroki, 1991).

Bueckner (Bueckner, 1958) has demonstrated mathematically that, for linear elastic materials, stress intensity factors resulting from a given applied loading may be computed using the stress distribution in the uncracked structure. Heaton (Heaton, 1976) has presented a mathematically rigorous proof that generalizes Bueckner's formulation to include both thermal and residual stress fields. The work of Bueckner and Heaton suggests that, for linear elastic materials, the redistribution of applied and residual stresses due to fatigue crack propagation is of no consequence when computing stress intensity factors and subsequent fatigue crack growth through use of equations (2) or (3).

The conclusions arrived at by Bueckner and Heaton are applicable to linear elastic materials only. The existence of plastic deformation at the crack tip, even under small-scale yielding conditions, will produce crack-generated residual stresses at the crack tip (Rice, 1967; Broek, 1986) and closure along the crack surface of the propagating crack (Elber, 1970; Elber, 1971). The superposition methodology is unable to account for the influence of these effects. Consequently, the use of linear elastic superposition techniques for prediction of the effects of residual stress on fatigue crack propagation may result in crack growth predictions that correlate poorly with experimental observations (Nelson, 1982).

Plasticity-Induced Crack Closure

As an alternative to superposition based fatigue crack growth prediction, the effective stress intensity factor range ΔK_{eff} first introduced by Elber can be used (Elber, 1970; Elber, 1971). The effective stress intensity factor range is employed to enable consideration of crack closure in fatigue crack growth predictions. Elber considers that as a crack propagates, crack closure occurs as a result of plastically deformed material left in the path taken by the crack. This material is referred to as the plastic wake. The plastic wake enables the crack to close before minimum load is reached, and Elber reasoned that the stress intensity factor at the crack tip does not change while the crack is closed even when the applied load is changing. The value of K when the crack is first fully opened is defined as K_o and the reduced range of K due to closure is given by

$$\Delta K_{eff} = K_{\max} - K_o \qquad (7)$$

The effective stress intensity factor range has a relationship with the fatigue crack growth rate of the form:

$$\frac{da}{dN} = g(\Delta K_{eff}) \qquad (8)$$

Since Elber first introduced ΔK_{eff}, several methods have been developed to predict K_o in a given structure. One such method is elastic-plastic finite element analysis, which can be used to study plasticity-induced crack closure by simulating fatigue crack growth as a discrete number of incremental crack extensions. If the initial residual stress is introduced as a distribution of incompatible strain, then this type of analysis can be made to incorporate the effect of initial residual stress as well. The presence of an initial residual stress field may cause significant differences in the crack closure behavior when compared to an analysis containing no initial residual stresses. The initial residual stress and its redistribution as the crack propagates may also result in final crack opening away from the crack tip, potentially complicating the methodology used to predict fatigue crack growth.

A number of researchers have simulated plasticity-induced closure using finite element analyses. McClung (McClung, 1989) and Solanki *et al*. (Solanki, 2003) have provided critical overviews of these analyses. The basic algorithm employed by the studies is the same. An elastic-plastic model is built with a suitably refined mesh, and remote tractions are applied to simulate cyclic loading. The crack tip node is released during each cycle, advancing the crack one element length and allowing a plastic wake to form. Crack closure is predicted by monitoring the contact between crack faces. This process is repeated until the crack opening stress values have stabilized. There are many variables to consider when using finite element to simulate closure: element type, mesh size, crack opening level determination, crack opening level stabilization, crack advance scheme, and constitutive model (Skinner, 2001).

Although many studies have been performed employing finite element analyses to simulate fatigue crack growth and crack closure, few have considered a crack growing through an initial residual stress field. Beghini and Bertini (Beghini, 1990) employed a finite element based technique to study the interaction between residual stress fields and crack closure. They found that the finite element analysis resulted in K_o values that compared fairly well with experimental values. More recently, Choi and Song (Choi,

1995) have also used a finite element analysis to predict crack closure in compressive residual stress fields. They found very good agreement between the finite element predictions and experimental measurements of K_o in a tensile residual stress field, but poorer agreement in a compressive residual stress field.

In the research reported here, fatigue crack growth from a hole under the influence of residual stress is considered. The crack closure level is computed from a finite element simulation and a fatigue life is predicted. This method of fatigue life prediction is compared with a superposition technique and experimental data.

CHAPTER II

LABORATORY TESTING OF FATIGUE CRACK GROWTH

Tensile Overload

Liu (Liu, 1979) measured fatigue crack growth from a hole using the specimen shown in Figure 2.1. Constant amplitude loading was used and a residual stress was induced near the hole using a single tensile overload. The material used was a 2024-T351 aluminum alloy in the LT orientation with t = 6.35 mm, w = 76.2 mm, h = 304.8 mm, and 2r = 19.05 mm. Tensile tests were conducted by Liu to determine the stress strain relationship for the alloy, shown in Figure 2.2. Center-cracked specimens were also tested under constant amplitude cyclic loading to develop baseline crack growth rate (da/dN) curves at R ratios of 0.1 and 0.7. This data is shown in Figure 2.3 in conjunction with the da/dN—ΔK_{eff} curve taken from NASGRO (NASGRO, 2002).

Two specimens with the dimensions shown in Figure 2.1 were tested. These specimens were subjected to one overload cycle σ = 250 MPa to introduce a residual stress field near the hole. An elox cut was then inserted into the edge of the hole. The specimens were then subjected to constant amplitude loading with R = 0.1. The test matrix is shown in Table 2.1.

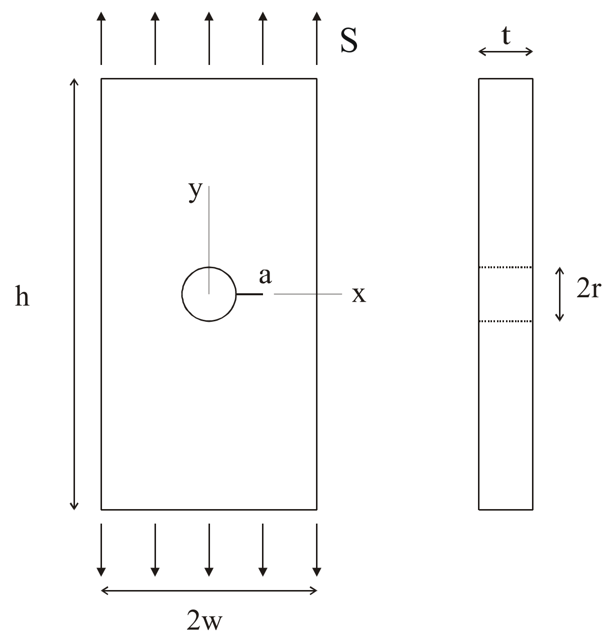

Figure 2-1 – Fatigue Crack Growth Specimen

Table 2-1 – 2024 Test Matrix

Test	Stress level, S	Elox cut length, l_n	Elox cut width, w_n
A2-30	103.4 MPa	1.994 mm	0.152 mm
A2-31	124.1 MPa	1.029 mm	0.152 mm

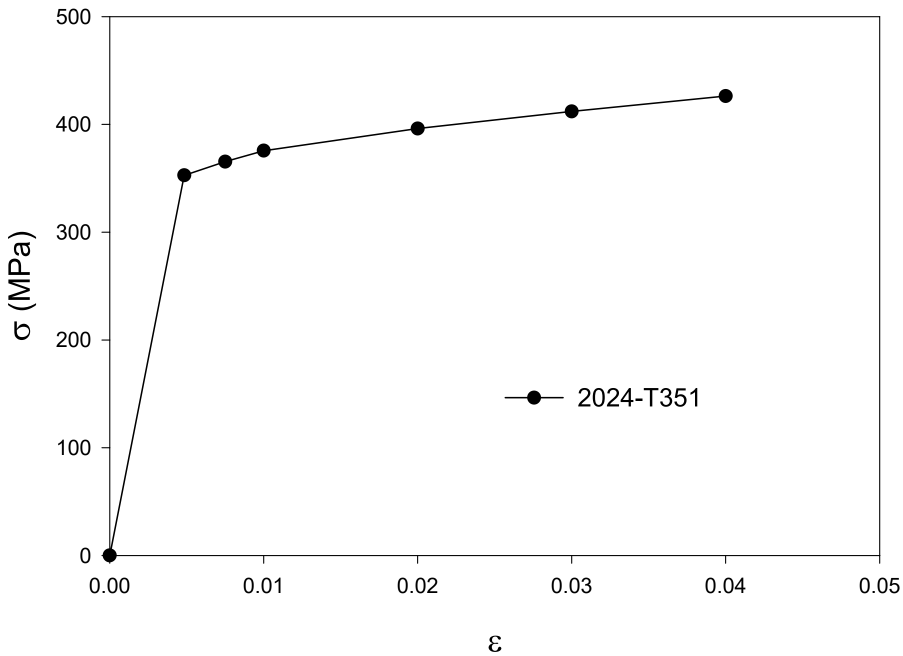

Figure 2-2 – 2024 Stress-Strain Curve

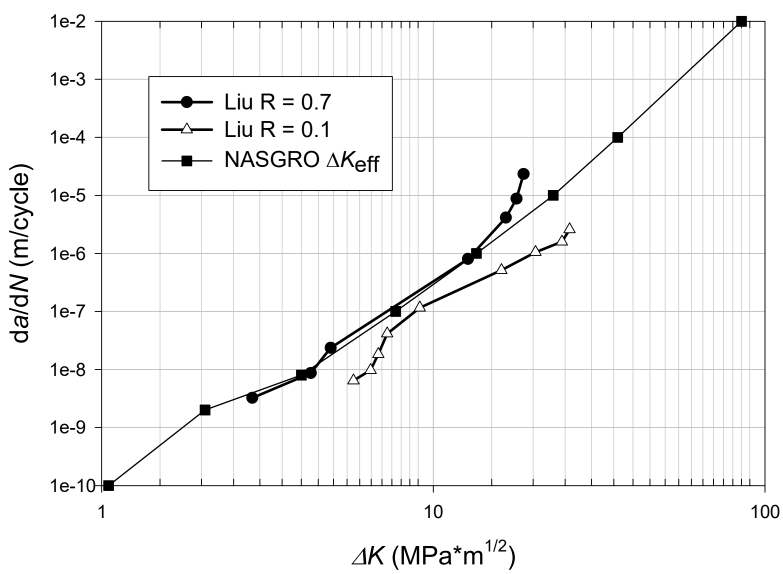

Figure 2-3 – 2024 Baseline Crack Growth Rate Data

Cold Expansion

Fatigue tests were also performed on plates containing a hole using the specimen shown in Figure 2.1. The test plan called for a 7075-T6 aluminum alloy with t = 2.03 mm, w = 22.23 mm, h = 203.2 mm, and $2r$ = 6.35 mm. An electrical-discharge machined (EDM) notch with notch length l_n = 0.254 mm and notch width w_n = 0.127 mm was inserted into one side of each hole to aid in crack initiation. Overall, ten specimens were tested with constant amplitude loading with R = 0.1 and σ_{max} = 117.2 MPa. Six specimens were tested with no residual stress. The test plan initially called for specimens with cold expanded holes with a final $2r$ = 6.35 mm to be tested. The cold expansion process was applied using the following steps:

1. Initial hole cut with $2r$ = 5.979 mm
2. A mandrel with diametrical interference d_i = 0.268 mm, corresponding to 4.48 % cold expansion, was pulled through hole to induce the residual stress
3. Cold expanded hole reamed to $2r$ = 6.35 mm

A fatigue test was performed on a cold expanded hole with a final $2r$ = 6.35, with no crack growth after one million cycles. The test plan was adjusted so that a reasonable failure time could be obtained. To relieve some of the residual stress around the hole, the final ream was increased so that the final $2r$ = 8.738 mm, and this value was used for all cold expanded specimens. In addition, failure at the grip was a problem with the cold expanded specimens, so the specimen widths were reduced to w = 21.59 mm and fiberglass shims were used to minimize damage to the specimens in the gripping area.

Four cold expanded specimens with a final ream $2r$ = 8.738 mm were successfully tested. Fatigue cracks were measured visually with a 25X magnification traveling microscope and an electronic micrometer.

No tensile tests or baseline crack growth rate tests were performed on the 7075-T6 material used for these tests. The stress strain relationship was taken from the Mil Handbook 5D (DOD, 1983) for the plasticity-induced crack closure simulations. This relationship is shown in Figure 2.4. The baseline crack growth rate data for R = 0.1 was taken from the NASGRO material database and the baseline crack growth rate data for R = 0.7 was taken from the Mil Handbook 5. The NASGRO da/dN—ΔK_{eff} relationship is also plotted. This data is shown in Figure 2.5. As this figure shows, the data for the R = 0.1 and the NASGRO da/dN—ΔK_{eff} relationship intersect, but the two lines should be parallel. This issue may be a problem with the data or with NASGRO, but was ignored in this work.

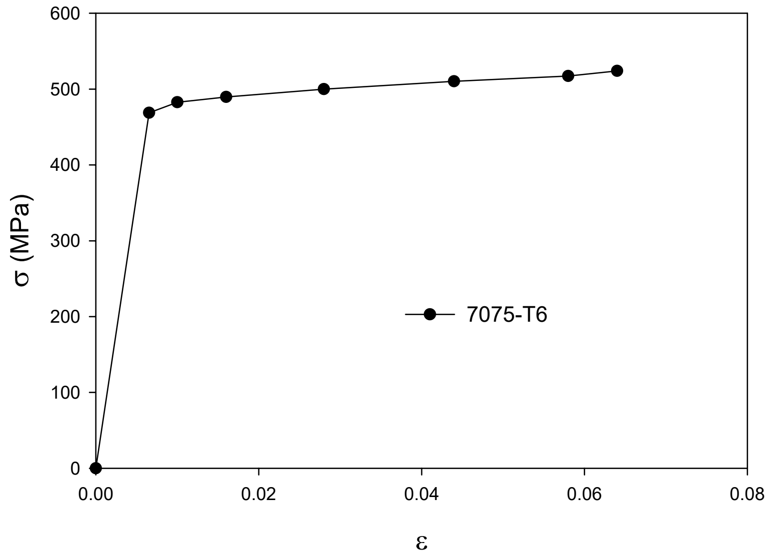

Figure 2-4 – 7075 Stress-Strain Curve

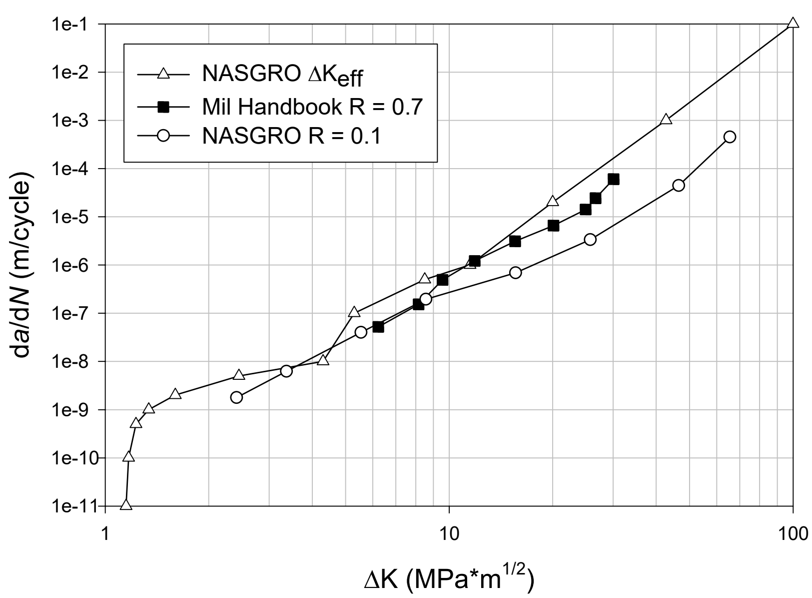

Figure 2-5 – 7075 Baseline Crack Growth Rate Data

CHAPTER III

FINITE ELEMENT ANALYSIS

Overview

Finite element analysis was used simulate the fatigue crack growth in all specimens tested. The commercial finite element program ANSYS version 8.0(ANSYS, 2003) was used. To simulate the elastic-plastic constitutive behavior of the material, the stress strain relationships shown in Figures 2.2 and 2.4 were input into the finite element code for the two materials. A two-dimensional mesh was built using 4-node plane stress elements, with one half of the specimen modeled because of symmetry. Plane stress was used since both geometries tested were relatively thin, although the assumption is more suited for the 7075 simulation since the test specimens were thinner. The plane stress assumption may lead to opening stress values that are too high for the 2024 simulations. The mesh used in the simulation of the tensile overload contained 6529 elements and 6687 nodes, and is shown in Figure 3.1. The mesh used in the simulation of the cold expansion was similar, with more axisymmetric refinement around the hole to enable the reaming process to be simulated.

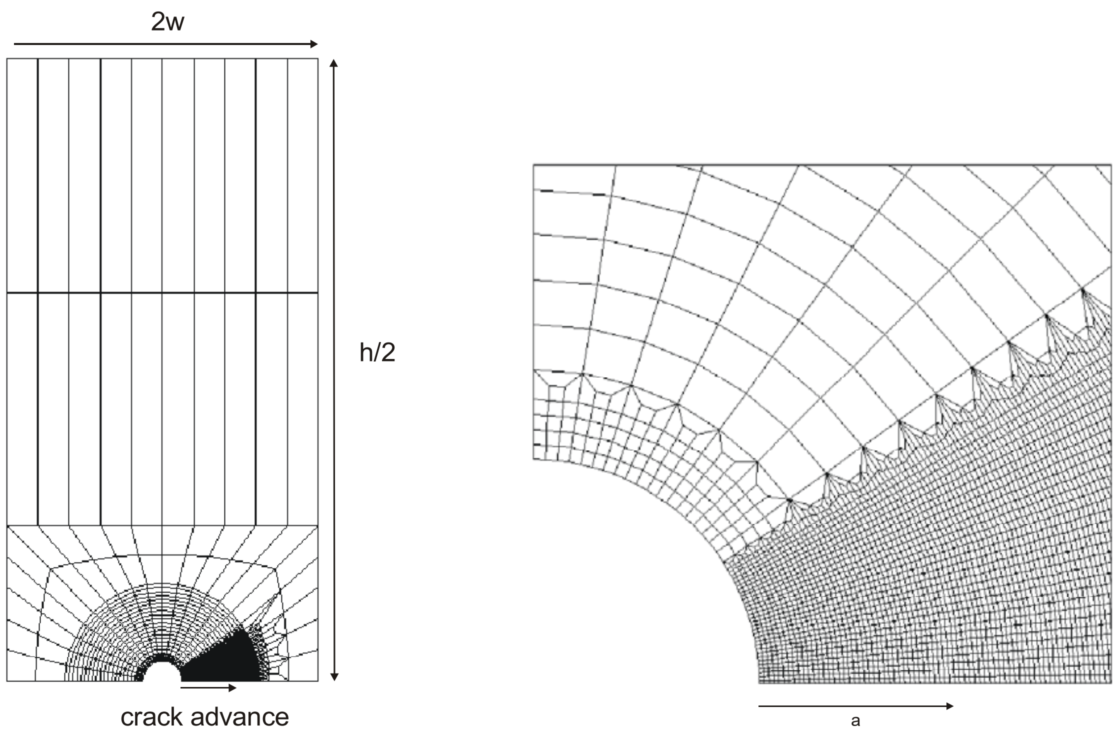

Figure 3-1 – Finite Element Model Mesh

Tensile Overload Simulation

To simulate the overload and induce the residual stress, a uniform stress of 250 MPa was applied and removed. After the overload, elements were removed from the mesh using the Element Kill command in ANSYS to simulate the slotting process. This command gives the removed elements a negligible stiffness to simulate the removal of material (ANSYS, 2003).

Cold Expansion Simulation

To simulate the cold expansion process, three steps were taken. A uniform displacement r_i = 0.134 mm was applied to all nodes on the hole surface. The displacement was then removed to simulate the unloading of the mandrel. The reaming process and EDM notching were then simulated by again using the Element Kill command to give the final residual stress state before the crack growth simulation.

Crack Growth Simulation

After the residual stress and the slotting were simulated, fatigue crack growth was modeled by repeatedly loading, advancing the crack, and then unloading. The model was incrementally loaded to the maximum load, at which time the crack tip node was released, allowing the crack to advance one element length per load cycle. The applied load was then incrementally lowered until the minimum load was attained. Crack surface closure was modeled by changing the boundary conditions on the crack surface nodes. During each increment of unloading, the crack surface nodal displacements were monitored. Between any two increments, if the nodal displacement became negative, the node was closed and a node fixity was applied to prevent crack surface penetration during further unloading. During incremental loading, the reaction forces on the closed nodes were monitored, and when the reaction forces became positive the nodal fixity was removed. A command listing for all the routines involved is included in Appendix A. A sample input file is included in Appendix B. The purpose of this analysis was the computation of the crack opening stress S_o. The crack opening stress was found as the applied stress that first fully opened the crack, regardless of the location along the crack that was last to open. A more detailed discussion of this type of analysis can be found in Solanki *et al*. (Solanki, 2003).

CHAPTER IV

FATIGUE CRACK GROWTH PREDICTION METHODOLOGIES

Superposition Prediction Methodology

To apply the superposition technique, the stress intensity factors were calculated. From Tada and Paris (Tada, 2000), for a crack of length *a* growing from a hole in an infinite plate under a uniform stress *S*:

$$[K(a)]_{\text{inf}} = S\,F(a)\sqrt{\pi a} \tag{9}$$

$$\begin{aligned} F(a) = {} & \left[1 + 0.2\left(1 - \frac{a}{r+a}\right) + 0.3\left(1 - \frac{a}{r+a}\right)^6\right] \cdot \\ & \left[2.243 - 2.64\left(\frac{a}{r+a}\right) + 1.352\left(\frac{a}{r+a}\right)^2 - 0.248\left(\frac{a}{r+a}\right)^3\right] \end{aligned} \tag{10}$$

A finite width correction factor $f(a)$ was taken from Isida (Isida, 1973) such that the stress intensity factor from the applied load is given by

$$(K)_L = f(a)[K(a)]_{\text{inf}} \tag{11}$$

$$f(a)=\sqrt{\frac{1}{\cos\left(\frac{\pi}{2}\frac{2r+a}{2W-a}\right)}\cdot\frac{\sin\left(2\frac{2r+a}{2W-a}\frac{a}{W}\right)}{\left(2\frac{2r+a}{2W-a}\frac{a}{W}\right)}} \tag{12}$$

The stress intensity factors for the initial residual stress $(K)_R$ were determined using a weight function computation and the residual stress resulting from the simulation of the overload or cold expansion and subsequent slotting. Denoting the weight function as *m(x,a)* and the residual stress in the uncracked slotted body as $\sigma(x)$, the stress intensity factor was computed as:

$$K_R=\int_0^a \sigma(x)\cdot m(x,a)dx \tag{13}$$

The weight function was taken from Wu and Carlsson (Wu, 1991). The stress intensity factors due to the applied loading and the residual stress were then added to give a resultant stress intensity at the maximum and minimum load. If any resultant stress intensity was less than zero, it was taken to be zero.

$$\begin{aligned} K &= (K_L) + (K_R) \qquad && K>0 \\ K &= 0 && K<0 \end{aligned} \tag{14}$$

The stress intensity factor range was then computed as the difference between the maximum and minimum resultant stress intensity factors, creating an effective $R = 0$ loading. A center-crack baseline crack growth rate curve for $R = 0$ was not determined by Liu, and the curve for $R = 0.1$ was used as an approximate replacement to determine the corresponding crack growth rates from the computed stress intensity factor ranges.

The crack growth rate was then used with a prescribed small da to find a corresponding dN, and these incremental values were added to the previous values of a and N to give the life prediction. The FORTRAN program used for the superposition calculations is included in Appendix C.

Finite Element Prediction Methodology

The crack growth was also predicted using the opening stresses from the finite element analysis, as shown schematically in Figure 4.1. Equation 11 was used to find the maximum stress intensity factor K_{max} and the opening stress intensity factor K_o using $S = S_{max}$ and $S = S_o$ respectively. The effective stress intensity range was then calculated as the difference between the maximum and opening stress intensity factors. Both the baseline crack growth curve for the Liu $R = 0.7$ data and the NASGRO da/dN—ΔK_{eff} curve were used as an effective crack growth rate curve $da/dN = g(\Delta K_{eff})$, and each curve was used to determine crack growth rates for the computed effective stress intensity ranges. These crack growth rates were used to determine the crack length as a function of the number of cycles. The FORTRAN program used for the closure calculations is included in Appendix D.

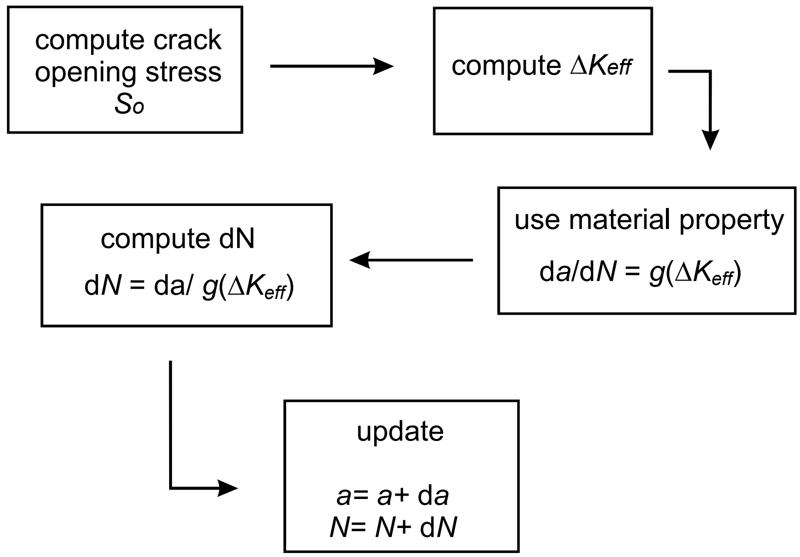

Figure 4-1 – Crack Closure-Based Crack Growth Prediction Methodology

CHAPTER V

RESULTS AND DISCUSSION

Tensile Overload Results

The residual stress determined from the finite element analysis is shown in Figure 5.1. The initial residual stress from the overload shows a maximum compressive stress at the edge of the hole nearly the same magnitude as yield strength of the material. After the slotting process was simulated, the compressive residual stress at the notch edge was increased. Since the slot is modeled as an idealized rectangle with sharp corners, the residual stress predictions may be overestimated near the notch end.

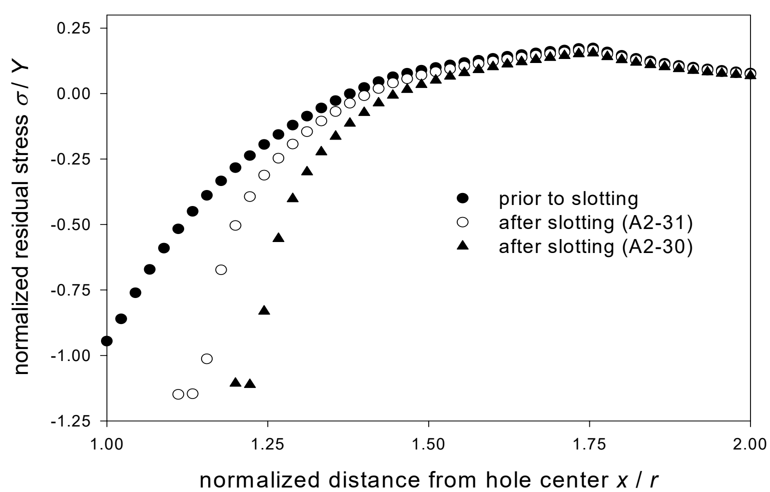

Figure 5-1 – Residual Stress Results from Finite Element Simulation of the Tensile Overload

The opening stress results given by the finite element crack growth analysis are shown in Figure 5.2. Both plots illustrate opening stresses that initially exhibit a sharp increase, then decrease gradually before again rising slowly. The initial sharp increase indicates that the crack growth is slower initially due to the residual stress at the edge of the notch. For test A2-31, the opening stress for a residual stress free specimen is shown for comparison. The opening stresses merge after approximately 5.2 mm of crack growth.

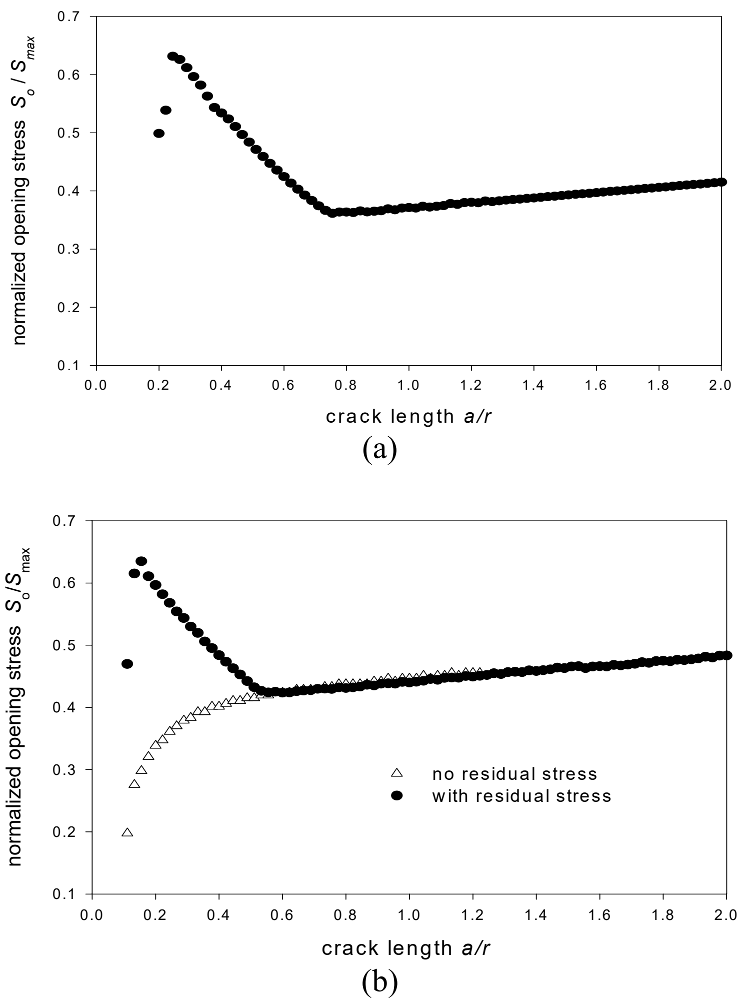

Figure 5-2 – Predicted Crack Opening Stress, Test A2-30 (a) and test A2-31 (b)

Figure 5.3 shows the predicted crack growth as a function of the number of cycles for test A2-30 from both the superposition and finite element methods. The results from each method are compared to the experimental data. The plot indicates that the superposition method predicts lives much longer than the experimental data. The finite element predictions are more conservative, leading to shorter lives than seen in the experiment. When the NASGRO da/dN—ΔK_{eff} curve is used, the finite element analysis prediction compares better with the experimental data when compared with the finite element analysis prediction that uses the Liu R = 0.7 data to define the da/dN—ΔK_{eff} relationship. The NASGRO curve extends to larger ΔK_{eff} values and does not exhibit a nonlinearity in the higher ΔK_{eff} region as seen in Figure 2.3. This nonlinearity is often seen in high R tests and its presence likely produced the poor fatigue crack growth predictions observed when using the R = 0.7 baseline data.

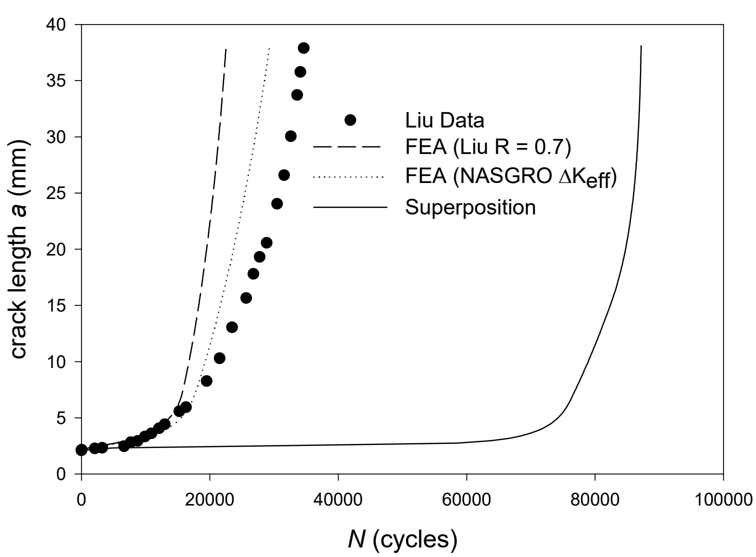

Figure 5-3 – 2024 Fatigue Crack Growth: Predicted and Actual, Test A2-30

Figure 5.4 shows the predicted crack growth rate for test A2-31 from both the superposition and finite element methods, compared with the experimental data. Again the NASGRO da/dN—ΔK_{eff} curve results in good agreement with data. The superposition method again overestimates the actual life, but for this test the difference is much smaller.

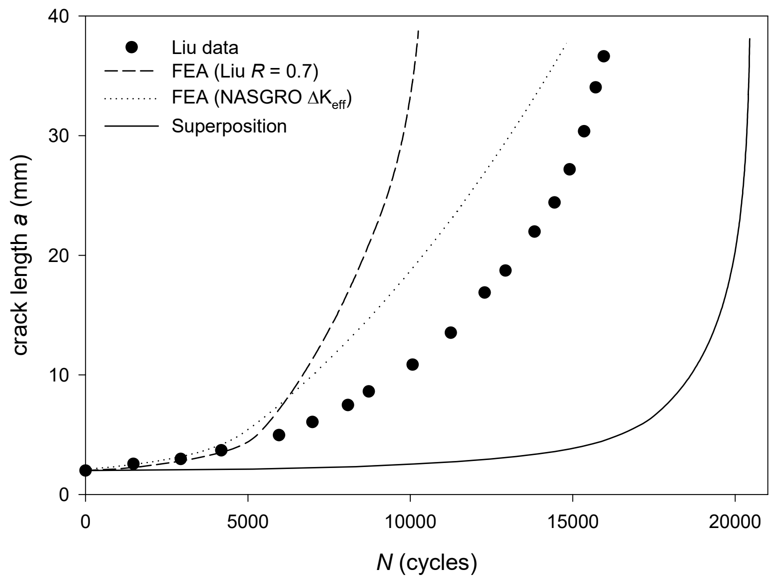

Figure 5-4 – 2024 Fatigue Crack Growth: Predicted and Actual, Test A2-31

To further justify the use of the one da/dN—ΔK_{eff} relationship over the other, a simple verification was employed. Fatigue crack growth in center-cracked specimens tested by Liu with $R = 0.1$ was predicted using a plane strain ($\alpha = 3$) opening stress value $S_o / S_{max} = 0.265$ from Newman (Newman, 1984) and calculating ΔK_{eff} using a center-crack stress intensity factor solution taken from Tada and Paris (Tada, 2000). The specimen tested was relatively thick, with $t = 6.35$ mm, so plane strain is a reasonable assumption. Both the NASGRO da/dN—ΔK_{eff} and Liu da/dN—ΔK_{eff} were used to predict fatigue crack growth. From Figure 5.5, use of the Liu da/d*N*—ΔK_{eff} relationship resulted in better correlation with the experimental data. The conflicting results may be due to the maximum values of Δ*K* calculated by the finite element method. The highest calculated Δ*K* is over 21 MPa*m$^{1/2}$, a value outside of the range of the Liu *R* = 0.7 data. The NASGRO equation is a better option given that no extrapolation is needed when the calculated Δ*K* values extend outside of the region of the Liu data.

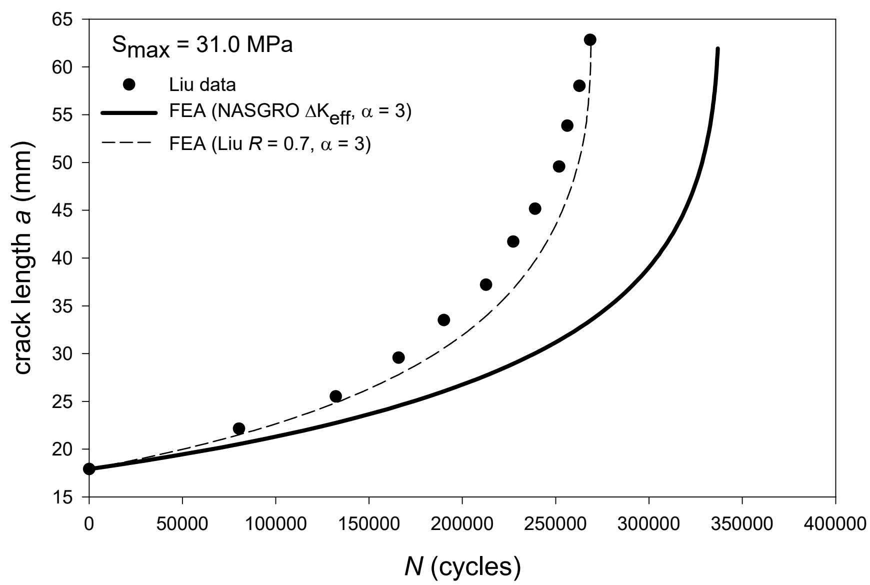

Figure 5-5 – 2024 Center-Crack Specimen Fatigue Crack Growth: Predicted and Actual

Next consider the effect of the notching process on the fatigue life. Figure 5.6 compares superposition-based predictions made using the initial residual stress and the redistributed residual stress after slotting. The difference is fairly small when comparing the results given by each residual stress. The redistribution is not significant here, but it may have a much greater influence in other cases.

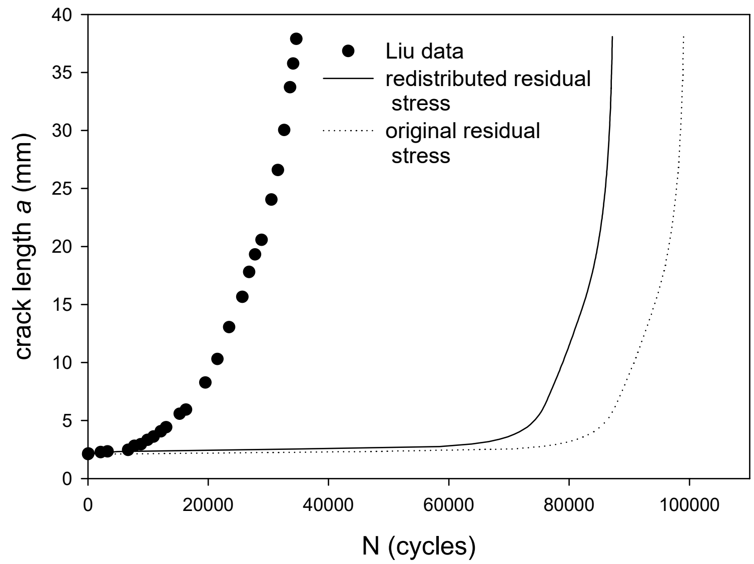

Figure 5-6 – 2024 Fatigue Crack Growth, Superposition Predictions with Slotted and Unslotted Residual Stress and Actual, Test A2-30

In addition to opening stresses, the finite element analysis also provided profiles of the crack as it opened and closed. Figures 5.7 and 5.8 give predicted profiles for a short crack and a long crack for test A2-31 to illustrate the different crack opening behavior at different crack lengths. Figure 5.7 shows that for short cracks remote closure occurs and the crack mouth is the last to open. Figure 5.8 indicates that remote closure does not occur when the crack becomes longer, with the crack tip being the last location to open. No distinction between these two different crack opening behaviors was made in the fatigue crack growth predictions presented here.

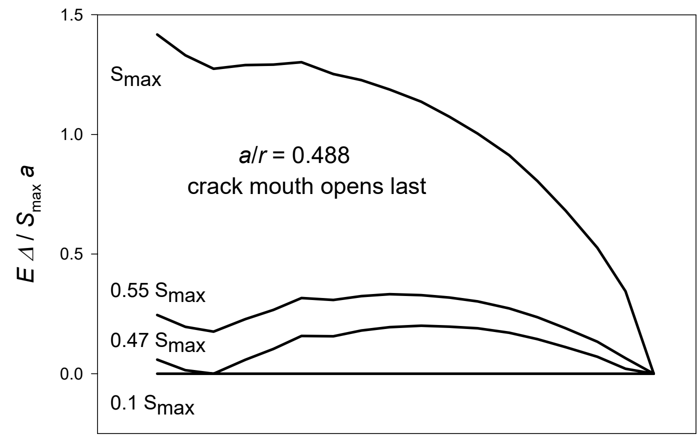

Figure 5-7 – Test A2-31 Short Crack Profile

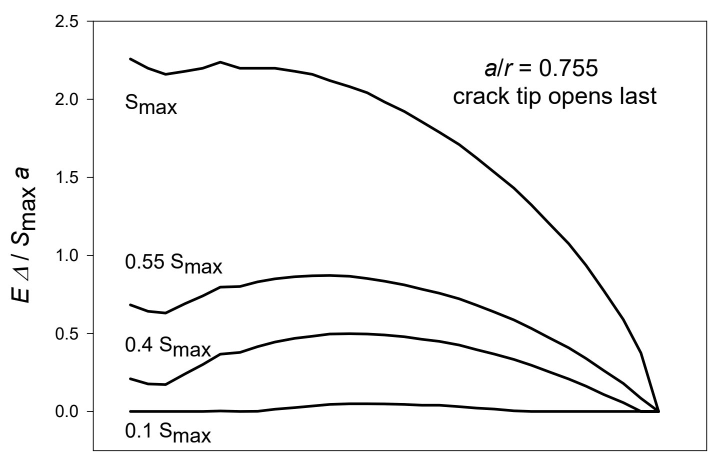

Figure 5-8 – Test A2-31 Long Crack Profile

Cold Expansion Results

The residual stress determined from the finite element analysis for the cold expanded specimens is shown in Figure 5.9. The residual stress after the cold expansion shows a maximum compressive stress at the edge of hole near the magnitude of the yield strength of the material. This residual stress corresponds to the first cold expanded test that resulted in no crack growth. After reaming, the residual stress is relaxed considerably, with the magnitude reduced by a factor of two. After the slotting process is simulated, the peak residual stress increases to reach the yield strength again. This residual stress may be overestimated because the slot was modeled as a rectangular notch. The finite element analysis predicted a high strain gradient at the corners of the slot, which would likely not be seen if a slot with more realistic rounded corners was considered.

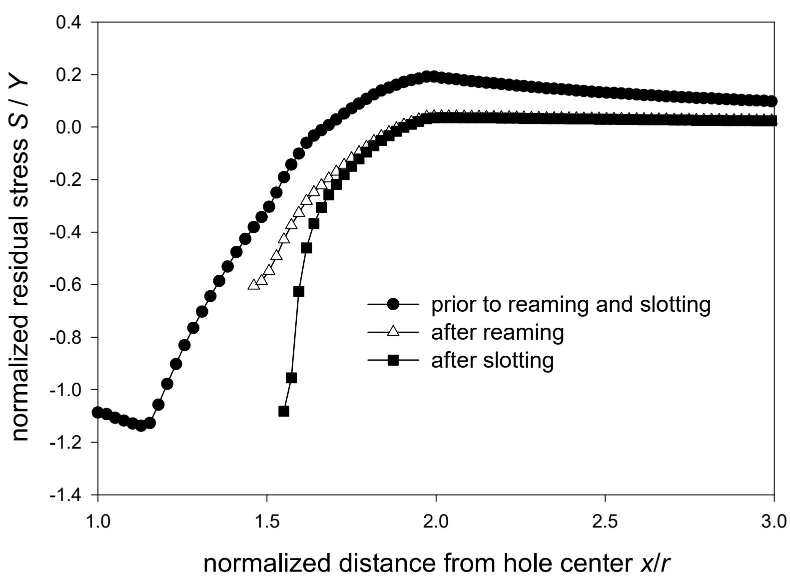

Figure 5-9 – Residual Stress Results from Finite Element Simulation of the Cold Expansion Process

The opening stress results given by the finite element crack growth analysis are shown in Figure 5.10. The results are similar to those from the tensile overload simulation. The opening stresses from the cold expanded simulation show an initial increase before decreasing to approximately the steady state value. The high initial opening stress values are a result of remote crack closure. The crack mouth node is the last to open until the opening stress value reaches steady state, which indicates the crack tip node is now the last to open. The opening stress values for the non-cold expanded hole start at zero and climb to the steady state value.

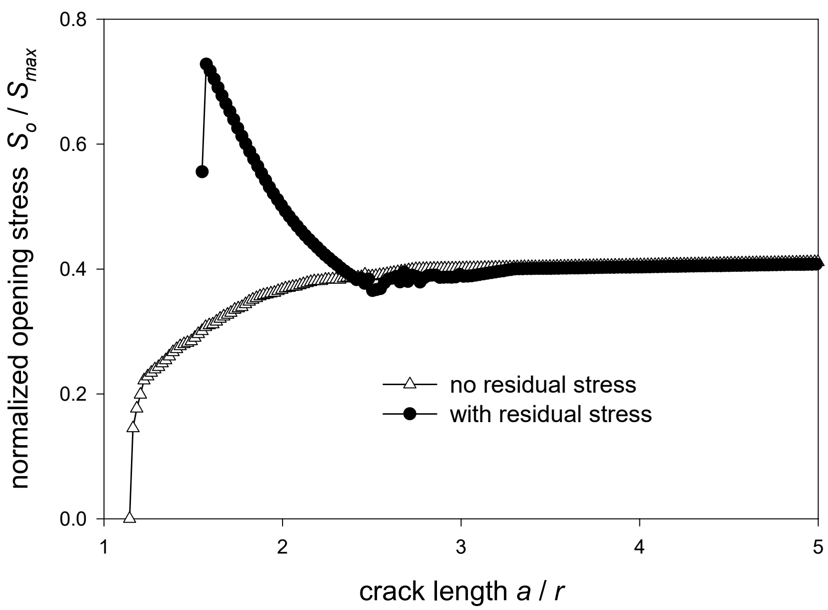

Figure 5-10 – Predicted Opening Stress for the 7075-T6 Simulations

Figure 5.11 shows the predicted crack growth as a function of the number of cycles for the cold expanded tests from the superposition and finite element methods. The results are compared with the experimental data. The superposition method predicts the crack growth much better than the finite element method. The finite element prediction is in error by a factor of approximately four when the NASGRO da/dN—ΔK_{eff} relationship is used, but this error is reduced to around two when the Mil da/dN—ΔK_{eff} relationship is used. This error questions the reliability of the opening stress calculations. A more refined mesh may give better results, but the crack growth algorithm had convergence problems with smaller elements. Although all predictions were conservative, the superposition method compared fairly well with the experimental data.

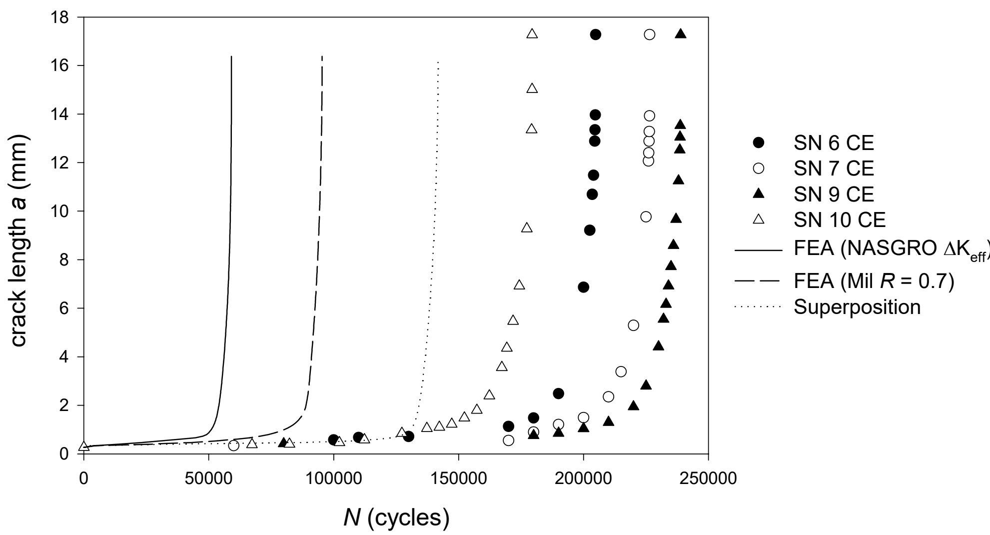

Figure 5-11 – 7075 Fatigue Crack Growth for Cold Expanded Hole: Predicted and Actual

The predicted crack growth for the non-cold expanded hole is compared with experimental data in Figure 5.12. Since the test simulated is simple constant amplitude with no residual stress, more accuracy was expected from the finite element predictions when compared with the data, but as the plot illustrates, the comparison is fairly poor with the prediction differing from the data by a factor of two with the NASGRO da/dN—

ΔK_{eff} relationship. Again the use of the Mil da/dN—ΔK_{eff} relationship reduces this error, but the results again point to the opening stress calculations as the probable cause of the poor agreement. A more refined mesh may produce the higher opening stress values needed to allow the prediction to agree better with the data. Any attempts to run a more refined mesh were unsuccessful because of convergence issues in the finite element code.

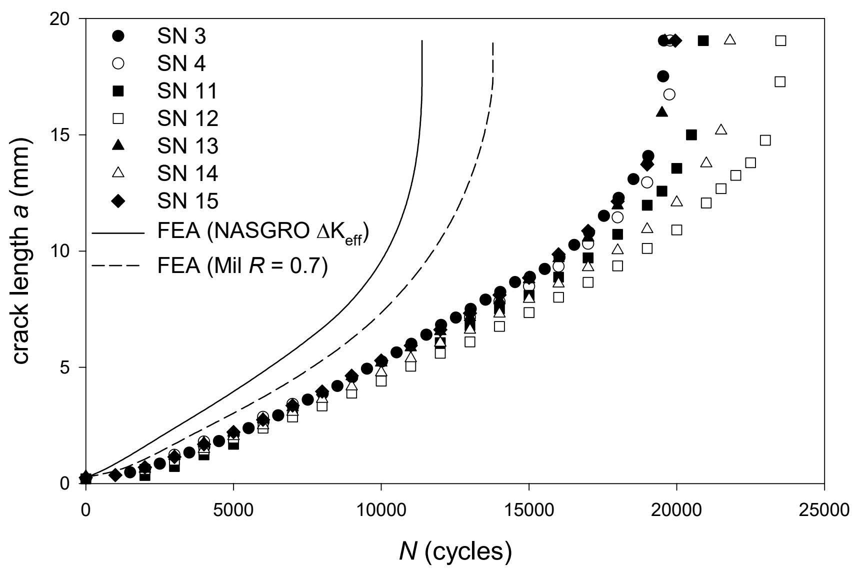

Figure 5-12 – 7075 Fatigue Crack Growth for Non-Cold Expanded Hole: Predicted and Actual

The residual stress redistribution from the slotting process was also investigated by using the slotted and unslotted residual stress with the superposition method to predict the fatigue crack growth. Figure 5.13 shows the predictions compared with the experimental data. The unslotted residual stress led to a better comparison than the slotted residual stress. The difference is small and demonstrates that the simulating the slotting process does not make a considerable difference in the resulting life predictions.

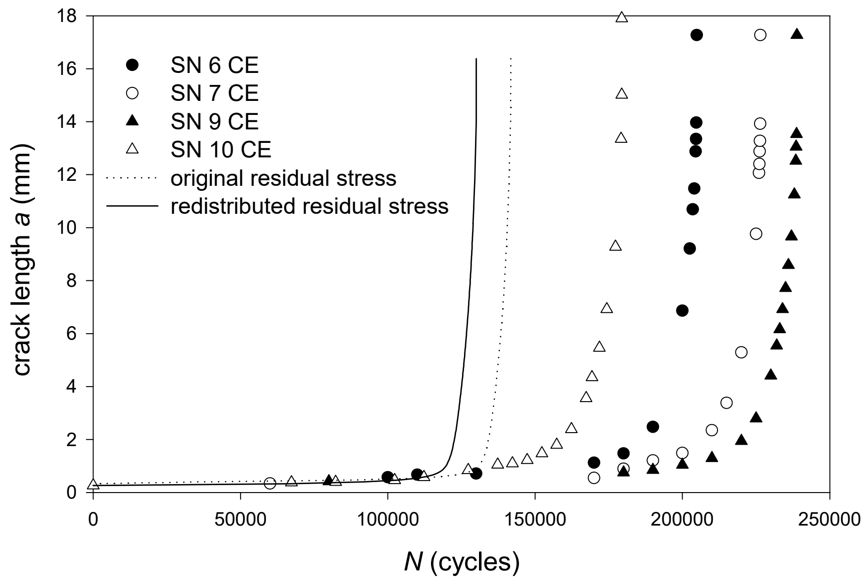

Figure 5-13 – 7075 Fatigue Crack Growth, Superposition Predictions with Slotted and Unslotted Residual Stress and Actual

Convergence Problems

Fatigue crack growth simulations were attempted with varying levels of mesh refinement. The purpose of this mesh refinement study was to determine if convergence of the computed opening stress values occurs. As the elements in the mesh become smaller, the opening stress values should not change if convergence is achieved. Figure 5.14 gives the opening stress predictions for three different meshes in the simulation of test A2-30. For the meshes corresponding to a crack growth element size d*a* of 0.6350 mm and 0.2117 mm, the opening stress values appear to converge. The third mesh with the smallest element size did not finish due to problems with the finite element code ANSYS. The code had difficulty in solving when the element size d*a* = 0.0706 mm was used. The simulation of test A2-30 completed enough cycles to use for comparison. As the figure illustrates, the computed opening stress values are increased rather considerably. This divergence indicates a problem with mesh refinement that was not rectified for the work presented here, and the results presented here may be unreliable. The higher opening stress values would lead to a longer predicted life and could possibly compare with the experimental data much better.

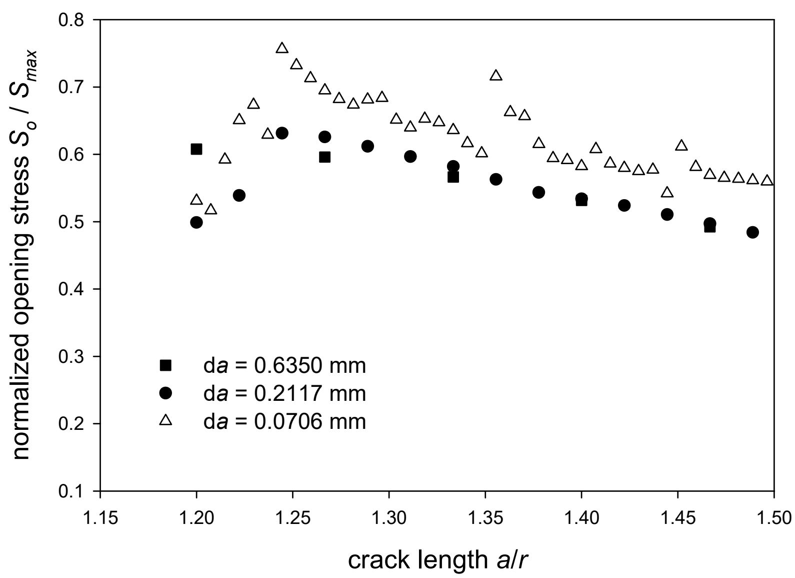

Figure 5-14 – Opening Stresses for Different Element Sizes in Test A2-30 Simulation

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

Conclusions

Crack growth predictions using elastic superposition are relatively simple and accurate. A slotting process for purposes of starting a fatigue crack introduces a significant redistribution of the residual stress, and this may have a considerable effect on the subsequent fatigue life. The finite element method presented here created conflicting results for the two different tests simulated. The finite element method worked well predicting for the tensile overload tests, but did not result in accurate predictions for the cold expanded tests. One factor may be mesh refinement, but the crack growth algorithm employed here did not allow for a more refined mesh. This problem should be investigated.

Crack growth simulations utilizing plasticity-induced crack closure concepts are sensitive to the crack growth rate relationship $da/dN = f(\Delta K_{eff})$ used. This represents a potentially serious problem and highlights the need for experimental methods to reliably measure this relationship. For the work presented here for the tensile overload tests, the baseline data for $R = 0.7$ from Liu's work was initially chosen as the $da/dN = f(\Delta K_{eff})$ relationship. This relationship could have also been defined by closure-free baseline data for $R = 0.6$ or $R = 0.8$, if such data were available, and this could have changed the results significantly. As stress ratio is increased, the data exhibits a nonlinearity at a lower ΔK, which will lead to inaccurate predictions. This nonlinearity occurs because of the specimen size used when generating the baseline data. If a narrow specimen is used, such as the specimen used by Liu, fracture occurs sooner, but if a wider specimen is used, fracture does not occur as soon and this nonlinearity is not seen.

Another concern with the crack growth relationship is the threshold region. In the finite element predictions, the high opening stresses lead to low ΔK values. These values may lie in the threshold region of the material tested, and there are conflicts about what the threshold value may be. Accurate predictions will not be produced until the threshold of the material tested is properly defined.

The tensile overload predictions are compared with a single test only, and the comparisons are not a sufficient validation due to the inherent scatter in fatigue crack growth data. More tests are needed to validate the prediction methodologies used in this study, although the superposition method appears to be as reliable as the finite element prediction method.

Suggested Future Work

A mesh refinement study needs to be completed to verify the work presented here. As shown here the opening stress calculations did not converge when smaller element sizes were used. The crack growth algorithm employed here needs to be adjusted to allow for more refined meshes in the simulations.

REFERENCES

ANSYS, 2003 — *ANSYS User's Manuals*, Revision 8.0, 2003

Beghini, 1990 — Beghini, M. and Bertini, L. "Fatigue Crack Propagation through Residual Stress Fields with Closure Phenomena," *Engineering Fracture Mechanics*, Vol. 36, No. 3, 1990, pp. 379-387.

Besuner, 1986 — Besuner, P.M., Harris, D.O. and Thomas, J.M., "A Review of Fracture Mechanics Technology," *NASA Contractor Report 3957*, NASA, 1986.

Broek, 1986 — Broek, D. *Elementary Engineering Fracture Mechanics*, 4th edn. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1986.

Bueckner, 1958 — Bueckner, H.F., "The Propagation of Crack and the Energy of Elastic Deformation," *Transactions of the ASME*, Series A-D 80, 1958, pp. 1225-1230.

Chandawanich, 1979 — Chandawanich, N. and Sharpe, Jr., W.N., "An Experimental Study of Fatigue Crack Initiation and Growth from Coldworked Holes," *Engineering Fracture Mechanics*, Vol. 11, 1979, pp. 609-620.

Chen, 1986 — Chen, P.C.T., "A New Method of Predicting Residual Stresses in Autofrettaged Gun Barrels," *Technical Report ARCCB-TR-86012,* US Army Armament Research and Development Center, Watervliet, 1986.

Choi, 1995 — Choi, H.C. and Song, J.H., "Finite Element Analysis of Closure Behaviour of Fatigue Cracks in Residual Stress Fields," *Fatigue and Fracture of Engineering Materials & Structures*, Vol.18, No. 1, 1995, pp. 105-117.

DOD, 1983 — Department of Defense, *Mil-Handbook 5*, Vol. 1, Department of Defense, Washington D.C., 1983.

Elber, 1970 — Elber, W., "Fatigue Crack Closure under Cyclic Tension," *Engineering Fracture Mechanics*, Vol. 2, 1970, pp. 37-44.

Elber, 1971 Elber, W. in Damage Tolerance in Aircraft Structures, *ASTM STP 486*, American Society for Testing and Materials, 1971, pp. 230-242.

FTI, 1991 FTI, *Extending the Fatigue Life of Metal Structures*, Materials Testing, Fatigue Technology Inc., 150 Andover Park West, Seattle, Washington, 1991.

Fukuda, 1978 Fukuda, S. and Tsuruta, Y. *Transactions of JWRI*, Vol. 7, 1978, pp. 67.

Heaton, 1976 Heaton, M.D., "On the Calculation of Stress Intensity Factors Due to Thermal and Residual Stress Fields," *CEGB Research Report NW/SSD/RR/158*, 1976.

Hsu, 1975 Hsu, Y.C. and Forman, R.G., "Elastic-Plastic Analysis of an Infinite Sheet Having a Circular Hole under Pressure," *Journal of Applied Mechanics*, Vol. 42, 1975, pp. 347-352.

Isida, 1973 Isida, M. "Laurent Series Expansion for Internal Crack Problems," *Methods of Analysis and Solutions of Crack Problems*, G. C. Sih, ed., Noordhoff International Publishing, 1973.

Kang, 2002 Kang, J., Johnson, W.S. and Clark, D.A., "Three-Dimensional Finite Element Analysis of the Cold Expansion of Fastener Holes in Two Aluminum Alloys," *Transactions of the ASME*, Vol. 124, 2002, pp. 140-145.

Kiciak, 1996 Kiciak, A., Glinka, G. and Burns, D.J., "Analysis of Fatigue Growth of Semi-Elliptical Cracks Through Residual Stress Fields," *Proceedings of 6th International Fatigue Congress (Fatigue 96)*, Editors G. Lüjenry and H Nowack, 1996, Pergamon, Vol. II, pp. 1207-1212.

Lam, 1989 Lam, Y.C. and Lian, K.S., "Effect of Residual Stress and its Redistribution on Fatigue Crack Growth," *Theoretical Applied Fracture Mechanics,* Vol. 12, No. 1, 1989, pp. 59-66.

Lee, 1998 Lee, Y., Chung, C., Park, Y. and Kim, H., "Effects of Redistributing Residual Stress on the Fatigue Behavior of SS330 Weldment," *International Journal of Fatigue*, Vol. 20, No. 8, 1998, pp. 565-573.

Leis, 1997 Leis, B., "Effect of Surface Conditioning and Processing on Fatigue Performance," *ASM Handbook, Fatigue and Fracture*, v 19, 1997.

Liu, 1979 Liu, A.F., "The Effect of Residual Stresses on Crack Growth from a Hole," *NOR 79-74*, Northrop Corporation Aircraft Group, 1979.

McClung, 1989 McClung, R.C. and Sehitoglu, H., "On the Finite Element Analysis of Fatigue Crack Closure—1. Basic Modeling Issues," *Engineering Fracture Mechanics*, Vol. 33, No. 2, 1989, pp. 237-252.

NASGRO, 2002 *NASGRO Reference Manual (version 4.02)*, NASA Johnson Space Center and Southwest Research Institute, 2002.

Nelson, 1982 Nelson, D.V., *Residual Stress Effects in Fatigue*, *ASTM STP 776*, American Society for Testing and Materials, 1982, pp. 172-194.

Newman, 1984 Newman Jr., J.C., "A Crack Opening Stress Equation for Fatigue Crack Growth," *International Journal of Fracture*, Vol. 24, 1984, pp. R131-R135.

Ozdemir, 1993 Ozdemir, A.T., Cook, R. and Edwards, L., "Residual Stress Distributions around Cold Expanded Holes," ICAF Symosium, Stockholm, 1993.

Parker, 1982 Parker, A.P*., Residual Stress Effects in Fatigue,* ASTM STP 776, American Society for Testing and Materials, 1982, pp. 13-31.

Pavier, 1997 Pavier, M.J., Poussard, C.G.C. and Smith, D.J., "A Finite Element Simulation of the Cold Working Process for Fastener Holes," *Journal of Strain Analysis*, Vol. 32, No. 4, 1997, pp. 287-300.

Poussard, 1995 Poussard, C., Pavier, M.J. and Smith, D.J., "Analytical and Finite Element Predictions of Residual Stresses in Cold Worked Fastener Holes," *Journal of Strain Analysis*, Vol. 30, No. 4, 1995.

Rice, 1967 Rice, J.R., *Fatigue Crack Propagation*, ASTM STP 415, American Society for Testing and Materials, 1967, pp. 247-311.

Rich, 1975 Rich, D.L. and Impellizzeri, L.F., "Fatigue Analysis of Cold-Worked and Interference Fit Fastener Holes," *Cyclic Stress-Strain and Plastic Deformation Aspects of Fatigue Crack Growth*, ASTM STP 637, 1975, pp. 153-175.

SAE, 1997	SAE Fatigue Design and Evaluation Steering Committee, *Fatigue Design Handbook*, 3rd edition, Society of Automotive Engineers, 1997.
Skinner, 2001	Skinner, J.D. and Daniewicz, S.R., "Simulation of Plasticity-Induced Fatigue Crack Closure in Part-Through Cracked Geometries Using Finite Element Analysis," Engineering Fracture Mechanics, Vol. 69, No. 1, 2001, pp. 1-11.
Solanki, 2003	Solanki, K., Daniewicz, S.R. and Newman Jr., J.C., "Finite Element Modeling of Plasticity-Induced Crack Closure with Emphasis on Geometry and Mesh Refinement Effects," *Engineering Fracture Mechanics*, Vol. 70, 2003, pp. 1475-1489.
Stephens, 2001	Stephens, R.I., Fatemi, A., Stephens, R.R. and Fuchs, H.O. *Metal Fatigue in Engineering*, John Wiley & Sons, New York, 2001.
Tada, 2000	Tada, H., Paris, P.C. and Irwin, G.R. *The Stress Analysis of Crack Handbooks*, 3rd ed., ASME, 2000.
Todoroki, 1991	Todoroki, A. and Kobayashi, H., "Prediction of Fatigue Crack Growth Rate in Residual Stress Fields," *Key Engineering Materials, Fracture and Strength '90*, Vol. 51-52, 1991, p. 367.
Underwood, 1977	Underwood, J.H., Pook, L.P. and Sharples, J.K., *Flaw Growth and Fracture*, ASTM STP 631, American Society for Testing and Materials, 1977, pp. 402-415.
Wilks, 1993	Wilks, M.D.B., Nowell, D. and Hills, D.A., "The Evaluation of Stress Intensity Factors for Plane Cracks in Residual Stress Fields," *Journal of Strain Analysis*, Vol. 28, No. 3, 1993, p. 145.
Wu, 1991	Wu, X. and Carlsson, A.J., *Weight Functions and Stress Intensity Factor Solutions*, Pergamon, 1991.
Zhu, 1987	Zhu, W. and Zha, Z., "An Elastic-Plastic Analysis of Autofrettaged Thick-Walled Cylinders," *Acta Mech. Sinica*, Vol. 19, 1987, pp. 245-255.

APPENDIX A.1

ANSYS INPUT FILE *APPBCS.MAC*

```
! This is the input file used for the application of boundary
! conditions for the model

/prep7

! Element Shape Checking Off
SHPP,OFF

! Define element type

E,,PLANE42,

! Define Material Properties for Solid Elements
MPTEMP,,,,,,,,
MPTEMP,1,0
MPDATA,EX,1,,10550
MPDATA,PRXY,1,,.33
TB,MISO,1,1,6,
TBPT,,(51.145/10550),51.145
TBPT,,.0075,54.15
TBPT,,.01,54.465
TBPT,,.02,57.448
TBPT,,.03,59.747
TBPT,,.04,61.815

! Begin Building Model: Read Solid ELements from File

nread,%JN%,crd
eread,%JN%,elm

! Simulate tensile overload or cold expansion if needed

!Overload
!Coldx

!Constrain Nodes on Bottom Surface of Plate in the Vertical-direction
!(Constraints will be removed during crack growth)

  NSEL,S,LOC,Y,0
  NSEL,R,LOC,X,c,100000
  NSEL,U,LOC,X,0,c-da*0.25
  NSEL,A,LOC,X,-100000,0
  D,ALL,UY,0

!Apply Appropriate Conditions in X-direction:

  NSEL,S,LOC,Y,0
  NSEL,R,LOC,X,-w
  D,ALL,UX,0
  NSEL,ALL

!Select Crack Mouth Node...Create Component CMNODES

NSEL,S,LOC,X,c
NSEL,R,LOC,Y,0
CM,CMNODES,NODE

CMSEL,ALL
NSEL,ALL
ESEL,ALL
```

```
WSORT,ALL,0  !Sort Elements to minimize wavefront

SAVE
FINISH
```

APPENDIX A.2

ANSYS INPUT FILE *OVERLOAD.MAC*

```
! This is the input file used to apply the tensile overload and induce
! the residual stress

! Apply overload and appropriate boundary conditions

NSEL,S,LOC,Y,12
SF,ALL,PRES,-36
NSEL,S,LOC,Y,0
D,ALL,UY,0
NSEL,S,LOC,X,0
D,ALL,UX,0
NSEL,ALL
ESEL,ALL
TIME,0.10
SOLVE

! Remove overload

SFDELE,ALL,ALL
DDELE,ALL,ALL
NSEL,S,LOC,X,
D,ALL,UX,0
NSEL,S,LOC,Y,
D,ALL,UY,0
NSEL,ALL
ESEL,ALL
TIME,0.20
SOLVE

! Remove elements from mesh if a slotting process is to be simulated

EKILL,6288
EKILL,6287
EKILL,6282
EKILL,6280
EKILL,6279
EKILL,6274
EKILL,6240
EKILL,6239
EKILL,6234
NSEL,S,LOC,Y,0
NSEL,R,LOC,X,,.449
DDELE,ALL,ALL
NSEL,ALL
ESEL,ALL
TIME,0.30
SOLVE
NSEL,ALL
DDELE,ALL,ALL
```

APPENDIX A.3

ANSYS INPUT FILE *COLDX.MAC*

```
! This is the input file used to apply the cold expansion and induce
! the residual stress

! Select hole surface nodes

*GET,NMAX,NODE,0,NUM,MAX
*DIM,ANODEX,,NMAX
*VGET,ANODEX(1),NODE,1,LOC,X
*DIM,ANODEY,,NMAX
*VGET,ANODEY(1),NODE,1,LOC,Y
*DIM,R,,NMAX
ii=1
*DO,J,1,NMAX
   R(J)=(ANODEX(J))**2+(ANODEY(J))**2
   *IF,R(J),LE,0.0157,THEN
      *IF,ii,EQ,1,THEN
         NSEL,S,,,J
         ii=ii+1
      *ENDIF
      *IF,ii,GT,1,THEN
         NSEL,A,,,J
         ii=ii+1
      *ENDIF
   *ENDIF
*ENDDO

! Apply hole displacements and appropriate boundary conditions

*GET,nodeno,node,,COUNT
ndno=0
a=.005
*DO,i,1,nodeno
  ndNO=ndNEXT(ndNO)
  *GET,nodeX,node,ndNO,loc,x
  *GET,nodeY,node,ndNO,loc,y
  *IF,nodex,gt,0,then
  al=ATAN(nodey/nodex)
  dx=a*cos(al)
  dy=a*sin(al)
  D,ndno,UX,dx
  D,ndno,UY,dy
  *ENDIF
  *IF,nodex,lt,0,then
    al=ATAN(nodey/nodex)
    dx=-1*a*cos(al)
    dy=-1*a*sin(al)
    D,ndno,UX,dx
    D,ndno,UY,dy
  *ENDIF
  *IF,nodex,eq,0,then
    dy=a
    D,ndno,UY,dy
  *ENDIF
*ENDDO
NSEL,S,LOC,Y,0
D,ALL,UY,0
NSEL,S,LOC,X,0
D,ALL,UX,0
NSEL,ALL
ESEL,ALL
TIME,0.10
```

```
SOLVE
! Remove hole displacements and apply boundary conditions

NSEL,ALL
ESEL,ALL
DDELE,ALL,ALL
NSEL,S,LOC,X,0
D,ALL,UX,0
NSEL,S,LOC,Y,0
D,ALL,UY,0
NSEL,ALL
ESEL,ALL
TIME,0.20
SOLVE

! Remove elements from mesh if a slotting process is to be simulated

EKILL,6288
EKILL,6288
EKILL,6288
EKILL,6288
EKILL,6288
NSEL,S,LOC,Y,0
NSEL,R,LOC,X,,.180
DDELE,ALL,ALL
NSEL,ALL
ESEL,ALL
TIME,0.30
SOLVE
NSEL,ALL
DDELE,ALL,ALL
```

APPENDIX A.4

ANSYS INPUT FILE *STRTCYC.MAC*

```
! This is the input file used for control of the cyclic loading

FirstLoad
ClearRST,BDrive,BDir,MaxDir
*DO,I,1,NLC
AdvanceCrack,I
ClearRST,BDrive,BDir,''
UnloadCrack,I
ClearRST,BDrive,BDir,MinDir
  LoadCrack,I
  ClearRST,BDrive,BDir,MaxDir
*ENDDO
```

APPENDIX A.5

ANSYS INPUT FILE *FIRSTLOAD.MAC*

```
! Apply Maximum Load on First Cycle:


/SOLU
Appload,height,StrsMax
!AUTOTS,ON
!NSUBST,5,10000,5,ON
TIME,0.45
SOLVE
SAVE
```

APPENDIX A.6

ANSYS INPUT FILE *ADVANCECRACK.MAC*

```
! AdvanceCrack.mac
! Macro File to advance crack uniformly one element
!
! Should be executed as:
!
! AdvanceCrack,LoadCycleNumber
!

AUTOTS,ON
NSUBST,5,1,1

/PREP7
SelCTNodes,arg1
*GET,NNODES,NODE,,COUNT
NODNO=0
*DO,JJ,1,NNODES
  NODNO=NDNEXT(NODNO)
  *GET,NODERF,NODE,NODNO,RF,FY
  DDELE,NODNO,UY
  F,NODNO,FY,NODERF
*ENDDO
NSEL,ALL
ESEL,ALL
/SOLU
ANTYPE,,REST

TimeVar=0.45+0.05/(NCGECut+1)+(arg1-1)
Time,TimeVar
SOLVE
SAVE

*DO,J,1,NCGECut-1
  /PREP7
  SelCTNodes,arg1
  NODNO=0
  *DO,JJ,1,NNODES
     NODNO=NDNEXT(NODNO)
     *GET,NODERF,NODE,NODNO,F,FY
     F,NODNO,FY,NODERF/CGERF
  *ENDDO
  NSEL,ALL
  ESEL,ALL
  /SOLU
  ANTYPE,,REST
  TimeVar=0.45+(J+1)*0.05/(NCGECut+1)+(arg1-1)
  Time,TimeVar
  SOLVE
  SAVE
*ENDDO

/PREP7
SelCTNodes,arg1
NODNO=0
*DO,JJ,1,NNODES
   NODNO=NDNEXT(NODNO)
   *GET,NODERF,NODE,NODNO,F,FY
   FDELE,NODNO,FY
*ENDDO
NSEL,ALL
ESEL,ALL
/SOLU
```

```
ANTYPE,,REST
TimeVar=arg1-0.5
Time,TimeVar
SOLVE
```

APPENDIX A.7

ANSYS INPUT FILE *UNLOADCRACK.MAC*

```
! Unload model

AUTOTS,ON
*CFOPEN,%JN%_unload_%arg1%,dat
CurrLInc=UIBCC
StrsLvl=StrsMax/StrsMax
RStrs=StrsMax
*DO,JJ,1,arg1
  SelCTNodes,JJ
  NSEL,U,D,UY,0
  *GET,NSNODES,NODE,,COUNT
  *IF,NSNODES,GT,0,THEN
    NODNO=0
    *DO,JJJ,1,NSNODES
       NODNO=NDNEXT(NODNO)
       NODYSTRS=UY(NODNO)
       *IF,MTYPE,EQ,'SC',THEN
         CSYS,11
         NDANG=NY(NODNO)
         NXLOC=NX(NODNO)
         CSYS,0
       *ELSE
         NDANG=NZ(NODNO)
         NXLOC=NX(NODNO)
       *ENDIF
       NodeStat=0
       *VWRITE,NODNO,NXLOC,NDANG,StrsLvl,NODYSTRS,NODESTAT,RSTRS
          (F6.0,2X,E12.6,2X,E10.4,2X,F8.6,2X,E12.6,2X,F4.0,2X,E12.6)
    *ENDDO
  *ENDIF
*ENDDO

*DO,J,1,1/UIDCC
  TimeVar=TimeVar+CurrLInc*0.5
  RStrs=StrsMax-(StrsMax-StrsMin)*(TimeVar+0.5-arg1)/0.5
  *IF,TimeVar,GE,arg1,Then
    RStrs=StrsMin
    TimeVar=arg1
    Time,TimeVar
    AppLoad,height,RStrs
    SOLVE
    SAVE
    *DO,JJ,1,arg1
    SelCTNodes,JJ
    NSEL,U,D,UY,0
    *GET,NSNODES,NODE,,COUNT
    *IF,NSNODES,GT,0,THEN
      NODNO=0
      *DO,JJJ,1,NSNODES
        NodeStat=0
        NODNO=NDNEXT(NODNO)
        NODYSTRS=UY(NODNO)
        *IF,MTYPE,EQ,'SC',THEN
          CSYS,11
          NDANG=NY(NODNO)
          NXLOC=NX(NODNO)
          CSYS,0
        *ELSE
          NDANG=NZ(NODNO)
          NXLOC=NX(NODNO)
        *ENDIF
```

```
        *IF,NodYStrs*1e10,LT,0,THEN
          CurrLInc=UIDCC
          OPENSTAT=1
          NODESTAT=1
          D,NODNO,UY,0
        *ENDIF
        *VWRITE,NODNO,NXLOC,NDANG,StrsLvl,NODYSTRS,NodeStat,RStrs
           (F6.0,2X,E12.6,2X,E10.4,2X,F8.6,2X,E12.6,2X,F4.0,2X,E12.6)
      *ENDDO
      *endif
      *enddo
    *EXIT
  *ENDIF
  Time,TimeVar
  StrsLvl=RStrs/StrsMax
  AppLoad,height,RStrs
  SOLVE
  SAVE
  ClearRST,BDrive,BDir,''
  OPENSTAT=0
  OpnRwCnt=0
  *DO,JJ,1,arg1
    SelCTNodes,JJ
    NSEL,U,D,UY,0
    *GET,NSNODES,NODE,,COUNT
    *IF,NSNODES,GT,0,THEN
      NODNO=0
      *DO,JJJ,1,NSNODES
        NodeStat=0
        NODNO=NDNEXT(NODNO)
        NODYSTRS=UY(NODNO)
        *IF,MTYPE,EQ,'SC',THEN
          CSYS,11
          NDANG=NY(NODNO)
          NXLOC=NX(NODNO)
          CSYS,0
        *ELSE
          NDANG=NZ(NODNO)
          NXLOC=NX(NODNO)
        *ENDIF
        *IF,NodYStrs*1e10,LT,0,THEN
          CurrLInc=UIDCC
          OPENSTAT=1
          NODESTAT=1
          D,NODNO,UY,0
        *ENDIF
        *VWRITE,NODNO,NXLOC,NDANG,StrsLvl,NODYSTRS,NodeStat,RStrs
           (F6.0,2X,E12.6,2X,E10.4,2X,F8.6,2X,E12.6,2X,F4.0,2X,E12.6)
      *ENDDO
    *ELSEIF,NSNODES,EQ,0,THEN
      OpnRwCnt=OpnRwCnt+1
    *ENDIF
  *ENDDO
  NSEL,ALL
  *IF,OPENSTAT,EQ,1,THEN
    Time,Timevar+CurrLInc*0.01
    SOLVE
    SAVE
    ClearRST,BDrive,BDir,''
  *ENDIF
  *IF,OpnRwCnt,EQ,arg1,THEN
```

```
    CurrLInc=UIACC
  *ENDIF
*ENDDO
*CFCLOSE
```

APPENDIX A.8

ANSYS INPUT FILE *LOADCRACK.MAC*

```
! Load model

AUTOTS,ON
*CFOPEN,%JN%_load_%arg1%,dat
CurrLInc=LIBCO
StrsLvl=StrsMin/StrsMax
RStrs=StrsMin
*DO,JJ,1,arg1
  SelCTNodes,JJ
  NSEL,R,D,UY,0
  *GET,NSNODES,NODE,,COUNT
  *IF,NSNODES,GT,0,THEN
    NODNO=0
    *DO,JJJ,1,NSNODES
       NODNO=NDNEXT(NODNO)
       *GET,NODYSTRS,NODE,NODNO,S,Y
       *IF,MTYPE,EQ,'SC',THEN
         CSYS,11
         NDANG=NY(NODNO)
         NXLOC=NX(NODNO)
         CSYS,0
       *ELSE
         NDANG=NZ(NODNO)
         NXLOC=NX(NODNO)
       *ENDIF
       NodeStat=0
       *VWRITE,NODNO,NXLOC,NDANG,StrsLvl,NODYSTRS,NODESTAT,RSTRS
          (F6.0,2X,E12.6,2X,E10.4,2X,F8.6,2X,E12.6,2X,F4.0,2X,E12.6)
    *ENDDO
  *ENDIF
*ENDDO
*DO,J,1,1/LIDCO
  TimeVar=TimeVar+CurrLInc*0.45
  RStrs=(StrsMax-StrsMin)*(TimeVar-arg1)/0.45+StrsMin
  *IF,TimeVar,GE,arg1+0.45,Then
    RStrs=StrsMax
    TimeVar=arg1+0.45
    Time,TimeVar
    AppLoad,height,RStrs
    SOLVE
    SAVE
    *EXIT
  *ENDIF
  Time,TimeVar
  StrsLvl=RStrs/StrsMax
  AppLoad,height,RStrs
  SOLVE
  SAVE
  ClearRST,BDrive,BDir,''
  OPENSTAT=0
  OpnRwCnt=0
  *DO,JJ,1,arg1
    SelCTNodes,JJ
    NSEL,R,D,UY,0
    *GET,NSNODES,NODE,,COUNT
    *IF,NSNODES,GT,0,THEN
      NODNO=0
      *DO,JJJ,1,NSNODES
        NodeStat=0
        NODNO=NDNEXT(NODNO)
        *GET,NodYStrs,NODE,NODNO,S,Y
```

```
        *IF,MTYPE,EQ,'SC',THEN
          CSYS,11
          NDANG=NY(NODNO)
          NXLOC=NX(NODNO)
          CSYS,0
        *ELSE
          NDANG=NZ(NODNO)
          NXLOC=NX(NODNO)
        *ENDIF
        *IF,NodYStrs*1e10,GT,0,THEN
          CurrLInc=LIDCO
          OPENSTAT=1
          NODESTAT=1
          DDELE,NODNO,UY
        *ENDIF
        *VWRITE,NODNO,NXLOC,NDANG,StrsLvl,NODYSTRS,NodeStat,RStrs
           (F6.0,2X,E12.6,2X,E10.4,2X,F8.6,2X,E12.6,2X,F4.0,2X,E12.6)
      *ENDDO
    *ELSEIF,NSNODES,EQ,0,THEN
      OpnRwCnt=OpnRwCnt+1
    *ENDIF
  *ENDDO
  NSEL,ALL
  *IF,OPENSTAT,EQ,1,THEN
    Time,Timevar+CurrLInc*0.01
    SOLVE
    SAVE
    ClearRST,BDrive,BDir,''
  *ENDIF
  *IF,OpnRwCnt,EQ,arg1,THEN
    CurrLInc=LIACO
  *ENDIF
*ENDDO
*CFCLOSE
```

APPENDIX A.9

ANSYS INPUT FILE *APPLOAD.MAC*

```
! Apply loading boundary conditions

   NSEL,S,LOC,Y,arg1
   SF,ALL,PRES,-arg2
   NSEL,ALL
```

APPENDIX A.10

ANSYS INPUT FILE *CLEARRST.MAC*

```
!  This input file removes unnecessary files to save disk space

SAVE
FINISH
/COPY,,ESAV,,,,%arg1%%arg2%%arg3%
/COPY,,MNTR,,,,%arg1%%arg2%%arg3%
/COPY,,DB,,,,%arg1%%arg2%%arg3%
/DELETE,,RST
/SOLU
ANTYPE,,REST
```

APPENDIX A.11

ANSYS INPUT FILE *SELCTNODES.MAC*

```
! This input file selects the crack-tip nodes

  NSEL,S,LOC,Y,0
  NSEL,R,LOC,X,c+da*(arg1-1.25),c+da*(arg1-0.75)
  NSEL,U,LOC,X,0,c-da*0.25
```

APPENDIX B

INPUT FILE *HOLE.DAT*

```
/BATCH
! This is a sample closure input file.
!
! This runs the script "appbcs.mac"  to import the Solid
! Elements and Nodes from the files "jobname.crd" & "jobname.elm",
! and applies necessary boundary conditions for Crack Growth
! It then calls "strtcyc.mac"  to run growth analysis.


!Note all lengths are in in, and pressures in Psi

/CONFIG,NPROC,1         ! Set the number of Processors to use in
                        ! analysis


!Loading information:

StrsMax=18              ! Maximum Applied Stress
StrsMin=1.8             ! Minimum Applied Stress
NLC=100                 ! Total Number of Loading Cycles to execute

!Geometry Information:

MTYPE='TC'

t=0.25                 ! Thickness of plate
w=3.0                  ! Plate Half-Width
height=12              ! Model Height
c=.041677              ! Initial Crack half-length
da=0.00083333          ! Crack Growth Increment

!  Crack Growth Options:

NCGECut=5              ! Number of bisections to matrix stiffness
                       ! before death
CGERF=2                ! Crack Growth Element Stiffness Reduction
                       ! Factor

*get,JN,ACTIVE,,JOBNAM
/TITLE, Plasticity Induced Closure of model %JN%

AppBCs                ! Import Solid model and apply BC's

BDrive='E:'           ! Drive for file backups
BDir='\backup'        ! Directory for file backups
MaxDir='\maxload'     ! Directory for backup at Max Load
MinDir='\minload'     ! Directory for Backup at Min Load

!Solution Information:

/SOLU                  ! Enter Solution Processor

LIBCO=0.005            ! Loading Increment before crack opening
LIDCO=0.025            ! Loading Increment during crack opening
LIACO=0.10             ! Loading Increment after crack opening
UIBCC=0.005            ! Un-load Increment before crack closing
UIDCC=0.025            ! Un-load Increment during crack closing
UIACC=0.10             ! Un-load Increment after crack closing
SOLCONTROL,ON
NSUB,1
NEQIT,8                ! Number of Equillibrium Iterations before
```

```
                       ! bisection
NROPT,FULL,,ON         ! Full Newton Rapson Option, Adaptive Descent ON
EQSLV,PCG,1.0e-8       ! Use the Pre-Conditioned Conjugate Solver (In
                       ! Core, default tolerance)

RESCONTROL,DEFINE,NONE,NONE,0   !  Set Resume Controls to act like
                                !  ANSYS 5.5.3 and below
                                !  (Single Frame Restart)
StrtCyc
```

APPENDIX C

FORTRAN PROGRAM *SUPERPOSITION.FOR*

```
program superposition

implicit none

integer :: i,j,k,allocatestatus,n,n1,c,curve,ii,jj,iii,nn, kk
real ::
filestatus,iostat,os,p,smax,r,w,pi,ai,da_avg,smin,xposition,r_sif
real, dimension(:), allocatable::xpos
real, dimension(:), allocatable::rsif
real, dimension(:), allocatable::a
real, dimension(:), allocatable::f1
real, dimension(:), allocatable::f2
real, dimension(:), allocatable::kmax
real, dimension(:), allocatable::kmin
real, dimension(:), allocatable::x
real, dimension(:), allocatable::y
real, dimension(:), allocatable::f3
real, dimension(:), allocatable::f4
real, dimension(:), allocatable::dk
real, dimension(:), allocatable::dadn
real, dimension(:), allocatable::dn
real, dimension(:), allocatable::m
real, dimension(:), allocatable::da
real, dimension(:), allocatable::Ktot
real, dimension(:), allocatable::a2
real, dimension(:), allocatable::mink
real, dimension(:), allocatable::maxk
real, dimension(:), allocatable::kmaxf
real, dimension(:), allocatable::kminf


open (10, file = 'xpos.txt')
open (20, file = 'rsif.txt')
open (30, file = 'a_vs_N.txt')
open (40, file = 'K.txt')


!Parameters--------------------------------------------
pi = 3.14159265359       !pi
r = 0.172                !radius 7075
w = 0.85                 !width 7075
smax = 17.0              !max stress 7075
smin = 1.7               !min stress 7075
ai =  0.18244669         !initial crack length 7075
kk =  5                  !number to start crack growth 7075

!Building opening stress array-------------------------

n=0
do while (filestatus.ge.0)
  n=n+1
  read (10,*, iostat=filestatus) xposition
end do
filestatus=0
n=n-1
rewind(10)

allocate (a(n), stat = allocatestatus)
if(allocatestatus/=0) stop "*** not enough memory"
do j=1,n
 read (10,*, iostat = filestatus) xposition
```

```
 a(j) = xposition-r
end do
filestatus = 0

allocate (rsif(n))

do jj=1,n
 read (20,*,iostat = filestatus) r_sif
 rsif(jj) = r_sif
enddo
filestatus = 0

!Allocate each array
allocate (f1(n))
allocate (f2(n))
allocate (kmax(n))
allocate (kmin(n))
allocate (x(n))
allocate (y(n))
allocate (f3(n))
allocate (f4(n))
allocate (dk(n))
allocate (dadn(n))
allocate (dn(n))
allocate (m(n))
allocate (Ktot(n))
allocate (da(n))
allocate (maxk(n))
allocate (mink(n))
allocate (kmaxf(n))
allocate (kminf(n))

!Calculate Kmax-Kmin
do i = kk,n-1
   x(i) = a(i)/w
   y(i) = (2.*r + a(i))/(2.*w - a(i))
   f1(i) =
((1./cos(pi*y(i)/2.))*(sin(2.*x(i)*y(i)))/(2.*x(i)*y(i)))**0.5
   f3(i) = (1.+ 0.2*(1.-a(i)/(r+a(i)))+0.3*(1.-a(i)/(r+a(i)))**6.)
   f4(i) = (2.243-2.64*(a(i)/(a(i)+r))+1.352*(a(i)/(a(i)+r))**2. -
0.248*(a(i)/(a(i)+r))**3.)
   f2(i) = f3(i) * f4(i)
   kmax(i) = smax*f1(i)*f2(i)*(pi*a(i))**0.5
   kmin(i) = smin*f1(i)*f2(i)*(pi*a(i))**0.5
   kmaxf(i) = kmax(i)+rsif(i)
   kminf(i) = kmin(i)+rsif(i)
   maxk(i) = kmaxf(i)
   if (kminf(i).le.0.00) then
   mink(i) = 0
   elseif (kminf(i).gt.0.00) then
   mink(i) = kminf(i)
   endif
   dk(i) = maxk(i) - mink(i)
   Ktot(i) = dk(i)
enddo


!Get dadN values for a certain K
! 7075 R = 0.1 data
do i = kk,n-1
 if (Ktot(i).lt.0.000001) then
```

```
    dadn(i) = 0
  endif
  if (Ktot(i).gt.0.000001) then
   if (Ktot(i).lt.3.06157) then
    dadn(i) = (3.6001637055102195788E-9)*Ktot(i)**3.7753570920855481331
   end if
   endif
  if (Ktot(i).gt.3.06157)then
   if (Ktot(i).lt.5.04442) then
    dadn(i) = (3.8655666925373295767E-9)*Ktot(i)**3.7117882262959823065
  end if
  end if
   if (Ktot(i).gt.5.04442) then
     if (Ktot(i).lt.7.79052) then
    dadn(i) = (4.2743000283776062390E-9)*Ktot(i)**3.6496779033280770548
  end if
   end if
   if (Ktot(i).gt.7.79052) then
     if (Ktot(i).lt.14.2105) then
    dadn(i) = (1.0424245595944371450E-7)*Ktot(i)**2.0937875530623353076
  end if
   end if
   if (Ktot(i).gt.14.2105) then
     if (Ktot(i).lt.23.4141) then
    dadn(i) = (5.8578961966810011868E-9)*Ktot(i)**3.1785461952732473597
  end if
   end if
   if (Ktot(i).gt.23.4141) then
     if (Ktot(i).lt.42.3161) then
    dadn(i) = (1.4109544414103275845E-10)*Ktot(i)**4.3601857019054836678
  end if
   end if
   if (Ktot(i).gt.42.3161) then
      dadn(i) = (1.7083361133003553886E-
14)*Ktot(i)**6.7683788157138267703
   end if
end do


!Determine dN
do i = kk,n-2
   iii=i+1
   da(i) = a(iii)-a(i)
   dn(i) = da(i)/dadn(i)

end do

!Determine N
m(kk)=0
do i = kk+1,n-1
 m(i) = m(i-1)+dn(i-1)
end do

!Write new file
do i = kk,n
   write (40,300) Ktot(i),kmax(i),kmaxf(i),kmin(i),kminf(i),rsif(i)
end do

!Write a vs N file
do i = kk,n
   write (30,200) a(i),m(i)
```

```
end do

100 format (F12.6)
200 format (F10.8,2x,F10.2)
300 format (F12.6,2x,F12.6,2x,F12.6,2x,F12.6,2x,F12.6,2x,F12.6)

!----------------------------------------------------------------------
! 7075 R = 0.1 data
!do i = kk,n-1
! if (Ktot(i).lt.0.000001) then
!   dadn(i) = 0
! endif
! if (Ktot(i).gt.0.000001) then
!  if (Ktot(i).lt.3.06157) then
!   dadn(i) = (3.6001637055102195788E-9)*Ktot(i)**3.7753570920855481331
!  end if
!  endif
! if (Ktot(i).gt.3.06157)then
!  if (Ktot(i).lt.5.04442) then
!  dadn(i) = (3.8655666925373295767E-9)*Ktot(i)**3.7117882262959823065
!end if
! end if
!  if (Ktot(i).gt.5.04442) then
!    if (Ktot(i).lt.7.79052) then
!  dadn(i) = (4.2743000283776062390E-9)*Ktot(i)**3.6496779033280770548
!end if
!  end if
!  if (Ktot(i).gt.7.79052) then
!    if (Ktot(i).lt.14.2105) then
!  dadn(i) = (1.0424245595944371450E-7)*Ktot(i)**2.0937875530623353076
!end if
!  end if
!  if (Ktot(i).gt.14.2105) then
!    if (Ktot(i).lt.23.4141) then
!  dadn(i) = (5.8578961966810011868E-9)*Ktot(i)**3.1785461952732473597
!end if
!  end if
!  if (Ktot(i).gt.23.4141) then
!    if (Ktot(i).lt.42.3161) then
!  dadn(i) = (1.4109544414103275845E-10)*Ktot(i)**4.3601857019054836678
!end if
!  end if
!  if (Ktot(i).gt.42.3161) then
!      dadn(i) = (1.7083361133003553886E-
!14)*Ktot(i)**6.7683788157138267703
!  end if
!end do

!----------------------------------------------------------------------
! Liu R = 0.1 data
!do i = kk,n-1
! if (Ktot(i).lt.0.000001) then
!   dadn(i) = 0
! endif
! if (Ktot(i).gt.0.000001) then
!  if (Ktot(i).lt.5.88979) then
!   dadn(i) = (8.2248783344000964625E-
!10)*Ktot(i)**3.4644080789555000806
!  end if
```

```
!  endif
! if (Ktot(i).gt.5.88979)then
!  if (Ktot(i).lt.6.21131) then
!  dadn(i) = (2.2223576357757589016E-16)*Ktot(i)**11.993582407661507535
!end if
! end if
!  if (Ktot(i).gt.6.21131) then
!    if (Ktot(i).lt.6.60708) then
!  dadn(i) = (2.4220134526963299327E-17)*Ktot(i)**13.207220559052529877
!end if
!  end if
!  if (Ktot(i).gt.6.60708) then
!    if (Ktot(i).lt.8.28973) then
!  dadn(i) = (3.2714554198402574140E-10)*Ktot(i)**4.5115116836984765469
!end if
!  end if
!  if (Ktot(i).gt.8.28973) then
!    if (Ktot(i).lt.14.6018) then
!  dadn(i) = (1.7417431069694599106E-8)*Ktot(i)**2.6321788629067804637
!end if
!  end if
!  if (Ktot(i).gt.14.6018) then
!    if (Ktot(i).lt.18.5087) then
!  dadn(i) = (5.9903011649139389655E-9)*Ktot(i)**3.0302662015961060081
!end if
!  end if
!  if (Ktot(i).gt.18.5087) then
!    if (Ktot(i).lt.22.2222) then
!  dadn(i) = (5.8077782313222249554E-8)*Ktot(i)**2.2518378326302186854
!end if
!  end if
!  if (Ktot(i).gt.22.2222) then
!      dadn(i) = (4.2413966884494287592E-
!17)*Ktot(i)**9.0357619270291578330
!  end if
!end do
end program superposition
```

APPENDIX D

FORTRAN PROGRAM *CLOSURE.FOR*

```
program closure

implicit none

integer :: i,j,k,allocatestatus,n,n1,c,curve
real :: filestatus,iostat,os,p,smax,r,w,pi,ai,da,da_avg,smin
real, dimension(:,:), allocatable::ostress
real, dimension(:), allocatable::a
real, dimension(:), allocatable::f1
real, dimension(:), allocatable::f2
real, dimension(:), allocatable::kmax
real, dimension(:), allocatable::kmin
real, dimension(:), allocatable::x
real, dimension(:), allocatable::y
real, dimension(:), allocatable::f3
real, dimension(:), allocatable::f4
real, dimension(:), allocatable::dk
real, dimension(:), allocatable::dadn
real, dimension(:), allocatable::dn
real, dimension(:), allocatable::m

open (10, file = 'opening_stresses.txt')
open (20, file = 'newstress.txt')
open (30, file = 'a_vs_N.txt')


!Parameters-------------------------------------------
pi = 3.14159265359        !pi
da_avg = 0.008333333      !crack growth increment
r = 0.375                 !radius
w = 3.0                   !width
smax = 15.0               !max stress
smin = 1.5                !min stress
ai = 0.075                !initial crack length

!Building opening stress array-------------------------
n=0
do while (filestatus.ge.0)
  n=n+1
  read (10,*, iostat=filestatus) c,os
end do
filestatus=0
n=n-1
rewind(10)

allocate (ostress(n,2), stat = allocatestatus)
if(allocatestatus/=0) stop "*** not enough memory"
do j=1,n
 read (10,*, iostat = filestatus) c,os
 ostress(j,1) = c
 ostress(j,2) = os*smax
end do
filestatus = 0

!Allocate each array
allocate (a(n))
allocate (f1(n))
allocate (f2(n))
allocate (kmax(n))
allocate (kmin(n))
allocate (x(n))
```

```
allocate (y(n))
allocate (f3(n))
allocate (f4(n))
allocate (dk(n))
allocate (dadn(n))
allocate (dn(n))
allocate (m(n))

!Build crack length array
a(1) = ai
do i = 2,n
  a(i)=a(i-1)+da_avg
end do

!Calculate Kmax-Kop
do i = 1,n
   x(i) = a(i)/w
   y(i) = (2.*r + a(i))/(2.*w - a(i))
   f1(i) =
((1./cos(pi*y(i)/2.))*(sin(2.*x(i)*y(i)))/(2.*x(i)*y(i)))**0.5
   f3(i) = (1.+ 0.2*(1.-a(i)/(r+a(i)))+0.3*(1.-a(i)/(r+a(i)))**6.)
   f4(i) = (2.243-2.64*(a(i)/(a(i)+r))+1.352*(a(i)/(a(i)+r))**2. -
0.248*(a(i)/(a(i)+r))**3.)
   f2(i) = f3(i) * f4(i)
   kmax(i) = smax*f1(i)*f2(i)*(pi*a(i))**0.5
   kmin(i) = ostress(i,2)*f1(i)*f2(i)*(pi*a(i))**0.5
   dk(i) = kmax(i) - kmin(i)
end do


!Determine da/dN from the dK

!liu2024

do i = 1,n
  if (dk(i).lt.3.8881) then
   dadn(i) = (1.2516991843250658322E-8)*dk(i)**2.4337176952303115773
 end if
 if (dk(i).gt.3.8881)then
  if (dk(i).lt.4.4638) then
   dadn(i) = (1.8650305810915970984E-11)*dk(i)**7.2270611156921409686
 end if
  end if
 if (dk(i).gt.4.4638)then
  if (dk(i).lt.11.5835) then
   dadn(i) = (3.6151593949425293280E-9)*dk(i)**3.7063210087488959679
 end if
  end if
 if (dk(i).gt.11.5835)then
  if (dk(i).lt.15.0707) then
   dadn(i) = (7.8178007785933744611E-12)*dk(i)**6.2114377037553923683
 end if
  end if
 if (dk(i).gt.15.0707)then
  if (dk(i).lt.16.2184) then
   dadn(i) = (1.2754555625174362996E-16)*dk(i)**10.275000595874495665
 end if
  end if
  if (dk(i).gt.16.2184) then
      dadn(i) = (1.9670205027375100520E-
28)*dk(i)**20.036800826570233243
```

```
  end if
end do

!Determine dN
do i = 1,n
  dn(i) = da_avg/dadn(i)
end do

!Determine N
m(1)=0
do i = 2,n
  m(i) = m(i-1)+dn(i-1)
end do

!Write new file
do i = 1,n
  write (20,100)
int(ostress(i,1)),ostress(i,2),a(i),dk(i),dadn(i),dn(i)
end do

!Write a vs N file
do i = 1,n
  write (30,200) a(i),m(i)
end do


100 format (I5,2x,F10.6,2x,F8.6,2x,F12.6,2x,F12.10,2x,F12.6)
200 format (F10.8,2x,F10.2)


!-----------------------------------------------------------------------
! liu2024

!do i = 1,n
!  if (dk(i).lt.3.8881) then
!   dadn(i) = (1.2516991843250658322E-8)*dk(i)**2.4337176952303115773
! end if
! if (dk(i).gt.3.8881)then
!  if (dk(i).lt.4.4638) then
!   dadn(i) = (1.8650305810915970984E-11)*dk(i)**7.2270611156921409686
!end if
!  end if
! if (dk(i).gt.4.4638)then
!  if (dk(i).lt.11.5835) then
!   dadn(i) = (3.6151593949425293280E-9)*dk(i)**3.7063210087488959679
!end if
!  end if
! if (dk(i).gt.11.5835)then
!  if (dk(i).lt.15.0707) then
!   dadn(i) = (7.8178007785933744611E-12)*dk(i)**6.2114377037553923683
!end if
!  end if
! if (dk(i).gt.15.0707)then
!  if (dk(i).lt.16.2184) then
!   dadn(i) = (1.2754555625174362996E-16)*dk(i)**10.275000595874495665
!end if
!  end if
!  if (dk(i).gt.16.2184) then
!      dadn(i) = (1.9670205027375100520E-
!28)*dk(i)**20.036800826570233243
!  end if
```

```
!end do

!------------------------------------------------------------------------
! nasgro2024
!
!do i = 1,n
! if (dk(i).lt.0.956) then
!   dadn(i) = (5.76302247289734E-09)*dk(i)**8.45119132609073
! end if
! if (dk(i).gt.0.956)then
!   if (dk(i).lt.1.866) then
!  dadn(i) = (4.81941267722244E-09)*dk(i)**4.47740310932126
!   end if
! end if
! if (dk(i).gt.1.866) then
!   if (dk(i).lt.3.64) then
!  dadn(i) = (2.15602474161943E-08)*dk(i)**2.07566198802441
!   end if
! end if
! if (dk(i).gt.3.64) then
!   if (dk(i).lt.7.007) then
!     dadn(i) = (2.15711640297585E-09)*dk(i)**3.85747929981341
!   end if
! end if
! if (dk(i).gt.7.007) then
!   if (dk(i).lt.12.286) then
!     dadn(i) = (1.34422738189659E-09)*dk(i)**4.10040424528682
!   end if
! end if
! if (dk(i).gt.12.286) then
!   if (dk(i).lt.20.931) then
!     dadn(i) = (7.71201447804285E-10)*dk(i)**4.32190467977269
!   end if
! end if
!  if (dk(i).gt.20.931) then
!   if (dk(i).lt.32.762) then
!     dadn(i) = (6.42109549554029E-11)*dk(i)**5.13926302256757
!   end if
! end if
! if (dk(i).gt.32.762) then
!   dadn(i) = (2.96915473619974E-11)*dk(i)**5.36031544399855
! end if
!end do


!------------------------------------------------------------------------
! nasgro7075

!Determine da/dN from the dK

!do i = 1,n
!if (dk(i).lt.1.065) then
!   dadn(i) = (7.962769157E-13)*dk(i)**135.081721
! end if
! if (dk(i).gt.1.065)then
!   if (dk(i).lt.1.119) then
!  dadn(i) = (0.00000000507628578)*dk(i)**32.539778
!   end if
! end if
! if (dk(i).gt.1.119) then
```

```
!   if (dk(i).lt.1.219) then
!  dadn(i) = (0.00000000792579391)*dk(i)**8.097947
!   end if
! end if
! if (dk(i).gt.1.219) then
!   if (dk(i).lt.1.456) then
!     dadn(i) = (0.00000001822083195)*dk(i)**3.894344
!   end if
! end if
! if (dk(i).gt.1.456) then
!   if (dk(i).lt.2.229) then
!     dadn(i) = (0.00000003502888531)*dk(i)**2.154606
!   end if
! end if
! if (dk(i).gt.2.229) then
!   if (dk(i).lt.3.913) then
!     dadn(i) = (0.00000007339983517)*dk(i)**1.231711
!   end if
! end if
! if (dk(i).gt.3.913) then
!   if (dk(i).lt.4.823) then
!     dadn(i) = (1.17649515E-13)*dk(i)**11.012317
!   end if
! end if
! if (dk(i).gt.4.823) then
!   if (dk(i).lt.7.735) then
!     dadn(i) = (0.00000001850443025)*dk(i)**3.407232
!   end if
! end if
! if (dk(i).gt.7.735) then
!   if (dk(i).lt.10.465) then
!     dadn(i) = (0.0000001807908036)*dk(i)**2.293057
!   end if
! end if
! if (dk(i).gt.10.465) then
!   if (dk(i).lt.18.2) then
!     dadn(i) = (0.0000000001195397501)*dk(i)**5.411172
!   end if
! end if
! if (dk(i).gt.18.2) then
!   if (dk(i).lt.38.858) then
!     dadn(i) = (0.0000000002482465762)*dk(i)**5.159304
!   end if
! end if
! if (dk(i).gt.38.858) then
!   dadn(i) = (9.854027554E-11)*dk(i)**5.411757
! end if
!end do

!----------------------------------------------------------------------
! mil handbook 7075 dKeff

!do i = 1,n
!if (dk(i).lt.7.42) then
!   dadn(i) = (2.0308144566525793683E-9)*dk(i)**3.9862086145233756815
! end if
! if (dk(i).gt.7.42)then
!   if (dk(i).lt.8.712) then
!  dadn(i) = (2.8064816245427991840E-12)*dk(i)**7.2714742549587944899
!   end if
! end if
```

```
! if (dk(i).gt.8.712) then
!   if (dk(i).lt.10.807) then
!  dadn(i) = (2.0479185972744096299E-9)*dk(i)**4.2259513603036082931
!   end if
! end if
! if (dk(i).gt.10.807) then
!   if (dk(i).lt.14.153) then
!     dadn(i) = (1.2517377091254135383E-8)*dk(i)**3.4653856953817474214
!   end if
! end if
! if (dk(i).gt.14.153) then
!   if (dk(i).lt.18.293) then
!     dadn(i) = (5.3659002704175143222E-8)*dk(i)**2.9161133478037422843
!   end if
! end if
! if (dk(i).gt.18.293) then
!   if (dk(i).lt.22.697) then
!     dadn(i) = (7.9593623010622689358E-9)*dk(i)**3.5726721804311167488
!   end if
! end if
! if (dk(i).gt.22.697) then
!   if (dk(i).lt.24.265) then
!     dadn(i) = (7.0543131267657414534E-
!15)*dk(i)**8.0362151974975716003
!   end if
! end if
! if (dk(i).gt.24.265) then
!   dadn(i) = (3.2151519104560811807E-14)*dk(i)**7.5605789979999724738
! end if
!end do


end program closure
```