Fatigue Crack Growth Analysis with Finite Element Methods and a Monte Carlo Simulation

Joshua H. Melson

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

Master of Science
In
Mechanical Engineering

Robert L. West, Chair

Norman E. Dowling, Co-Chair

John M. Kennedy

May 2, 2014

Blacksburg, VA

Keywords: Fracture, eXtended Finite Element Method,
Phantom Nodes, Monte Carlo

Fatigue Crack Growth Analysis with Finite Element Methods and a Monte Carlo Simulation

Joshua H. Melson

Abstract

Fatigue crack growth in engineered structures reduces the structures load carrying capacity and will eventually lead to failure. Cycles required to grow a crack from an initial length to the critical length is called the fatigue fracture life. In this thesis, five different methods for analyzing the fatigue fracture life of a center cracked plate were compared to experimental data previously collected by C.M. Hudson in a 1969 NASA report studying the R-ratio effects on crack growth in 7075-T6 aluminum alloy. The Paris, Walker, and Forman fatigue crack growth models were fit the experimental data. The Walker equation best fit the data since it incorporated R-ratio effects and had a similar Root Mean Square Error (RMSE) compared to the other models. There was insufficient data in the unstable region of crack growth to adequately fit the Forman equation.

Analytical models were used as a baseline for all fatigue fracture life comparisons. Life estimates from AFGROW and finite elements with mid-side nodes moved to their quarter point location compared very with the analytical model with errors less than 3%. The Virtual Crack Closure Technique (VCCT) was selected as a method for crack propagation along a predefined path. Stress intensity factors (SIFs) for shorter crack lengths were found to be low, resulting in an overestimated life of about 8%. The eXtended Finite Element Method with Phantom Nodes (XFEM-PN) was used, allowing crack propagation along a solution dependent path, independent of the mesh. Low SIFs throughout growth resulted in life estimates 20% too large. All finite element analyses were performed in Abaqus 6-13.3. An integrated polynomial method was developed for calculating life based on Abaqus' results, leading to coarser meshes with answers closer to the analytical estimate. None of the five methods for estimating life compared well with the experimental data, with analytical errors on life ranging from 10-20%. These errors were attributed to the limited number of crack growth experiments run at each R-ratio, and the large variability typically seen in growth rates.

Monte Carlo simulations were run to estimate the distribution on life. It was shown that material constants in the Walker model must be selected based on their interrelation with a multivariate normal probability density function. Both analytical and XFEM-PN simulations had similar coefficients of variation on life of approximately 3% with similar normal distributions.

It was concluded that Abaqus' XFEM-PN is a reasonable means of estimating fatigue fracture life and its variation, and this method could be extended to other geometries and three-dimensional analyses.

Acknowledgments

I would like to thank my adviser and committee chair, Dr. Bob West, for his technical guidance, availability to answer questions, and patience throughout my graduate career. Without his support, this would not have been possible.

I would also like to thank my committee co-chair, Dr. Norman Dowling, who's technical expertise in fracture mechanics was invaluable. He was always willing to meet and clear up any questions encountered along the way.

Thank you to my committee member, Dr. John Kennedy, who helped guide me through undergraduate and graduate school.

Thanks to The MathWorks, the College of Engineering, and the Mechanical Engineering Department of Virginia Tech for providing funding to pursue my master's degree.

Contents

List of Tables

List of Figures

Nomenclature

α	Ratio of half crack length to half width of plate
$\alpha,\ \beta$	Empirical exponents in the Power law
$\boldsymbol{\mu}$	Mean coefficient vector
$\boldsymbol{\rho}$	Correlation coefficient matrix
$\boldsymbol{b}_{ji}$	Coefficient vector
$\boldsymbol{u}$	Displacement vector
ΔK	Stress intensity range
$\Delta\, a$	Element length at crack tip
δ_{ij}	Kronecker delta
$\epsilon_{max},\ \epsilon_{min}$	Maximum and minimum strain
η	Empirical exponent in the BK law
η_1	Empirical exponent in the Reeder law
Γ	Contour surrounding the crack tip
$\mathcal{G}$	Linear elastic energy release rate
$\mathcal{G}_c$	Critical linear elastic energy release rate
$\mathcal{G}_{Ic},\ \mathcal{G}_{IIc},\ \mathcal{G}_{IIIc}$	Mode dependent critical energy release rates
$\mathcal{G}_I,\ \mathcal{G}_{II},\ \mathcal{G}_{III}$	Mode dependent energy release rates
$\mathcal{G}_{pl}$	Energy release rate at Paris limit
$\mathcal{G}_{th}$	Threshold energy release rate
ν	Poisson's ratio
ω	Angular frequency

$\overline{\Delta K}$	Equivalent stress intensity range at $R = 0$
$\overline{R}$	Nodal residual in Abaqus' direct cyclic solver
$\overline{u}$	Nodal displacements in Abaqus' direct cyclic solver
$\phi,\ \psi$	Level set functions
$\sigma,\ \tau$	Far field normal and shear stress
$\sigma^2,\ \mathbf{\Sigma}$	Variance and covariance matrix
$\sigma_{max},\ \sigma_{min}$	Maximum and minimum stress
$\sigma_m,\ \sigma_a$	Mean and alternating stress
$\sigma_x,\ \sigma_y,\ \sigma_z$	Normal stress in the x-, y-, and z-direction
$\sigma_{yld},\ \sigma_{ult}$	Yield and ultimate stress
$\tau_{xy},\ \tau_{yz},\ \tau_{xz}$	Shear stress in the x-y, y-z, and x-z plane
ε	Random error
a	Half crack length
a_c	Critical half crack length at fracture
a_f	Final measured half crack length at fracture
a_i	Initial half crack length
$C,\ m$	Coefficients for the Paris fatigue crack growth model
$C_0,\ m,\ \gamma$	Coefficients for the Walker fatigue crack growth model
$c_1,\ c_2$	Coefficients for Abaqus' onset of crack growth model
$C_2,\ m_2$	Coefficients for the Forman fatigue crack growth model
$c_3,\ c_4$	Coefficients for Abaqus' fatigue crack growth model
$C_3,\ m_3$	Coefficients for Broek and Rice's fatigue crack growth model
$CR_0,\ CR_n$	Ratio of the largest residual coefficient to the time average force for the constant and periodic terms, respectively
$CU_0,\ CU_n$	Ratio of the largest correction to the displacement coefficients over the largest displacement coefficient for the constant and periodic terms, respectively
da/dN	Fatigue crack growth rate

E	Young's modulus
F	Force component; Dimensionless geometry factor
F_i	Enrichment terms for the finite element shape functions
G	Shear modulus
H	Jump function; Heaviside function
I_0	Frequency at which residuals of successive iterations are checked
I_A	Maximum number of cutbacks allowed in an increment
I_R	Starting point for the logarithmic rate of convergence checks
J	Non-linear elastic energy release rate
J_c	Critical non-linear elastic energy release rate
K	Stress intensity factor
K_c	Critical stress intensity factor
$K_{Ic},\ K_{IIc},\ K_{IIIc}$	Mode dependent critical stress intensity factor
$K_I,\ K_{II},\ K_{III}$	Mode dependent stress intensity factors
$K_{max},\ K_{min}$	Maximum and minimum stress intensity factor, respectively
K_{th}	Threshold stress intensity factor
N	Fatigue cycles
N_f	Fatigue fracture life
$N_i(\boldsymbol{x})$	Finite element shape functions
n_j	Outward normal vector to contour Γ
P_f	Final load at fracture
Q	Forman plotting variable
r	Radial distance from crack tip; Ratio for modified Simpson's rule
$r,\ \theta$	Axes in a polar coordinate system
R_0	Unknown constant residual coefficient
r_e	Enrichment zone radius
$R_k^s\ R_k^c$	Periodic residual coefficient

$r_{p\sigma}$	Plane stress plastic zone radius
S	Set of nodes
u_0	Unknown constant displacement coefficient
u_k^s u_k^c	Periodic displacement coefficient
W	Stress work density
$x,\ y,\ z$	Axes in a rectangular coordinate system
b	Half width of plate specimen
h	Half height of plate specimen
t	Thickness of plate specimen; time
AFGROW	Air Force Growth fracture mechanics and fatigue crack growth analysis software tool
CCP	Center Crack Plate
CPD	Crack Propagation Direction
EPFM	Elastic Plastic Fracture Mechanics
FE	Finite Element
FEM	Finite Element Method
FEQP	Finite Elements with Quarter Points
LEFM	Linear Elastic Fracture Mechanics
LSM	Level Set Method
MERR	Maximum Energy Release Rate
MTS	Maximum Tangential Stress
SIF	Stress Intensity Factor
VCCT	Virtual Crack Closure Technique
XFEM	eXtended Finite Element Method
XFEM-PN	eXtended Finite Element Method with Phantom Nodes
CAE	Abaqus' Complete Abaqus Environment file extension
INP	Abaqus' analysis input file extension
ODB	Abaqus' output database file extension
STA	Abaqus' analysis status file extension

Chapter 1

Introduction

1.1 Overview and Statement of Need

Engineered structures can fail in various ways including yielding, buckling, and brittle fracture. In the presence of a flaw or stress raiser, cracks can form and grow to a critical length where the structure's strength has been reduced to a point where fracture occurs. When this occurs with the application of a cyclic load, the process is known as fatigue crack growth [1]. The cyclic loads are typically subcritical service loads that would not normally cause failure. Structures can be designed to minimize stress raisers, but flaws are inherent in all materials. Therefore, design and analysis of structures should take into account cracks that may form from these flaws.

Frequent inspections are used to look for cracks, and if a crack is found the component can be fixed or replaced as necessary. Inspection intervals could then be set at some fraction of the fatigue fracture life of the structure. Fatigue fracture life is calculated as the number of cycles required to grow a crack from a minimum detectable size to a critical size when the structure fails. Analytical fatigue crack growth models are available in the literature for many generalized geometries including an edge crack, center crack, and embedded penny crack just to name a few [2]. As geometry, loading and boundary conditions become more complex, the analytical equations quickly become more complicated and it may be difficult to account for all effects on crack propagation.

The Finite Element Method (FEM) has been used for decades to assist engineers in analyzing complex, cracked structures. Until recently, cracks had to be modeled as part of the structure's geometry. As the crack grew, the model would be rebuilt and remeshed, requiring significant user interaction or specialized programs. The eXtended Finite Element Method (XFEM) was developed in 1999 by Belytschko and Black [3], where cracks could be defined arbitrarily, independent of the mesh. A second iteration of the XFEM was introduced in 2006 by Song, Areias, and Belytschko [4], where Phantom Nodes (XFEM-PN) and a Level Set Method (LSM) were used to locate the crack. This method naturally lends itself to fatigue crack growth, where a crack propagates along a solution dependent path, independent of

the mesh. The commercial finite element code, Abaqus, employs XFEM-PN for automated crack growth in a low-cycle fatigue analysis [5]. A fundamental aspect of this thesis is to take a critical look at the code's ability to estimate fatigue fracture life.

Fatigue crack growth is highly stochastic, and the actual fatigue fracture life may be difficult to determine. For this reason, distributions on the life estimate are typically used. Monte Carlo simulations can be performed to model the complex interactions between the input distributions such as the variations in geometry, loading, and material properties. This thesis uses XFEM-PN as a tool for predicting the distribution of fatigue fracture life estimates based on the variability in the fatigue crack growth models..

1.2 Research Goals and Objectives

The primary goals of this thesis are to investigate the capabilities of the eXtended Finite Element Method with Phantom Nodes (XFEM-PN) for modeling crack propagation and estimating fatigue fracture life. Then, use XFEM-PN to quantify the distribution of fatigue fracture life. The following objectives were met to realize these goals:

1. Fit fatigue crack growth rate models to experimental data.
2. Analyze crack growth using methods established in the literature.
3. Analyze crack growth using XFEM-PN and compare results to the experimental data and established methods.
4. Combine the variability in the fitted crack growth rate models with the XFEM-PN to determine the distribution on fatigue fracture life estimates.

1.3 Scope of Thesis

Before the XFEM-PN can be used to model fatigue crack growth and estimate fatigue fracture life in everyday structures with complex geometry, complex loading and boundary conditions, material variability, and numerous other external factors, we must first focus on a simplified model with a known analytical solution. This thesis focuses on pure Mode I crack propagation in a Center Cracked Plate (CCP). The CCP offers simplified geometry, simplified boundary and loading conditions, and has been studied since the beginning of fracture mechanics.

The scope is further limited to Linear Elastic Fracture Mechanics (LEFM) in two dimensions. CCP crack propagation data was previously collected by Hudson [6] for 7075-T6 aluminum alloy, thus the analysis will be limited to the same homogeneous, isotropic material and specimen geometry. Monotonic oscillatory loads are analyzed with no consideration for contact. All finite element analyses were performed in Abaqus 6-13.3, unless otherwise stated, with no user defined subroutines.

A large number of stochastic variables are available for analyzing the variability of fatigue fracture life; however, the scope of this thesis is limited to the variability in the constants of the fitted fatigue crack growth models.

1.4 Thesis Outline

This thesis begins in Chapter 2 with a review of fracture mechanics and fatigue crack growth analysis procedures currently available in the literature. Fatigue crack growth models are fit to experimental data in Chapter 3. Resulting growth models are input into multiple fatigue crack growth analysis tools in Chapter 4 and compared to XFEM-PN. Chapter 5 uses the Monte Carlo method to analyze the variability in the life estimates. Finally, conclusions are made in Chapter 6 with a brief look at future work in three-dimensional fatigue crack growth. Additional data, results, and MATLAB and Python code can be found in the Appendices.

Chapter 2

Literature Review

The following literature review has been provided to give a brief background into fracture mechanics and fatigue crack growth laws that will be used in the remainder of this thesis. Established techniques in fatigue fracture analysis are explored, and examples of their use as they apply to this thesis are provided. Finally, a look at the Monte Carlo method, its assumptions, and applications to engineering analyses are given.

2.1 Material Models

The following sections give a brief background on fracture mechanics, fatigue crack growth models, methods used by Abaqus for fatigue crack growth analyses. These methods include approximating the critical mixed-mode energy release rates and rules for determining crack propagation directions which are used in the Abaqus analysis presented in Chapter 4.

2.1.1 Fracture Mechanics

Fracture mechanics is broadly divided into two types: Linear Elastic Fracture Mechanics (LEFM), and Elastic Plastic Fracture Mechanics (EPFM). LEFM assumes small deformations and minimal yielding at the crack tip, while EPFM can account for large deformations and plastic effects [1].

In LEFM, the Stress Intensity Factor (SIF), K, is a measure of the stress field at a crack tip and is calculated with Eq. (2.1) where F is a dimensionless geometry factor, σ is a remote nominal stress, and a is the crack length. This equation was developed from a theory of elasticity solution of the stress field around a sharp notch [7]. From the elastic solution it was concluded that the stress field is proportional to $1/\sqrt{r}$ where r is the radial distance from the crack tip. This stress proportionality results in a stress singularity at the crack tip where $r \to 0$.

$$K = F\sigma\sqrt{\pi a} \tag{2.1}$$

SIFs are divided into three modes based on the displacement at the crack tip, as shown in Fig. 2.1. Mode I, the opening mode, is caused by a displacement perpendicular to the crack plane, which is typically a result of tensile stresses. Mode I is primarily responsible for crack growth. Displacements perpendicular to the crack tip edge from in-plane shear stresses cause the sliding mode, Mode II. Out-of-plane shear stresses result in displacements parallel to the crack tip edge and a tearing mode, Mode III. To differentiate between modes, subscripts I, II, and III are appended to K. [8]

An energy approach to fracture mechanics was first explored by Griffith [9] in 1921 in a paper analyzing the brittle fracture of glass. His approach states that the energy required to grow a crack by an area, $t\,(da)$, is equal to the change in potential energy, dU, over the change in crack area as shown in Eq. (2.2) [7]. In LEFM, the potential energy is comprised of the internal strain energy.

$$\mathcal{G} \;=\; -\frac{1}{t}\frac{dU}{da} \tag{2.2}$$

His work was extended to metals by Irwin [10] in 1957, who showed that the strain energy release rate, $\mathcal{G}$, is related to K in LEFM using the Young's modulus, E, and Poisson's ratio, ν, in Eq. (2.3).

$$\mathcal{G} = \frac{K^2}{\overline{E}} \tag{2.3}$$

$$\overline{E} = \begin{cases} E & \text{for plane stress} \\ \dfrac{E}{1-\nu^2} & \text{for plane strain} \end{cases} \tag{2.4}$$

As described by Anderson [11], Rice [12] proposed the use of a line integral around the crack tip, called the J-integral, to characterize the strain energy release rate for EPFM when large deformations are present. J, as calculated in Eq. (2.5), assumes an idealized non-linear elastic material response. In this equation, W is the strain energy density, Γ is a contour

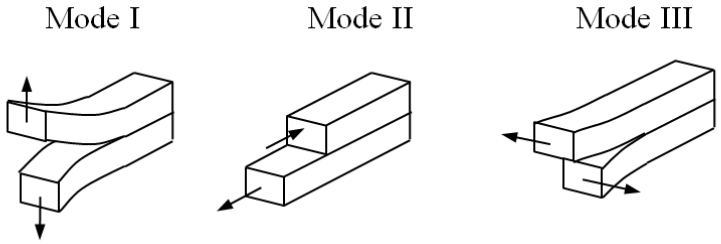

Fig. 2.1: Three modes of displacement at a crack tip.

surrounding the crack tip, n is the outward normal to this contour, u is the displacement, and δ is the Kronecker delta.

$$J = \int_{\Gamma} \left(W\delta_{1j} - \sigma_{ij} \frac{\partial u_i}{\partial x_1} \right) n_j \, d\Gamma \tag{2.5}$$

Shih and Asaro [13] related the J-integral to K for homogenous, isotropic materials using LEFM with the following equation, where $\overline{E}$ is calculated with Eq. (2.4) and G is the shear modulus.

$$J = \mathcal{G} = \frac{1}{\overline{E}} \left(K_I^2 + K_{II}^2 \right) + \frac{1}{2G} K_{III}^2 \tag{2.6}$$

Fracture is assumed to occur when K reaches the critical stress intensity factor, K_c, which is dependent upon the material and specimen geometry. Should the specimen be large enough that a state of plane strain exists, K_c will approach a minimum value called the plain strain critical stress intensity factor, or fracture toughness. The critical value concept can be extended to the critical energy release rate, $\mathcal{G}_c$. [7]

2.1.2 Mixed-Mode Critical Energy Release Rate

Fracture often occurs under a combination of modes, thus an equivalent critical value is needed to include the contributions from each mode. The total energy release rate is calculated as the sum of the energy released from each mode, $\mathcal{G} = \mathcal{G}_I + \mathcal{G}_{II} + \mathcal{G}_{III}$, but this is not the case with the critical values. A universally accepted method for determining a mixed mode $\mathcal{G}_c$ does not exist, thus an appropriate method must be selected on a case-by-case basis. The following mixed-mode $\mathcal{G}_c$ calculations are available in Abaqus 6-13.3.

In 1984, Whitcomb [14] used the superposition of stresses to develop a power law relationship describing the mixed mode delamination of composites during buckling assuming that Mode I and II are primarily responsible for delamination. One of the results from his report was Eq. (2.7) where α and β are empirically determined constants.

$$\mathcal{G}_c = \left(\frac{\mathcal{G}_I}{\mathcal{G}_{Ic}} \right)^{\alpha} + \left(\frac{\mathcal{G}_{II}}{\mathcal{G}_{IIc}} \right)^{\beta} \tag{2.7}$$

While testing glass/epoxy composites in 1996, Benzeggagh and Kaine [15] observed that $\mathcal{G}_c$ increased as fracture transitioned from pure Mode I to pure Mode II. They developed the mixed Mode I-II, semi-empirical criterion in Eq. (2.8) as a combination of the critical values based on the fraction of $\mathcal{G}_{II}$ present in the analysis. The exponent, η, is fit to experimental data.

$$\mathcal{G}_c = \mathcal{G}_{Ic} + (\mathcal{G}_{IIc} - \mathcal{G}_{Ic}) \left(\frac{\mathcal{G}_{II}}{\mathcal{G}_I + \mathcal{G}_{II}} \right)^{\eta} \tag{2.8}$$

This was extended to three dimensions in 2006 by Reeder [16] at NASA. Eq. (2.9) combines all three fracture modes and η_1 is an empirical constant. Reeder also concluded that Eq. (2.9) has no physical significance in two dimensions and thus should not be used in those cases.

$$\mathcal{G}_c = \mathcal{G}_{Ic} + (\mathcal{G}_{IIc} - \mathcal{G}_{Ic}) \left(\frac{\mathcal{G}_{II} + \mathcal{G}_{III}}{\mathcal{G}_I + \mathcal{G}_{II} + \mathcal{G}_{III}} \right)^{\eta_1} + (\mathcal{G}_{IIIc} - \mathcal{G}_{IIc}) \left(\frac{\mathcal{G}_{III}}{\mathcal{G}_{II} + \mathcal{G}_{III}} \right) \left(\frac{\mathcal{G}_{II} + \mathcal{G}_{III}}{\mathcal{G}_I + \mathcal{G}_{II} + \mathcal{G}_{III}} \right)^{\eta_1} \quad (2.9)$$

2.1.3 Fatigue Crack Growth Models

Besides static loading, crack growth can occur when a subcritical load is repetitively applied. In 1961 Paris, Gomez, and Anderson hypothesized that the fatigue crack growth rate, da/dN, was related to the stress intensity range, $\Delta K = K_{max} - K_{min}$ [17]. On a log-log plot, it is now known that da/dN has a sigmoidal relation with ΔK as seen Fig. 2.2 and any textbook on fracture mechanics [8, 11].

The crack growth curve can be divided into three regions of generalized behavior. In the threshold region, crack growth is slow as ΔK asymptotically approaches the threshold value, ΔK_{th}, where crack growth may not occur. The slope of the crack growth curve is approximately linear in the intermediate, or Paris, region. And the unstable region is characterized by rapid, unstable crack growth where K_{max} asymptotically approaches K_c and fracture is imminent. Numerous crack growth models have been developed to describe the relationship between da/dN and ΔK within these regions. A select few are presented in this thesis.

In 1963, Paris and Erdogan [18] approximated the intermediate crack growth region with a power law relationship known today as the Paris equation, Eq. (2.10), where C and m are empirically determined material constants. One shortcoming of this relationship is that it applies to a single stress ratio, $R = \sigma_{min}/\sigma_{max}$, where stresses are defined far from the crack tip.

$$\frac{da}{dN} = C\,(\Delta K)^m \quad (2.10)$$

Walker [19] reported on the effects of the R-ratio on crack propagation for aluminum alloys in 1970, concluding that increasing R resulted in an increased growth rate. Eq. (2.11) was developed to address the effect of R-ratio on crack growth rate. Constants C_0, m, and γ are empirically determined from experimental data.

$$\frac{da}{dN} = \frac{C_0}{(1-R)^{m(1-\gamma)}}\,(\Delta K)^m \quad (2.11)$$

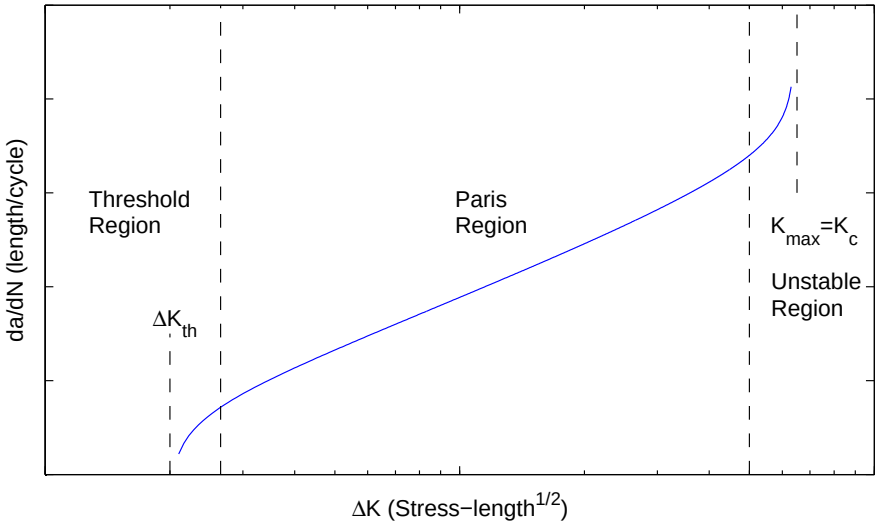

Fig. 2.2: Typical crack growth curve on log-log axes.

Paris and Walker equations work well in the Paris region of crack growth, but do not address the asymptotic behavior in the unstable region. To address this behavior, Forman, Kearney, and Engle proposed a model in 1967 [20], known as the Forman equation, Eq. (2.12). As K_{max} approaches K_c, the denominator approaches zero and da/dN tends toward infinity. Constants C_2 and m_2 are empirically determined through experimental testing.

$$\frac{da}{dN} = \frac{C_2 \left(\Delta K\right)^{m_2}}{\left(1-R\right) K_c - \Delta K} = \frac{C_2 \left(\Delta K\right)^{m_2}}{\left(1-R\right)\left(K_c - K_{max}\right)} \tag{2.12}$$

To incorporate the curvature in the threshold region, Broek and Rice [21] developed Eq. (2.13) for rail steels, as presented by Farris, Keer, and Steele [22]. The bracketed term adjusts the growth towards zero as ΔK approaches K_{th}, and the denominator forces the equation to infinity as K_{max} approaches K_c. Again, C_3 and m_3 are empirically determined constants.

$$\frac{da}{dN} = C_3 \left[\left(\Delta K\right)^2 - K_{th}^2\right] \frac{\left(\Delta K\right)^{m_3 - 1}}{K_c - K_{max}} \tag{2.13}$$

Eqs. (2.12) and (2.13) require data over a wide range of ΔK's. Since this is sometimes difficult to obtain, Dowling [8] recommends using either Eq. (2.10) or (2.11) for growth in the Paris region and bounding it by ΔK_{th} and $K_{max} = K_c$, or in terms of the stress intensity range, $\Delta K_c = K_c \left(1 - R\right)$.

The previously presented equations do not address the many factors that influence crack growth including load frequency, load sequencing, temperature, and environmental factors just to name a few. The following examples are given to show the narrow window in which Eqs. (2.10) through (2.13) apply. Hartman and Schijve [23] showed that an increase in

loading frequency produced a decrease in growth rates in aluminum alloys. Yokobori and Kiyoshi [24] followed this research by proposing the addition of a frequency term to Eq. (2.10) to address these affects. Hartman and Schijve's report [23] also found significant differences in the growth rates depending on the environments in which the tests were performed. Their report focused on wet and dry oxygen, air, and argon environments. In 1973, Yokobori and Aizawa [25] found that increasing the temperature of aluminum alloys increased the crack growth rate.

Finally, sequencing of the loads can play a large role in crack growth rates. An example is when overloading occurs prior to normal service loads. Overloading the crack can lead to large plastic deformation at the crack tip. The plasticly deformed region is surrounded by undeformed material that, upon removal of the applied overload, elasticly returns to its original configuration. The undeformed region places a compressive stress on the deformed region around the crack tip. The compressive stress around the crack tip results in crack growth retardation [8].

2.1.4 Onset of Crack Growth

Abaqus 6-13.3 requires a form of Eq. (2.14) to define the fatigue crack growth criterion. This section briefly address it's applicability to the current analysis.

Murri, Salpekar, and O'Brien [26] proposed a method to determine the maximum $\mathcal{G}_I$ that can be applied for the onset of delamination of tapered, unidirectional laminate. Eq. (2.14) proposes that the $\mathcal{G}_{Imax}$ required for the onset of delamination is a function of the cycles, N, and constants c and d. Analyses using this onset model are covered by Murri, O'Brien, and Rousseau [27] and a mixed-mode analysis is presented by Giannis [28].

According to ASTM Standard D6115-97 [29], this method is only known to apply to unidirectional composites. Section 4.5.1 of this thesis discusses setting the constants c and d to zero to remove Eq. (2.14) from the analysis.

$$\mathcal{G}_{Imax} = cN^d \tag{2.14}$$

2.1.5 Crack Extension Direction

Many models are available for predicting the direction of crack extension based on either stress, strain, energy, or any combination of these. Appropriate models should be selected based on material and loading conditions. The three models presented here were selected since they are popular in the literature and are available in Abaqus 6-13.3 without the need for a user defined subroutine.

Erdogan and Sih [30] proposed the Maximum Tangential Stress (MTS) which states that crack extension will occur radially from the crack tip and perpendicular to the maximum applied tensile load. These two criterion are met when the tangential stress, σ_θ, is maxi-

mized and the shear stress, $\tau_{r\theta}$, is zero. Erdogan and Sih developed parametric equations to determine the propagation angle, θ, in K_I-K_{II} space. Setting $\tau_{r\theta} = 0$ and $\partial\sigma_\theta/\partial\theta = 0$, the parametric equations were solved in the Abaqus Theory Manual [31] and are provided in Eq. (2.15). Validation experiments were performed on Plexiglas sheets and Erdogan and Sih reported good agreement with the parametric equations..

$$\theta = \cos^{-1}\left(\frac{3K_{II}^2 + \sqrt{K_I^4 + 8K_I^2K_{II}^2}}{K_I^2 + 9K_{II}^2}\right) \tag{2.15}$$

In 1974, Hussain, Pu, and Underwood [32] developed the Maximum Energy Release Rate (MERR) criterion which was based on Griffith and Irwin's work. MERR states that crack extension will occur at an angle, θ, that maximizes $\mathcal{G}$ in Eq. (2.6) for a mixed Mode I-II condition. Equations are provided in [32] that compute $K_I = K_I(\theta)$, $K_{II} = K_{II}(\theta)$, and assume $K_{III} = 0$. Hussain et al. state that the MERR model compared well to experimental data collected for crack growth in steel foil.

Cotterell and Rice [33] suggested in 1980 that crack extension occurs in a direction where $K_{II} = 0$ for isotropic, homogeneous materials. Cotterell and Rice also proposed that the MTS and MERR models' solutions meet the $K_{II} = 0$ criterion once the crack has extended. Curved and kinked crack models were analyzed in their article, but no experimental verification was reported.

Each crack extension direction criterion will give slightly different results. Analysis of crack propagation in Abaqus should be run with each criterion described in this section and compared to experimental data for selection of the most appropriate one.

2.2 Methods of Fracture Analysis

A fracture analysis typically starts with an initial crack size, or crack initiation criteria based on stress or strain, and propagates the crack until a critical value is reached such as K_c or $\mathcal{G}_c$. Growth rates are calculated using a fatigue crack growth model such as the Walker equation, Eq. (2.11). Several methods have been developed in the literature and a select few are presented in the following sections.

2.2.1 Analytical Methods

Integrating a fatigue crack growth law such as Eq. (2.10) is the most basic approach explored in this thesis for solving for the fatigue fracture life of a specimen. The integral approach assumes self-similar Mode I crack propagation and is used a validation tool in this thesis.

Programs have been developed to aid in these calculations such as AFGROW, created by The Air Force Research Laboratory and LexTech, Inc. [34], and NASGRO, created by NASA Johnson Space Center and Southwest Research Institute® [35]. Programs like these access

a database of equations for SIFs, and apply a load spectrum and fatigue crack growth law to analyze crack growth and estimate fatigue fracture life. AFGROW is used in this thesis for the crack growth analysis. While these programs work well for the catalog of design situations, a finite element analysis will likely be required for more complex geometry, boundary conditions, and loading conditions.

2.2.2 Finite Element Methods

Finite Element Methods (FEMs) are used to provide approximate solutions to partial differential equations in the form of either boundary-value or initial-value problems. The FEM was first developed as early as 1851 and gained wide-spread use with the advent of the digital computer in the 1960s [36]. The FEM has been applied to problems in stress and strain analysis, wave propagation, heat transfer, electrical and magnetic fields, and fluid flow. Currently, finite element analysis software is readily available in commercial codes like Abaqus and ANSYS. These codes are continually updated with the most recent advances in the literature such as the eXtended Finite Element Method for fracture mechanics.

Early Applications of the FEM. Some of the first applications of FEM to fracture mechanics were reported in 1969 by Watwood [37] for a center cracked plate, and Anderson, Garron, and Ruggles [38] for fracture of a solid propellant rocket motor cartridge. These early applications of the FEM to problems in fracture mechanics were followed in 1970 by Chan, Tuba, and Wilson [39] who analyzed compact and rotating test specimens. All authors noted that the stress singularity at the crack tip was unattainable with linear elements, and thus their analyses quickly led to Griffith's energy release rate or to the contour integral for J and K. All reports showed good agreement between energy release rates, contour integrals, and analytical methods.

Specialized Crack Tip Elements. In 1970, in an effort to address the issues with modeling the stress singularity, Byskov [40] developed a triangular crack tip element containing necessary shape functions to accurately model the singularity. This approach was expanded to circular elements by Wilson [41] in 1973 and rectangular elements in 1974 by Hardy [42]. The circular element has had many versions including a mixed-mode element by Holston [43] in 1976 and a bending specific version developed by Jiang and Cheung [44] in 1995.

Finite Elements with Quarter Points. To remove the need for specialized elements, Henshell and Shaw [45] published a paper in 1975 which concluded that moving the mid-side nodes of quadratic elements adjacent to the crack tip to their quarter point location adequately models the $1/\sqrt{r}$ stress singularity. Barsoum [46] simultaneously published similar findings without knowledge of Henshell and Shaw's work. A schematic of this concept is shown in Fig. 2.3. The following year, Barsoum [47] showed that all nodes of the elements at the crack tip should be collapsed to a single node for linear elastic analyses, and that this

method could be extended to model the $1/r$ singularity for a perfectly plastic analysis when the collapsed nodes remain independent.

To analyze the fatigue crack growth using Finite Elements with Quarter Points (FEQP), finite element models must be built at incremental crack lengths since explicit crack propagation is not possible. Typically a concentric meshing scheme is used around the crack, as seen in Fig. 2.3. Højfeldt and Østervig [48] illustrated this procedure in 1986 for cracks in shoulder fillets of a shaft. In general this process proceeds as follows:

1. Build and mesh a finite element model including the crack in the geometry.
2. Apply quasi-static load from minimum to maximum value.
3. Determine ΔK's or $\Delta \mathcal{G}$'s.
4. Determine crack extension direction.
5. Calculate incremental crack growth length.
6. Determine the cycles required to grow the crack this incremental length based on a fatigue crack growth model.
7. Add cycles to the previous cycle count.
8. Rebuild the model and mesh with new crack geometry.
9. Repeat steps 2 through 8 until critical values are reached.

Recent publications show that this process has remained basically unchanged, including Alegre and Cuesta's [49] 2010 paper where crack propagation paths where compared for various initial crack angles. Both [48] and [49] observed good comparison between experimental and finite element results.

Virtual Crack Closure Technique. Rybicki and Kanninen [50] developed the modified crack closure integral in 1977 which was based on Irwin's 1957 assertion that the energy needed to grow a crack is equal to the work required to close a crack of the same length. Eq. (2.16) was developed in the paper for Mode I and Mode II energy release rates for a unit thickness. This was later extended to Mode III by Shivakumar, Tan, and Newman [51]. Referring to Fig. 2.4 to define variables; $F_{\_,cd}$ are force components required to keep nodes c and d from separating, $u_{\_,ab}$ are the displacement components between nodes a and b, Δa is the incremental crack length over which $\mathcal{G}$ is being calculated, and l is the element length in front of the crack tip. These equations apply only to uniform meshes where $\Delta a = l$; equations are provided in [50] for $\Delta a \neq l$.

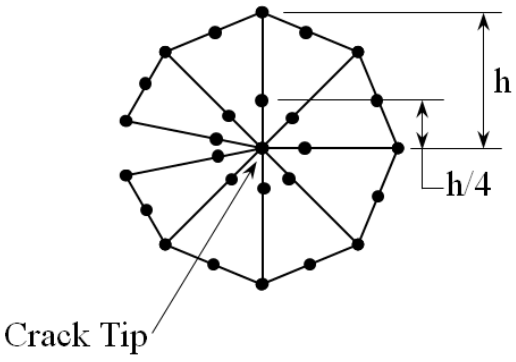

Fig. 2.3: Schematic mesh of elements around a crack with mid-side nodes moved to the quarter point location.

$$\mathcal{G}_I = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} F_{y,cd} u_{y,ab} \tag{2.16a}$$

$$\mathcal{G}_{II} = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} F_{x,cd} u_{x,ab} \tag{2.16b}$$

$$\mathcal{G}_{III} = \lim_{\Delta a \to 0} \frac{1}{2\Delta a} F_{z,cd} u_{z,ab} \tag{2.16c}$$

Today, Rybicki and Kanninen's method is known as the Virtual Crack Closure Technique (VCCT) and the technique has been used in a myriad of publications for static and cyclic analyses and three dimensional configurations that may be too difficult to mesh using FEQP. In general, the process used in the literature to model VCCT cyclic crack growth is similar to that for FEQP as shown in Mabson et al. [52] and Hosseini-Toudeshky, Ghaffari, and Mohammadi [53]. Mabson et al. analyzed the fatigue fracture life of double cantilevered beam specimens while Hosseini-Toudeshky et al. looked at the crack growth path in curved aluminum panels with stiffeners and composite repair patches. Both studies observed good agreement between experimental data and VCCT results.

eXtended Finite Element Method. The finite element methods discussed so far require that the crack be incorporated into the geometry so that elements align with the crack boundary. The eXtended Finite Element Method (XFEM) was developed in 1999 by Belytschko and Black [3] as a method where cracks could be defined arbitrarily within the mesh. Their method was based on the partition of unity presented by Melenk and Babuška [54] in 1996 which broadly states that the sum of all shape functions must be one, thus allowing additional terms to be added to the shape functions with *a priori* knowledge of the solution.

Belytschko and Black suggested using enriched shape functions shown in Eq. (2.17) for the displacement field, $\boldsymbol{u}$. $N(\boldsymbol{x})$ are standard shape functions, $\boldsymbol{b}$ and $\boldsymbol{c}$ are nodal coefficient, (r, θ)

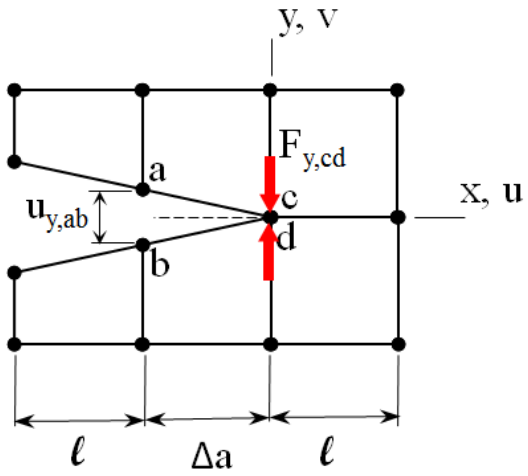

Fig. 2.4: Schematic finite element representation of the VCCT method.

is a polar coordinate system originating at the crack tip, and S_1 is a set containing all nodes in the FE model. Crack tip enrichment terms, F, are only activated in the element containing the crack tip, represented by node set S_3. Moës, Dolbow, and Belytschko [55] added the jump function enrichment term, $H(\boldsymbol{x})$, which is used to represent the discontinuous displacement field in elements that have been completely bisected by a crack. S_2 is the node set where the jump function is active. Fig. 2.5 illustrates where crack tip and jump function enrichment terms would be used. SIFs calculated with contour integrals around the crack tip in [3, 55] were in excellent agreement with analytical values.

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i \in S_1} N_i(\boldsymbol{x}) \boldsymbol{u}_i + \sum_{j \in S_2} N_j(\boldsymbol{x}) \boldsymbol{b}_j H(\boldsymbol{x}) + \sum_{k \in S_3} N_k(\boldsymbol{x}) \left[\sum_{n=1}^{4} \boldsymbol{c}_{nk} F_n(r, \theta) \right] \tag{2.17a}$$

$$\{F_i(r,\theta)\}_{i=1}^{4} \equiv \left\{ \sqrt{r}\cos\frac{\theta}{2},\ \sqrt{r}\sin\frac{\theta}{2},\ \sqrt{r}\sin\frac{\theta}{2}\sin\theta,\ \sqrt{r}\cos\frac{\theta}{2}\sin\theta \right\} \tag{2.17b}$$

$$H(\boldsymbol{x}) = \begin{cases} 1 & \text{for } y > 0 \\ -1 & \text{for } y < 0 \end{cases} \tag{2.17c}$$

A Level Set Method (LSM) was developed by Osher and Sethian [56] in 1988 to locate moving boundaries such as grain boundaries and inclusions in material models. Stolarska, Chopp, Mëos, and Belytschko [57] incorporated the LSM concept into XFEM in 2001. The LSM implementation in XFEM uses two nearly orthogonal signed distance functions, $\psi(\boldsymbol{x}, t)$ and $\phi(\boldsymbol{x}, t)$, to locate the crack front. $\psi(\boldsymbol{x}, t) = 0$ locates the crack face and $\phi_i(\boldsymbol{x}, t) = 0$ defines the i^{th} crack tip. Nodal values for ψ and ϕ are incorporated into the mesh with the standard shape functions with Eq. (2.18). Level set regions used to define a crack are illustrated in Fig. 2.6. As the crack propagates through the mesh, values of ψ are stored in the output file

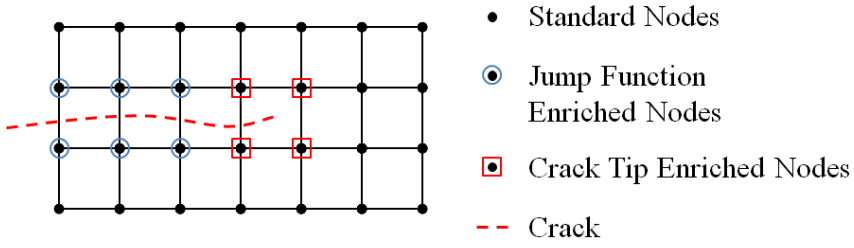

Fig. 2.5: XFEM nodes with enrichment and jump functions.

since the previously formed crack face is assumed not to change. Stolarska et al. modified Eq. (2.17c) to include the LSM values resulting in Eq. (2.19).

$$\phi_i(\boldsymbol{x},t) = \sum_{j\in S_2} \phi_{ij}(t)\, N_j(\boldsymbol{x}) \tag{2.18a}$$

$$\psi(\boldsymbol{x},t) = \sum_{j\in S_2} \psi_j(t)\, N_j(\boldsymbol{x}) \tag{2.18b}$$

$$H(\boldsymbol{x}) = H(\psi(\boldsymbol{x},t)) = \begin{cases} 1 & \text{for } \psi(\boldsymbol{x},t) > 0 \\ -1 & \text{for } \psi(\boldsymbol{x},t) < 0 \end{cases} \tag{2.19}$$

XFEM has become extremely popular since its inception in 1999. Most crack propagation analyses follow a similar processes used by FEQP and VCCT; however, remeshing is typically not an issue. The basic procedure is as follows:

1. Build and mesh a finite element model.
2. Define the crack location by nodal level set values.

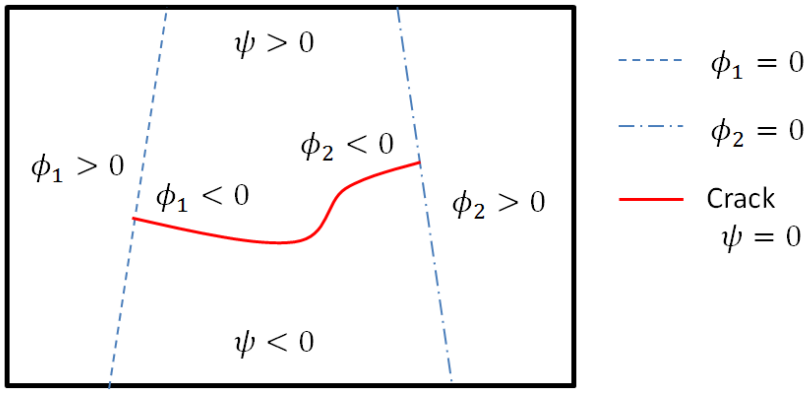

Fig. 2.6: Level Set Method used for XFEM.

3. Apply quasi-static load from minimum to maximum value.
4. Determine ΔK's or $\Delta \mathcal{G}$'s.
5. Determine crack extension direction.
6. Calculate incremental crack growth length.
7. Determine the cycles required to grow the crack this incremental length based on a fatigue crack growth model.
8. Add cycles to the previous cycle count.
9. Redefine crack location by adjusting nodal level set values without remeshing.
10. Repeat steps 3 through 9 until critical values are reached.

XFEM was incorporated into an Abaqus user defined subroutine by Shi, Chopp, Lua, Sukumar, and Belytschko [58] in 2010 to analyze cyclic crack growth in standard and modified compact specimens, as well as a complex helicopter part. Shi et al.'s subroutine used a *modified* VCCT method for calculating the energy release rates at the crack tip. A significant amount of published XFEM fatigue crack growth articles deal with complex geometry that would be time consuming or impossible to mesh using FEQP or VCCT methods. These include Singh, Mishra, Bhattacharya, and Pati's [59] analysis of the effects on crack growth by randomly adding inclusions, holes, and minor cracks to two dimensional center cracked and edge cracked plates. Or Pathak, Singh, and Singh's [60] analysis of embedded and surface elliptical cracks, lens cracks, and arbitrary shaped cracks of various aspect ratios and inclinations. Each of these articles noted good agreement with analytical and experimental values when available.

eXtended Finite Element Method with Phantom Nodes. Song, Areias, and Belytschko [4] developed an adaptation of XFEM to model arbitrary crack locations within a mesh using phantom nodes in 2006. Drawing conclusions from previous work by Hansbo and Hansbo [61] and Belytschko and Black [3], the eXtended Finite Element Method with Phantom Nodes (XFEM-PN) represents the discontinuous displacement field with superposed elements each containing phantom nodes and phantom degrees of freedom. Prior to fracture, superposed elements move together, acting as a single element. Once bisected by a crack, real nodes and phantom nodes move independently of each other, schematically shown in Fig. 2.7. Eq. (2.20) represents the displacement field, $\boldsymbol{u}$, as the sum of the displacements from elements 1 and 2. $S_1$and S_2 are the sets of nodes corresponding to each element and H is the Heaviside step function. XFEM-PN has the advantage over standard XFEM in that no crack tip enrichment terms are needed to model the stress singularity, thus it can be integrated directly into most FE codes. Abaqus 6-13.3 combines phantom nodes and LSM for crack propagation analyses [5].

$$\boldsymbol{u}\left(\boldsymbol{x},t\right)=\sum_{i\in S_1}\boldsymbol{u}_i^{(1)}\left(t\right)N_i(\boldsymbol{x})H\left(-f\left(\boldsymbol{x}\right)\right)+\sum_{i\in S_2}\boldsymbol{u}_i^{(2)}\left(t\right)N_i(\boldsymbol{x})H\left(f\left(\boldsymbol{x}\right)\right) \tag{2.20a}$$

$$H(x)=\begin{cases}1 & \text{for } x>0\\ 0 & \text{for } x\leq 0\end{cases} \tag{2.20b}$$

Steps followed to model fatigue crack growth with XFEM-PN would be the same as those with the standard XFEM. Published literature on XFEM-PN for fatigue crack propagation was not as abundant compared to XFEM when this thesis was written. Kucharczyk, Sharaf, and Münstermann [62] successfully modeled the reduction in load capacity due to crack growth through steel grains; however this paper used traction separation laws with cohesive elements, and not the fatigue crack growth models such as the Paris or Walker equations.

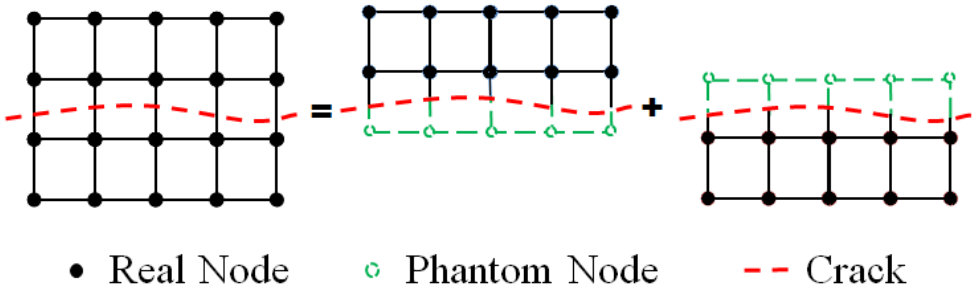

Fig. 2.7: Schematic mesh with phantom nodes.

2.3 Stochastic Analyses in Fracture Mechanics

Fatigue crack growth is inherently stochastic and difficult to predict. Virkler, Hillberry, and Goel [63] published a paper in 1979 which analyzed the statistical distribution of crack propagation in 2024-T3 aluminum center cracked plates. Tests were conducted on sixty-eight identical specimens with constant amplitude loading. Experimental results showed large amounts of variability in crack growth rates not only between samples but also within each sample. The variability within the each sample was attributed to the non-homogeneity of the material at the microscopic level, thus growth rate Eqs. (2.10)-(2.13) only apply in an average sense macroscopically.

One way to model a stochastic process is to use a Monte Carlo method where constants are randomly selected from a Probability Density Function (PDF). A normal PDF is based on the constant's mean value, μ, and variance, σ^2. The analysis is performed using samples from the distribution of the random variables. Repeated analyses are performed continuing to draw samples from the distributions of the random variables until a mean outcome from the analysis is realized. The fundamental principles of the Monte Carlo method are the law of large numbers, and the central limit theorem [64]. The law of large numbers states that the mean value of a sample approaches the mean value of the population as the number of samples approach infinity. The central limit theorem is used to estimate the uncertainty in the mean value, thus relating the standard deviation of the sample, s, to the standard deviation of the population, σ.

Annis [65] performed a Monte Carlo analysis and compared results to experimental data presented in Virkler's report [63]. If random numbers were selected for C and m in the Paris equation, the Monte Carlo simulation resulted in a variation more than seven times greater than the variation in the experimental data. To fix the error, he proposed that C cannot be picked independently of m thus a bivariate normal distribution must be used. Multivariate normal distributions could be used if more than two constants are correlated. Applying this concept; Annis reduced the error to approximately 1%.

Chapter 3

Crack Growth Rates for 7075-T6 Aluminum Alloy

In 1969, C. Michael Hudson published a report for NASA on the effects of the R-ratio on fatigue crack growth rates for two types of aluminum alloys, 7075-T6 and 2024-T3 [6]. The data collected for the 7075-T6 aluminum alloy is the focus of this thesis. The following sections discuss the experimental data that was collected and the tests performed, three fatigue crack growth models fit to the collected data, and a selection of a model to be used in the remainder of the thesis.

3.1 Experimental Data

Hudson [6] performed three material characterization tests on the 7075-T6 aluminum alloy specimen to determine the properties needed for a fatigue crack growth analysis: (1) tensile test for elastic properties, (2) fatigue crack growth test for a crack growth rate model, and (3) residual static strength test for the critical stress intensity factor, K_c.

The aluminum alloy specimen measures 12 in. (305 mm) wide by 35 in. (889 mm) tall and 0.09 in. (2.29 mm) thick. It was loaded by five pins on both the top and bottom as seen in Fig. 3.1. An initial stress raiser 0.1 in. (2.54 mm) long was made in the center of the plate with an electrical discharge machine (EDM). A detailed drawing can be found in Fig. A.1 of Appendix A.

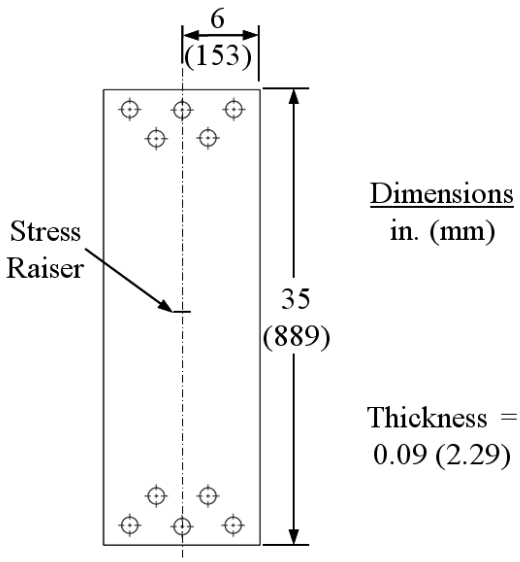

Fig. 3.1: Center Cracked Plate Specimen

Elastic properties were determined from a standard tensile test and are listed in Table A.1. Experimental data from the tensile tests was not provided in Hudson's report.

Fatigue crack growth tests were run on multiple axial-load fatigue-testing machines which applied monotonic oscillatory loads with a mean stress, σ_m, and alternating stress, σ_a, levels listed in Table A.2 of Appendix A. As the crack propagated through the specimens, the cycle count was recorded at incremental crack lengths. Two tests were performed at each load level and the reported results are the average results from both tests. It should be noted that the incremental crack lengths provided in Hudson's paper are coarse compared to current ASTM Standard E647. Section 8.8 of ASTM Standard E647 [66] suggests a $\Delta a \approx 0.05\,\text{in.}$ while Hudson's measurements ranged from $\Delta a = 0.1$ to $0.2\,\text{in.}$ Only tests with an $R \geq 0$ were analyzed in this thesis since Hudson found that $R < 0$ had the same growth rate as specimens loaded with $R = 0$.

Approximately one-third of the tests were stopped prior to failure in the fatigue crack growth test, and these specimens were used in the residual static strength test. It was not specified in Hudson's report which specimens were stopped prior to failure in fatigue crack growth test. For the static strength test, an increasing axial load was applied to the specimens until rapid unstable crack growth was visually observed with a camera. When failure was observed, the final load, P_f, and final crack length, a_f, were recorded. Resulting P_f and a_f values listed in Table A.3 were used to calculate K_c for the specimen. Visually determining when failure occurs does not conform to current ASTM Standard E399. Section 9 of ASTM Standard E399 [67] suggests the use of a load versus displacement, or P-v, curve to determine when failure occured. The visual method may result in an overestimated final load and final crack length, which would lead to an overestimated K_c. The effects of an overestimated K_c will be mentioned again in Section 4.2 when analyzing the estimated critical crack lengths, a_c, from analytical equations.

3.2 Fatigue Crack Growth Models

Many crack growth models have been developed to describe the relationship between da/dN and ΔK within the three regions of generalized crack growth behavior seen in Fig. 2.2. Three models explored in this thesis are the Paris, Walker, and Forman crack growth models.

3.2.1 Paris Crack Growth Model

The first crack growth model investigated in this thesis is the Paris equation, Eq. (3.1). Crack length and cycles provided in Table A.2 were used to determine the crack growth rate, da/dN. A three-point incremental polynomial method, as described in Appendix XI of ASTM Standard E647 [66], was selected to calculate da/dN since it gave smoother results than the secant method. Using more than three points to define the polynomial removed too much data without any noticeable gains. The stress intensity range, ΔK, was calculated with Eq. (3.2) where $\Delta\sigma$ is the stress range calculated as $\Delta\sigma = 2\sigma_a$ and a is the half-crack length. For the dimensionless geometry factor, F, Eq. (3.3) from Tada [2] was used since this form is accurate to within 1% of experimental data for any $\alpha = a/b$ and $h/b \geq 1.5$ where h is the half-height and b is the half-width of the specimen. For this specimen, the half-height was measured arbitrarily 1.5 in. below the centerline of loading holes as it was assumed that the load would be uniformly distributed at this point. Measuring down to the horizontal centerline of the plate resulted in an $h/b = 2.0$.

$$\frac{da}{dN} = C\,(\Delta K)^m \tag{3.1}$$

$$\Delta K = F(\alpha)\Delta\sigma\sqrt{\pi a} \tag{3.2}$$

$$F(\alpha) = \frac{1 - 0.5\alpha + 0.326\alpha^2}{\sqrt{1-\alpha}} \tag{3.3}$$

Constants C and m were found using a least squares linear regression in MATLAB after transforming the data to log-log space. Constants for models fit to each R-ratio are listed in Table 3.1 and a plot is provided in Fig. 3.2 showing the data points and fitted models.

The Paris equation fits well for each R-ratio since all Root Mean Square Errors (RMSE) are relatively small. R-ratios of 0.2 and 0.33 had the smallest error since all data appeared be recorded within the intermediate region of crack growth where growth rates are linear in log-log space. Models with $R = 0$, 0.5, and 0.8 have a RMSE > 0.1 which is a result of the increased growth rates near the end of each data set where K_{max} approaches K_c. The largest RMSE was with $R = 0.8$ which may also be attributed to the smaller data set. Error in the model with $R = 0.67$ is attributed to the slower growth rates where $da/dN < 10^{-6}$ in./cycle. The slower growth rates could be a result of the variability in growth rates or growth rates could be approaching the threshold region of crack growth, but no other data points were available in this region for comparison. The Walker equation was used next, combining all Paris models into one, using the entire data set, and addressing the R-ratio effect.

Table 3.1: Paris Equation Coefficients

R-ratio	C $\frac{\text{in./cycle}}{\left(\text{ksi }\sqrt{\text{in.}}\right)^m}$	C $\frac{\text{mm/cycle}}{\left(\text{MPa }\sqrt{\text{mm}}\right)^m}$	m	RMSE
0	1.56×10^{-9}	9.83×10^{-14}	3.64	0.114
0.2	4.28×10^{-10}	4.72×10^{-15}	4.13	0.083
0.33	3.79×10^{-9}	2.86×10^{-13}	3.59	0.095
0.5	5.47×10^{-9}	3.40×10^{-13}	3.64	0.128
0.67	3.36×10^{-9}	3.55×10^{-14}	4.14	0.108
0.8	2.50×10^{-9}	3.86×10^{-15}	4.68	0.184

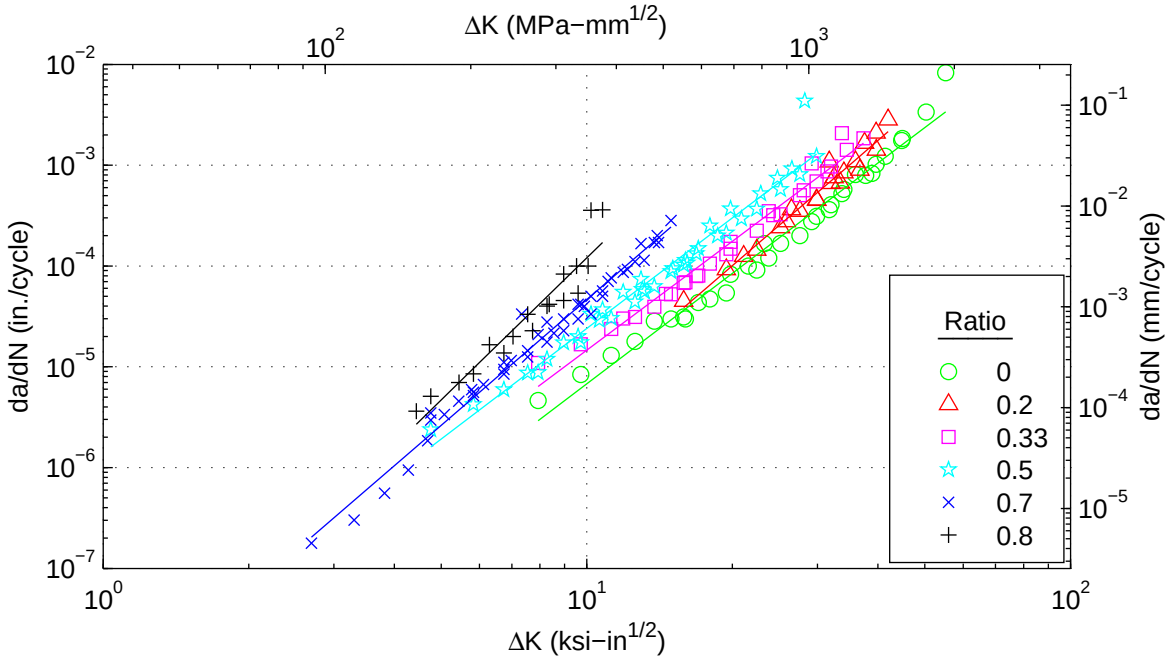

Fig. 3.2: Paris equation fit to Hudson's experimental data.

3.2.2 Walker Crack Growth Model

The Walker equation, Eq. (3.4), collapses data from different R-ratios to a single line when plotting da/dN versus $\overline{\Delta K}$ in Eq. (3.5). $\overline{\Delta K}$ is defined as the equivalent stress intensity range at $R = 0$. The same da/dN and ΔK values calculated previously were used for the Walker equation. A least squares multiple linear regression was used in MATLAB to fit the model after transforming the data to log-log space. The coefficients and a plot of the results are shown in Table 3.2 and Fig. 3.3, respectively.

$$\frac{da}{dN} = \frac{C_0}{(1-R)^{m(1-\gamma)}} (\Delta K)^m \tag{3.4}$$

$$\overline{\Delta K} = \frac{\Delta K}{(1-R)^{1-\gamma}} \tag{3.5}$$

The Walker model fits well to all data with an RMSE value comparable to the Paris models. Error was attributed to the increased growth rates in experiments with $R = 0$, 0.5, and 0.8 as they approached fracture and decreased growth rates in the $R = 0.67$ experiment. The Forman equation was used next to address the increased growth rates in the unstable region.

Table 3.2: Walker Equation Coefficients

C_0		m	γ	RMSE
$\frac{\text{in./cycle}}{(\text{ksi}\sqrt{\text{in.}})^m}$	$\frac{\text{mm/cycle}}{(\text{MPa}\sqrt{\text{mm}})^m}$			
8.63×10^{-10}	2.66×10^{-14}	3.84	0.564	0.130

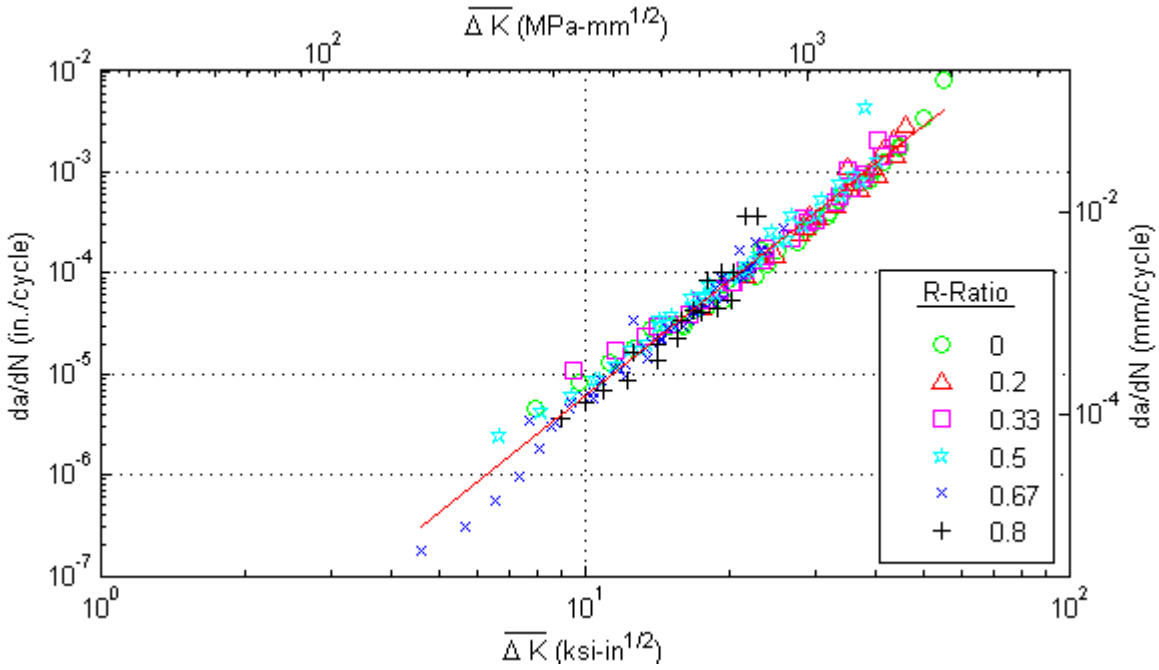

Fig. 3.3: Walker equation fit to Hudson's experimental data.

3.2.3 Forman Crack Growth Model

The Forman equation, Eq. (3.6), was used to model the asymptotic increases in the crack growth rate as K_{max} approached K_c. When Q in Eq. (3.7) was plotted against ΔK, data

from different R-ratios collapse to a single line in log-log space and the curvature in the unstable region was removed.

$$\frac{da}{dN} = \frac{C_2 \left(\Delta K\right)^{m_2}}{\left(1 - R\right) K_c - \Delta K} \tag{3.6}$$

$$Q = \frac{da}{dN} \left[\left(1 - R\right) K_c - \Delta K\right] \tag{3.7}$$

Data from the residual static strength test was used to determine the critical stress intensity factor for the material and CCP geometry. Eq. (2.1) was rewritten in the form of Eq. (3.8) where t is the plate thickness and all other variables were previously defined. Applying this equation to the data in Table A.3 and averaging the results gives $K_c = 72.0\,\text{ksi}\,\sqrt{in.}$ $(2500\ \text{MPa}\sqrt{mm})$.

$$K_c = F\left(\frac{a_f}{b}\right) \frac{P_f}{2bt} \sqrt{\pi a_f} \tag{3.8}$$

Similar to the Walker equation, this model addresses the R-ratio effects, thus only one model was needed to describe the entire data set. Following the same procedure as before, and using a least square linear regression on the transformed data, the coefficients in Table 3.3 and the plot Fig. 3.4 were developed.

Table 3.3: Forman Equation Coefficients

C_2		m_2	RMSE
$\frac{\text{in./cycle}}{\left(\text{ksi}\,\sqrt{\text{in.}}\right)^{m_2-1}}$	$\frac{\text{mm/cycle}}{\left(\text{MPa}\,\sqrt{\text{mm}}\right)^{m_2-1}}$		
3.39×10^{-7}	1.24×10^{-10}	3.14	0.119

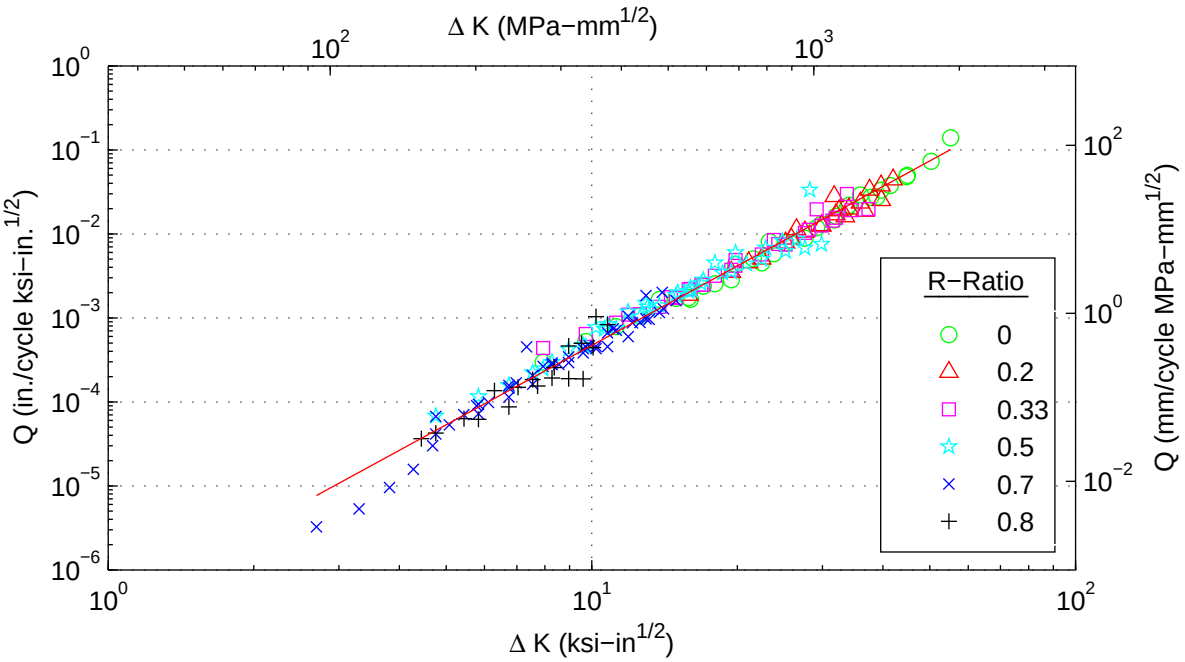

Fig. 3.4: Forman equation fit to Hudson's experimental data.

The Forman equation fits well to all data with an RMSE similar to the Walker and Paris models. Error was attributed to the decreased growth rates from the experiment with $R = 0.67$. Comparing Figs. 3.3 and 3.4, it appears that only a few data points where located in the unstable region due to the coarse crack measurements. It is recommended that additional data be collected in the unstable region to adequately fit this model.

3.3 Conclusion

The Paris equation is widely used in fracture mechanics and fits well to the data for each R-ratio. The six Paris equations were combined into one with the Walker equation which accounts for the stress ratio effects and uses the entire data set to fit the model. The Forman equation has the added advantage of modeling the asymptotic curvature near fracture. However, Hudson's experimental data only has a few data points in this region due to the coarse measurements. The Walker and Forman models had similar RMSE values, suggesting that the models equally represent data set. As will be discussed in Chapter 4, Abaqus 6-13.3 is not able to incorporate the Forman equation without the use of a user defined subroutine which is outside the scope of this thesis. Since no data appeared to be collected in the threshold region, the Broek and Rice's model could not be used. For these reasons, the Walker equation has been selected for the analysis in the remainder of this thesis.

Chapter 4

Estimating Fatigue Fracture Life

It has been well documented in the literature that the R-ratio [6, 19, 68], mean stress level [69, 70], and loading frequency [23, 24] can affect fatigue fracture life. To limit the affect of these fatigue fracture parameters, four loading conditions have been selected from Hudson's report. Each selection has the same mean stress, σ_m, while increasing the R-ratio. It was not possible to select a single loading frequency while maintaining σ_m. Selected loading conditions are listed in Table 4.1.

Table 4.1: Analyzed Loading Conditions

Loading	σ_m		σ_a		R	Frequency
Condition	(ksi)	(MPa)	(ksi)	(MPa)		(Hz)
1	15.0	103	15.0	103	0	0.5/13.7*
2	15.0	103	10.0	68.9	0.2	0.5
3	15.0	103	5.00	34.5	0.5	13.7
4	15.0	103	3.00	20.7	0.67	13.7

* Indicates multiple tests run a different frequencies.

The following sections explore five different methods for evaluating the fatigue fracture life of a Center Cracked Plate (CCP) model: (1) analytical methods based on published equations, (2) AFGROW, a program common in the aerospace industry used for crack analysis, (3) Finite Elements with Quarter Points (FEQP), an analysis procedure where the crack is manually grown which requires remeshing, (4) Virtual Crack Closure Technique (VCCT), where a crack propagates along a predefined path, and (5) the eXtended Finite Element Method with Phantom Nodes (XFEM-PN) which allows a crack to propagate along a solution dependent path [5]. First, the specimen model and general assumptions are established.

4.1 Center Cracked Plate Model

The tests described in Hudson's 1969 NASA report were performed on a thin CCP specimen which was assumed to be in a two-dimensional state of plane stress since $t \ll h$ and $t \ll b$, where t is the thickness, h is the half-height, and b is the half-width. The plate was loaded by five pins near the top and bottom edges. For simplification, the pins have been removed and an evenly distributed stress was assumed across the top and bottom edges of the plate. According to a study by Isda in 1971 [71], the effects of the boundary condition on the stress intensity factor are minimal when $h/b \geq 1.5$ for a center cracked plate. As shown in Fig. 4.1, the simplified model arbitrarily begins 1.5 inches below the centerline of the inner loading holes. This results in $h/b = 2.0$ which is within the acceptable range of values.

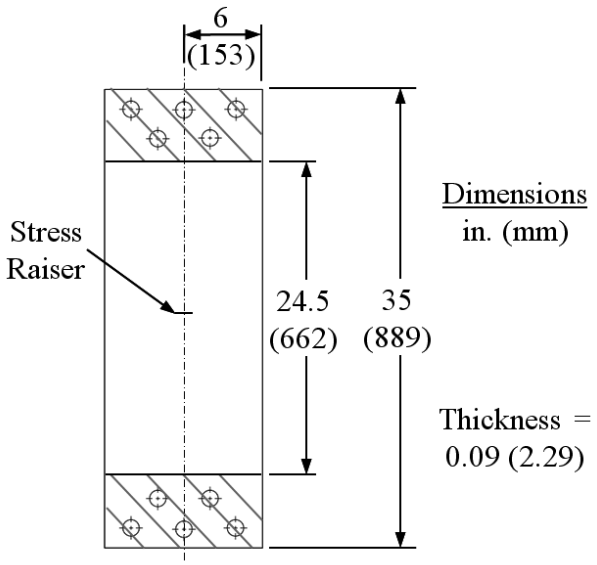

Fig. 4.1: Simplified Center Crack Plate Model

4.2 Analytical Method

The analytical method was selected as a baseline for the comparison to all other techniques. The concept for the analytical method was to integrate the Walker equation, Eq. (4.1), from an initial crack length, a_i, to a critical crack length at fracture, a_c, resulting in the estimated fatigue fracture life, N_f, of the specimen. The following sections briefly describe the process and present the results with comparisons to the experimental data.

$$\frac{da}{dN} = \frac{C_0}{(1-R)^{m(1-\gamma)}} (\Delta K)^m \tag{4.1}$$

4.2.1 Model Setup

The critical crack length, a_c, was calculated by rearranging Eq. (2.1) and using the critical stress intensity factor, K_c, resulting in Eq. (4.2). This equation requires an iterative solution since $F = F(a/b)$. This equation was solved in MATLAB using a system of non-linear equations and the function `fsolve()` with the default termination tolerance of 10^{-6}.

$$a_c = \frac{1}{\pi}\left(\frac{K_c}{F(a_c/b)\,\sigma_{max}}\right)^2 \tag{4.2}$$

Eq. (4.3) was substituted into Eq. (4.1) and rearranged for integration. The resulting Eq. (4.4) does not have a closed form solution in general, because of the dimensionless geometry factor, thus it was numerically integrated in MATLAB with the `integral()` function. The code used for this analysis can be found in Appendix F.

$$\overline{\Delta K} = \frac{\Delta K}{(1-R)^{1-\gamma}} \tag{4.3}$$

$$N_f = \int_{N_i}^{N_c} dN = \frac{1}{C_0}\int_{a_i}^{a_c} \frac{da}{\left(\overline{\Delta K(a)}\right)^m} \tag{4.4}$$

This method assumes self-similar crack growth, thus only the Mode I Stress Intensity Factor (SIF), K_I, is present. Applying the Maximum Tangential Stress (MTS) criterion, restated in Eq. (4.5), for the crack extension direction confirms this assumption since $\theta = \cos^{-1}(1) = 0$, with $K_{II} = 0$.

$$\theta = \cos^{-1}\left(\frac{3K_{II}^2 + \sqrt{K_I^4 + 8K_I^2K_{II}^2}}{K_I^2 + 9K_{II}^2}\right) \tag{4.5}$$

4.2.2 Results

Table 4.2 list the results for critical crack length and estimated fatigue fracture life. For a comparison, the last recorded crack length and fatigue cycles from Hudson's report are also listed. The analytical critical crack lengths are approximately twice as large as the final measured crack lengths, while the life estimates are within 10-20% of the experiments. One reason for the discrepancy in crack length is that K_c may have been overestimated based on the visual testing method used in Hudson's report and mentioned in Section 3.1. Also, the coarse crack measurements of 0.1-0.2 in. are not able capture the exact crack length when fracture occurred. In the unstable region, high growth rates only require a few cycles to propagate a crack over a significant length, thus N_f is not sensitive to the final crack length. This was confirmed by a sensitivity analysis presented in AFGROW's Handbook for Damage Tolerance Design [72]. Adding to the discrepancy of critical crack length, Hudson's report

did not specify whether specimens were fractured in the fatigue crack growth test or were removed prior to failure for the residual strength test.

Table 4.2: Results from the Analytical Fatigue Fracture Analysis

Loading Condition	R	Critical Crack Length, a_c, in. (mm)		Fatigue Fracture Life, N_f, cycles	
		Experimental	Analytical	Experimental	Analytical
1	0	0.8 (20.3)	1.68 (42.7)	3 050	2 280
2	0.2	1.4 (35.6)	2.24 (56.9)	8 420	7 540
3	0.5	1.4 (35.6)	2.99 (75.9)	42 500	49 600
4	0.67	1.8 (45.7)	3.34 (84.8)	154 000	179 200

For $R = 0$ and 0.2, the analytical method gave a conservative life estimate, while it overestimated N_f for R-ratios of 0.5 and 0.67. The inconsistency in fatigue cycle estimates may be attributed to several factors including the loading frequency and plastic zone size at the crack tip. The loading frequency was not consistent across all experiments, and analysis of its effects is outside the scope of this thesis.

Fatigue cycles may also be affected by the plastic zone ahead of the crack tip, which may require the use a J-integral analysis using Elastic Plastic Fracture Mechanics (EPFM). Plastic zone size was analyzed by Rice [73] in 1967 who developed the formula for the plane stress plastic zone radius, $r_{p\sigma}$, presented in Eq. (4.6) for plane stress conditions. A flow chart provided by Dowling in Chapter 8 of [8] provides a logical progression to determine if LEFM is applicable in the presence of the plastic zone. The path followed in the flow chart in [8] is described as follows. First, for the analysis to be plane strain Eq. (4.7) must be satisfied for all plate dimensions listed, where t is the thickness, h is the half-height, b is the half-width, a is the crack half-length, and $K = K(a)$. Eq. (4.7) was not satisfied for any of the loading conditions since the right hand side of the equation was greater than $t = 0.09\,\text{in.}$ (2.29 mm) for all a's, therefore a state of plane stress exists.

$$r_{p\sigma} = \frac{1}{2\pi}\left(\frac{K}{\sigma_{yld}}\right)^2 \tag{4.6}$$

$$t,\ (b-a),\ h \geq 2.5\left(\frac{K}{\sigma_{yld}}\right)^2 \tag{4.7}$$

The next criterion in Eq. (4.8) was used to determine if the region of K-dominance was affected by the geometry of the plate. In the region of K-dominance, limited plasticity at the crack tip can be neglected, and SIFs adequately characterize the stress field. The K-dominance region must be small relative to the distance to any of the listed geometric boundaries. A distance of $8r_{p\sigma}$ is typically acceptable [8]. If a geometric boundary intrudes on the K-dominance region, then gross yielding may occur and an EPFM analysis may be warranted. Eq. (4.8) was satisfied for all loading conditions. Therefore, K continues

to describe the stress field surrounding the crack tip and LEFM is still applicable for the analysis in this thesis.

$$a,\ (b-a)\,,\ h \geq 8r_{p\sigma} \tag{4.8}$$

Probably the most likely reason for the discrepancy in fatigue fracture life is the fact that only two tests were run at each loading condition, and only the average results were provided in Hudson's report. As Virkler et al. [63] showed from sixty-eight fatigue crack growth tests, there is large variability in growth rates for the same material, geometry, and loading condition. As a result it is not likely to draw any definitive conclusions from a two specimen test.

Fig. 4.2 shows a plot of crack length versus fatigue cycle count for the analytical method and experimental data with $R = 0.2$. To build the analytical curve, Eq. (4.4) was incrementally solved between a_i and a_c until a smooth curve was formed. Experimental data was plotted directly from Table A.2. The experimental data has a Root Mean Square Error (RMSE) of 13.2% calculated with Eq. (4.9), where n is the number of samples. The maximum error calculated with Eq. (4.10) was 15.6% compared to the analytical curve. Eqs. (4.9) and (4.10) are used throughout the remainder of this thesis, with the appropriate substitutions for N. Results for all loading conditions can be found in Appendix D.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(\frac{N_{Current\,Analysis}}{N_{Analytical\,Result}} - 1\right)_k^2} \times 100\% \tag{4.9}$$

$$\%\ \text{Error} = \max\left(\frac{N_{Current\,Analysis}}{N_{Analytical\,Result}} - 1\right) \times 100\% \tag{4.10}$$

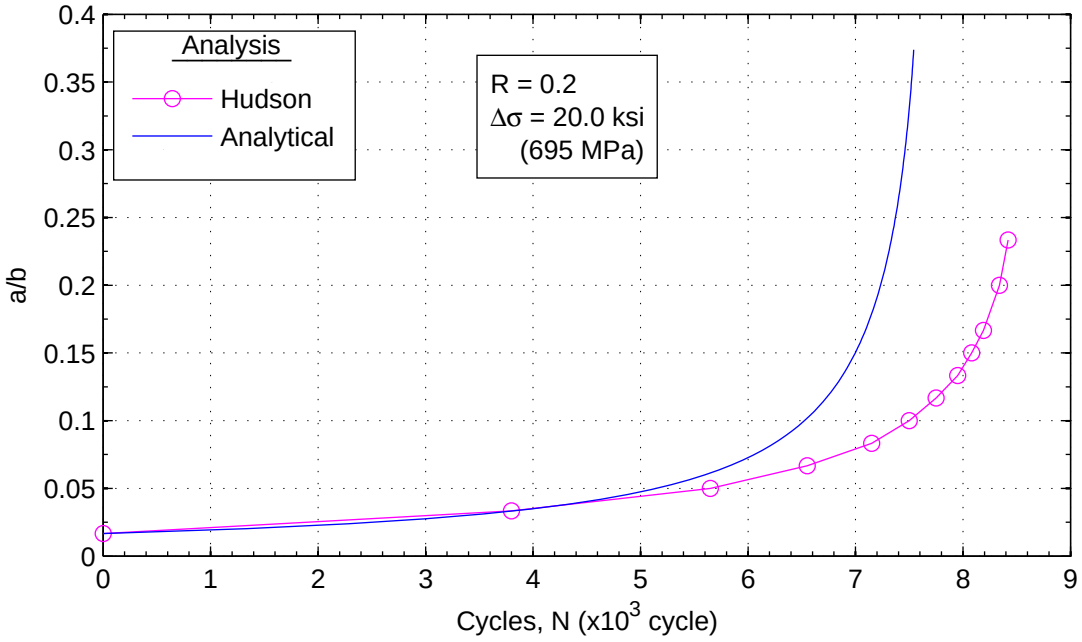

Fig. 4.2: Comparison of cycle count during crack propagation for the analytical method and experimental data ($R = 0.2$).

4.3 AFGROW

AFGROW was used as a comparison tool to a well-established commercial code in the aerospace, mechanical, and civil industries. Conceptually, all that was needed for this analysis was specimen geometry, a loading spectrum, and material properties. A description of the model setup and a comparison to the analytical method are provided in the following sections.

4.3.1 Model Setup

The center cracked plate is a classic, predefined model in AFGROW. Required inputs include the geometric dimensions shown in Fig. 4.1, the elastic material properties and Walker equation constants listed in Table 4.3, and a constant amplitude loading spectrum. All elastic material properties are from Hudson's report unless otherwise noted. Material properties that did not apply to the crack analysis in this thesis were left at default values. The loading spectrum consisted of the maximum stress, σ_{max}, and R-ratio listed in Table 4.4. AFGROW processed crack growth incrementally by increasing the crack length by a length of 0.01 in. until K_c was reached. Cycles were summed over each increment for the final estimate of fatigue fracture life.

Table 4.3: AFGROW Material Data

AFGROW Material Data		
$C^{(1)}$	$8.63 \times 10^{-10} \frac{\text{in./cycle}}{\left(\text{ksi } \sqrt{\text{in.}}\right)^{m/2}}$	
$n^{(2)}$	3.84	
$m^{(3)}$	0.564	
E	10100 ksi	
$\nu^{(4)}$	0.33	
Coeff. Thermal Expan.	1.25E-05 in./in./°F	(default)
σ_{yld}	75.9 ksi	
K_c	72.012 ksi$\sqrt{\text{in.}}$	
${K_{Ic}}^{(4)}$	26.0 ksi$\sqrt{\text{in.}}$	
ΔK_{th}	2.0 ksi$\sqrt{\text{in.}}$	(default)
Lower Limit on R shift	-0.3	(default)
Upper limit on R shift	0.8	(default)

(1) C_0 in Walker equation Eq. (4.1)
(2) m in Walker equation Eq. (4.1)
(3) γ in Walker equation Eq. (4.1)
(4) Found at www.matweb.com [74]

4.3.2 Results

Critical crack lengths and fatigue fracture life estimates for the AFGROW analysis are listed in Table 4.5. Compared to the analytical method, critical crack lengths are within 0.6% and life estimates are within 2.5%. A plot of the normalized stress intensity range for $R = 0.2$ in Fig. 4.3 shows that the stress intensity factors compare well with those from the analytical method, with an RMSE of 0.4% and a maximum error 0.8% at the final crack length. Discrepancies in ΔK_I values were a result of different equations for the dimensionless geometry factor, F. Additionally, the error in ΔK_I's are within an acceptable range since Tada's equation for F, Eq. (3.3), was only expected to be accurate within 1% of experimental data. Fig. 4.4 shows that the cycle count during crack propagation for $R = 0.2$ compare well, with an RMSE of 2.1% and a maximum error of -2.2% at the final crack length. Similar results for ΔK, a_c, and N_f were found for all loading conditions in Appendix D.

Table 4.4: AFGROW Load Spectrum

Loading Condition	σ_{max} (ksi)	(MPa)	R
1	30.0	207	0
2	25.0	172	0.2
3	20.0	138	0.5
4	18.0	124	0.67

Table 4.5: Results from the AFGROW Fatigue Fracture Analysis

Loading Condition	R	Critical Crack Length, in. (mm)						Fatigue Fracture Life, cycles		
		Experimental		Analytical		AFGROW		Experimental	Analytical	AFGROW
1	0	0.8	(20.3)	1.68	(42.9)	1.69	(42.7)	3 050	2 280	2 220
2	0.2	1.4	(35.6)	2.24	(56.6)	2.23	(56.9)	8 420	7 540	7 370
3	0.5	1.4	(35.6)	2.99	(76.2)	3.00	(76.3)	42 500	49 600	48 510
4	0.67	1.8	(45.7)	3.34	(84.7)	3.33	(85.1)	154 000	179 200	175 300

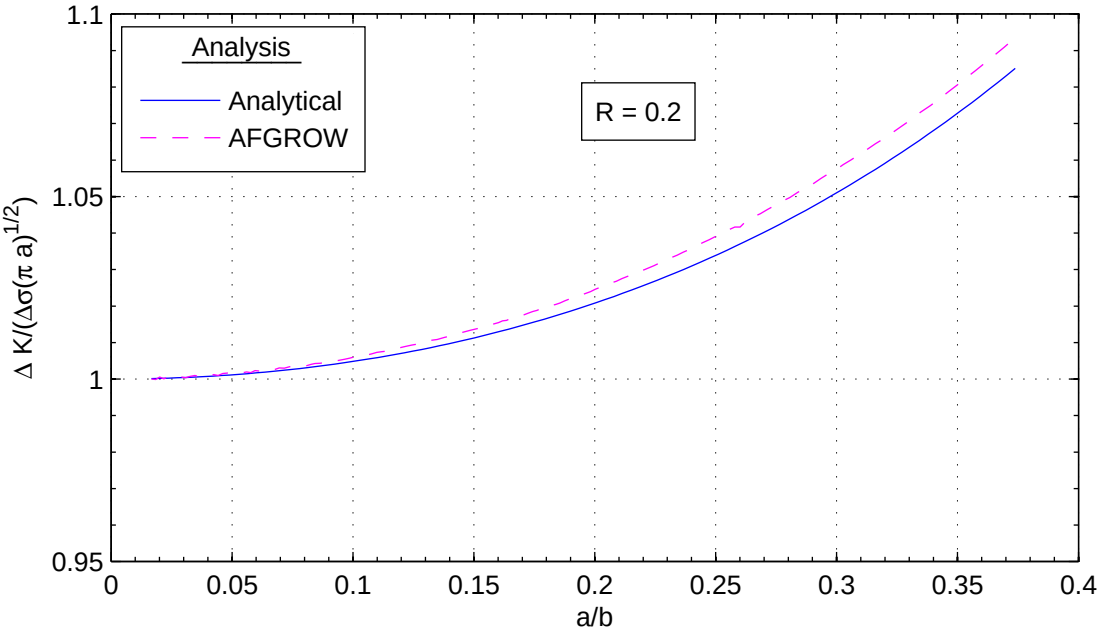

Fig. 4.3: Comparison of the normalized ΔK_I during crack propagation for the analytical method and AFGROW ($R = 0.2$).

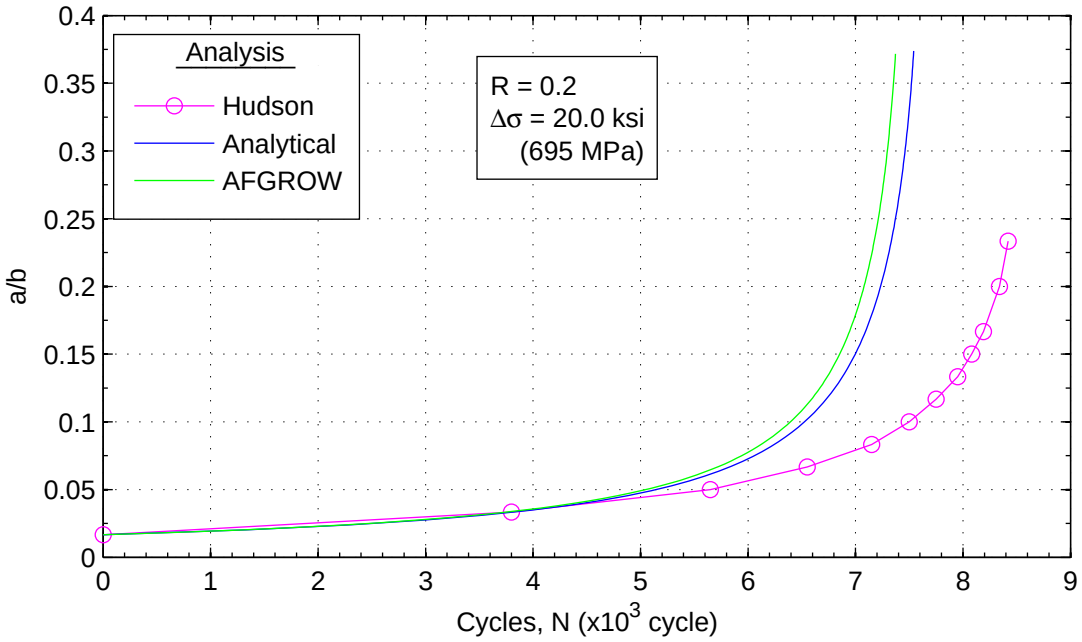

Fig. 4.4: Comparison of cycle count during crack propagation for the analytical method and AFGROW ($R = 0.2$).

4.4 Finite Elements with Quarter Points (FEQP)

The first FEM explored uses the method of moving the mid-side nodes of elements adjacent to the crack tip to their quarter point location. This method was selected because it represents a classical FEM solution to a fracture mechanics problem, as mentioned in Section 2.2.2. Only static loads are possible, and the crack must be manually grown and remeshed at each crack increment. Finite element analyses were performed in Abaqus 6-13.3. Results from Abaqus were post processed in MATLAB to estimate the cycles to failure, N_f. The following sections discuss the model setup, mesh convergence analysis, and results compared to the analytical equations..

4.4.1 Model Setup

A two-dimensional half model was developed due to the symmetry condition about the vertical mid-plane of the plate. Quarter models could not be used due to the limitations of the contour integral used by Abaqus to calculate K. Zero vertical displacement was specified along the bottom plate edge and zero horizontal displacement along the symmetry mid-plane. A pressure load was applied along the top surface of the plate to simulate a uniform nominal axial stress. Material properties included E and ν were assigned with a section thickness of 0.09 inches (2.29 mm).

A focused meshing strategy was employed to increase the number of elements in the vicinity of the crack tip, requiring the partitioning scheme in Fig. 4.5 where the crack has been highlighted in red. The model used a structured meshing algorithm everywhere, except in the focused region where a sweeped meshing algorithm was used with element density biased toward the crack tip.

Two-dimensional plane stress, quadratic quadrilateral elements (CPS8) were selected. Quadratic elements were required since they have the mid-side nodes needed for the FEQP analysis. Edges of elements adjacent to the crack tip were collapsed to a single point, and mid-side nodes were relocated to the element's quarter point location to better represent the stress singularity at the crack tip. The resulting course mesh is shown in Fig. 4.6.

Circular partitioning used in the meshing strategy serves a dual purpose. Abaqus evaluates the J-integral using contour integrals. A ring of elements around the crack tip are selected for this calculation. Building the model so the elements fall within a regular circular region helps avoid fluctuations in the results. SIFs are extracted from J for a linear elastic material using Eq. (4.11) [31].

$$J = \mathcal{G} = \frac{1}{E}\left(K_I^2 + K_{II}^2\right) + \frac{1}{2G}K_{III}^2 \tag{4.11}$$

An FEQP analysis does not allow for crack propagation, thus the crack was manually grown from a_i to a_c. With this limitation, the Crack Propagation Direction (CPD) was assumed to be perpendicular to the vertical mid-plane of the plate and uniform nominal axial stress.

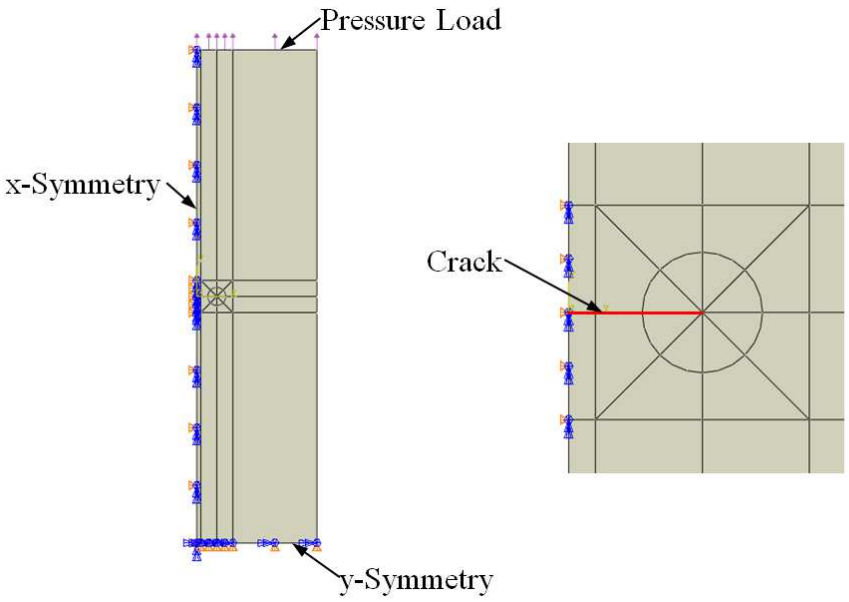

Fig. 4.5: Partitioning, loading, and boundary conditions used in the FEQP model with an inset zoomed in at the crack tip.

This assumes that the crack would propagate by pure Mode I fracture. Each crack increment required a new FEQP model which was remeshed to the new geometry. A modified Simpson's rule was used to select the minimum number of crack lengths needed to adequately estimate life. In Chapter 11 of [8], Dowling suggests using the modified Simpson's rule presented in Eqs. (4.12) and (4.13) to numerically integrate Eq. (4.4). This set of equations places more crack increments, a_n, close to a_i where the slope of the $y = dN/da$ is steepest, and gradually increase the spacing as the slope flattens at larger crack lengths. r was found by selecting an even number of crack lengths, n, starting at zero for the initial crack length and solving Eq. (4.12) with $a_n = a_c$. The value of n was selected by increasing its value in even increments until the result of Eq. (4.13) did not change significantly.

$$a_n = r^n a_i \tag{4.12}$$

$$\int_{a_j}^{a_{j+2}} y\,da = \sum_{j=0}^{n-2} \frac{a_j\left(r^2-1\right)}{6r}\left[y_j r\left(2-r\right) + y_{j+1}\left(r+1\right)^2 + y_{j+2}\left(2r-1\right)\right] \tag{4.13}$$

Abaqus calculates the CPD based on the Maximum Tangential Stress (MTS), Maximum Energy Release Rate (MERR), and $K_{II} = 0$ crack extension criteria mentioned in Section 2.1.5. CPD results are used here to confirm the previous crack growth direction assumption. Abaqus' q-vector, which describe the current crack extension direction, was set parallel to the crack plane for each model which is essential to extract meaningful SIFs.

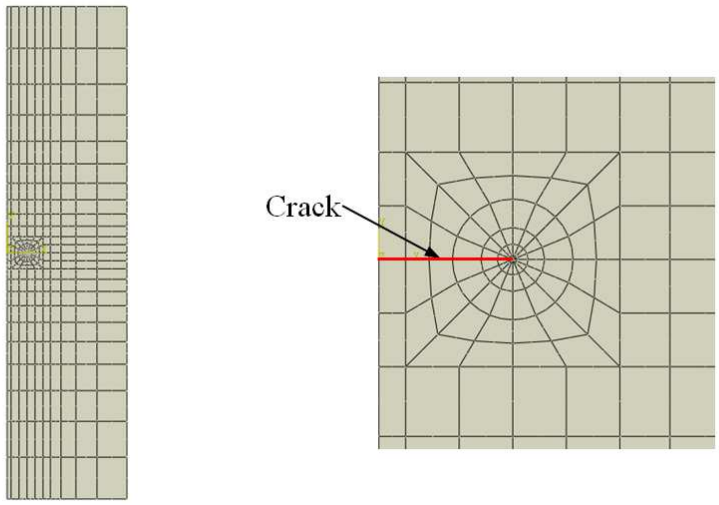

Fig. 4.6: Coarse meshing scheme for the FEQP model with an inset zoomed in at the crack tip.

Considering the time it takes to build and remesh the models for each of the loading conditions and crack lengths, it was decided to parametrize the FEQP model. A script was developed in MATLAB and Python to access Abaqus' Complete Abaqus Environment (CAE) file to adjust the partitions, remesh the model, and run the analysis. Stress intensity factors were extracted from the output database (ODB) files, and the values were averaged excluding the first and second contours as these typically fluctuated compared to other contours which were approximately constant. The fluctuation in SIFs for the first and second contour was a result of the stress singularity at the crack tip and the selection of the modified shape functions for the quarter point nodes. The first contour was defined by the nodes at the crack tip, and the second contour included all nodes within elements adjacent to the crack tip. Subsequent contours include additional rings of elements. After the second contour, the effects of the stress singularity are reduced. It is recommended in Abaqus' Analysis User Guide to discard the first few contours that exhibit this fluctuation [5].

4.4.2 Convergence Analysis

Due to the number of models at different crack lengths and loading conditions used in this analysis, three models were selected to be converged with a single loading condition. Convergence of the three models bounds the accuracy of the solution, assuring that the solutions of the intermediate crack lengths will also be converged. Additionally, in a Linear Elastic Fracture Mechanics (LEFM) analysis the stress intensity is proportional to the loading. Crack lengths of 0.1 in. (2.54 mm), 1.0 in. (25.4 mm), and 3.0 in. (76.2 mm) were selected to be the convergence models with $R = 0$.

All three models converged immediately with average ΔK_I values less than 1% error relative to the analytical results for the first mesh which is within an acceptable error since Tada's

equation for F, Eq. (3.3), was only expected to be accurate within 1% of experimental data. Successive mesh refinements showed less than a 0.1% difference in ΔK_I. Percent differences were calculated with Eq. (4.14), which will be used throughout the remainder of this thesis with appropriate values substituted for ΔK. Table 4.6 lists the percent error in average ΔK_I values relative to the analytical method and percent difference between mesh refinements. Mesh refinements were based on the number of elements along the radial distance of the focused region, r_e.

$$\% \text{ Diff} = \max\left(\frac{\Delta K_{Current\,Mesh}}{\Delta K_{Previous\,Mesh}} - 1\right) \times 100\% \tag{4.14}$$

Table 4.6: Convergence analysis for the FEQP model

		Avg.ΔK_I Percent Error Relative to Analytical Results and Percent Difference Between Mesh Refinement					
Crack Length a, in. (mm)		0.1 (2.54)		1.0 (25.4)		3.0 (76.2)	
Analytical ΔK_I ksi$\sqrt{\text{in.}}$ (MPa $\sqrt{\text{mm}}$)		5.61 (195)		18.0 (624)		36.1 (1250)	
Mesh	Num. Radial Elements	% Error	% Diff.	% Error	% Diff.	% Error	% Diff.
1	4	-0.22	-	0.29	-	0.94	-
2	8	-0.12	0.09	0.31	0.02	0.96	0.02
3	16	-0.10	0.02	0.31	0.00	0.96	0.00

Fig. 4.7 shows the normalized ΔK_I variation at different contour intervals, where r is the radius from the crack tip. Contour locations are represented by the markers on each line. The number of contours in each mesh was increased so that all elements in the circular focused region were included in each calculation. Fig. 4.7 shows the fluctuation in ΔK_I values for the first and second contour as a result of the stress singularity and modified shape functions. These two contour values were not included when averaging ΔK_I for the crack length. The RMSE of all contour intervals and each mesh are listed in Table 4.7. Mesh #2 had an RMSE of 0.31% relative to the analytical result. Proceeding to the third mesh does not offer significant improvement in the RMSE. Results for convergence models with $a = 0.1$ and 3.0 in. had similar ΔK_I variations, and the figures are found in Appendix C.

Table 4.7: RMSE of ΔK_I Contour Values at $a = 1.0$ in. (25.4 mm) in Fig. 4.7.

Mesh	ΔK_I RMSE Relative to Analytical Results (%)
1	0.33
2	0.31
3	0.31

The average ΔK_I values for first mesh were in excellent agreement with the analytical method, but the second mesh was selected for added confidence that the mesh density would

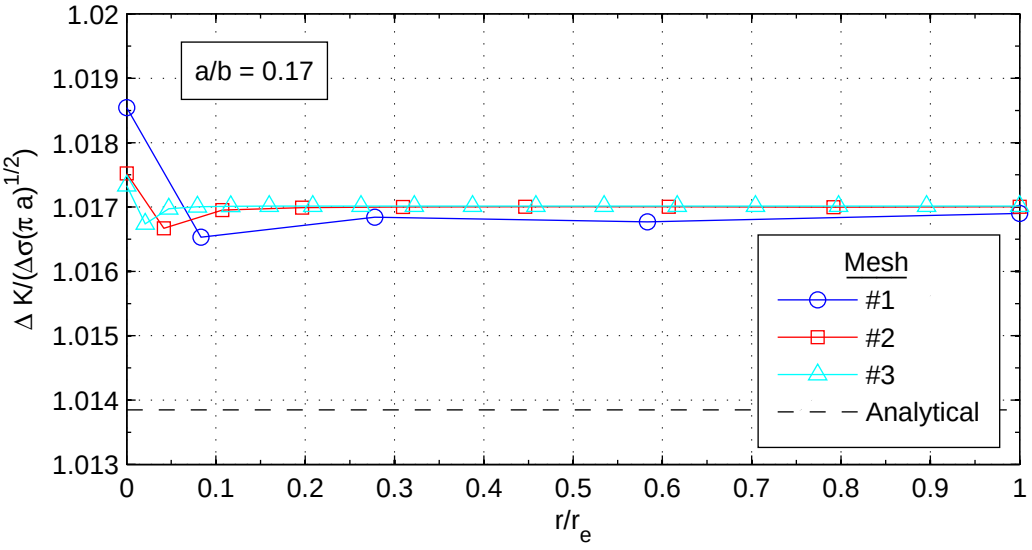

Fig. 4.7: FEQP normalized ΔK_I variation with contour intervals: $a = 1.0\,\text{in.}$ (25.4 mm)

be adequate for the intermediate crack lengths. Additionally, more ΔK_I contour values were averaged with this selection. All models solved in under a minute so the added processing time was insignificant.

4.4.3 Results

Based on the modified Simpson's rule analysis, it was found that ten crack lengths are adequate to estimate life. However, twenty crack lengths were used for added fidelity in the crack growth curves. Critical crack lengths for this analysis were based on the analytical method since these were needed to setup the models. Therefore, critical crack lengths for the FEQP and analytical analyses are the same, as seen in Table 4.8. Fatigue fracture life estimates compare well with the analytical method with errors less than 0.5%.

Table 4.8: Results from the FEQP Fatigue Fracture Analysis

Loading Condition	R	Critical Crack Length, a_c, in. (mm)			Fatigue Fracture Life, N_f, cycles		
		Experimental	Analytical	FEQP	Experimental	Analytical	FEQP
1	0	0.8 (20.3)	1.68 (42.7)	1.68 (42.7)	3 050	2 280	2 270
2	0.2	1.4 (35.6)	2.24 (56.9)	2.24 (56.9)	8 420	7 540	7 530
3	0.5	1.4 (35.6)	2.99 (75.9)	2.99 (75.9)	42 500	49 600	49 490
4	0.67	1.8 (45.7)	3.34 (84.8)	3.34 (84.8)	154 000	179 200	178 780

Plots of the normalized stress intensity variation and cycle count during crack growth for $R = 0.2$ are shown in Figs. 4.8 and 4.9, respectively. SIFs compare well as expected from the convergence analysis with an RMSE of 0.3% and maximum error of 0.8% at the final crack length, both within an acceptable range compared to Tada's equation for F. As a result of

the minimal error in the stress intensity factors, the crack length versus cycle count was also in excellent agreement with the analytical method with an RMSE of 0.1% and a maximum error of -0.2% at the final crack length. Results in Appendix B for the remaining loading conditions show similar behavior for both SIFs and fatigue cycles versus crack length.

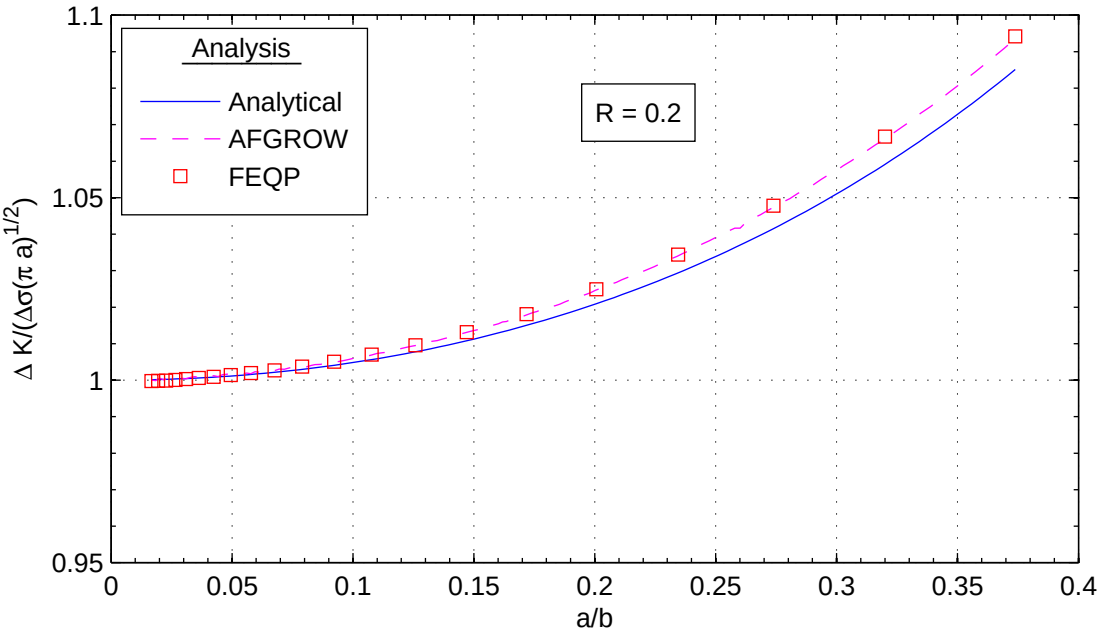

Fig. 4.8: Comparison of the normalized ΔK_I during crack propagation for the analytical and FEQP methods ($R = 0.2$).

A CPD analysis shows that the MTS, MERR, and $K_{II} = 0$ methods resulted in crack extension angles of zero degrees for all $a/b < 0.3$. Crack extension angles for $a/b > 0.3$ do not exceed 0.05°, which is a result of numerical noise and was ignored. This analysis confirms the prior assumption that pure Mode I crack growth occurs perpendicular to the mid-plane of the plate. CPD angles were plotted in Fig. 4.10 for $R = 0.2$ and other loading conditions are found in Appendix B.

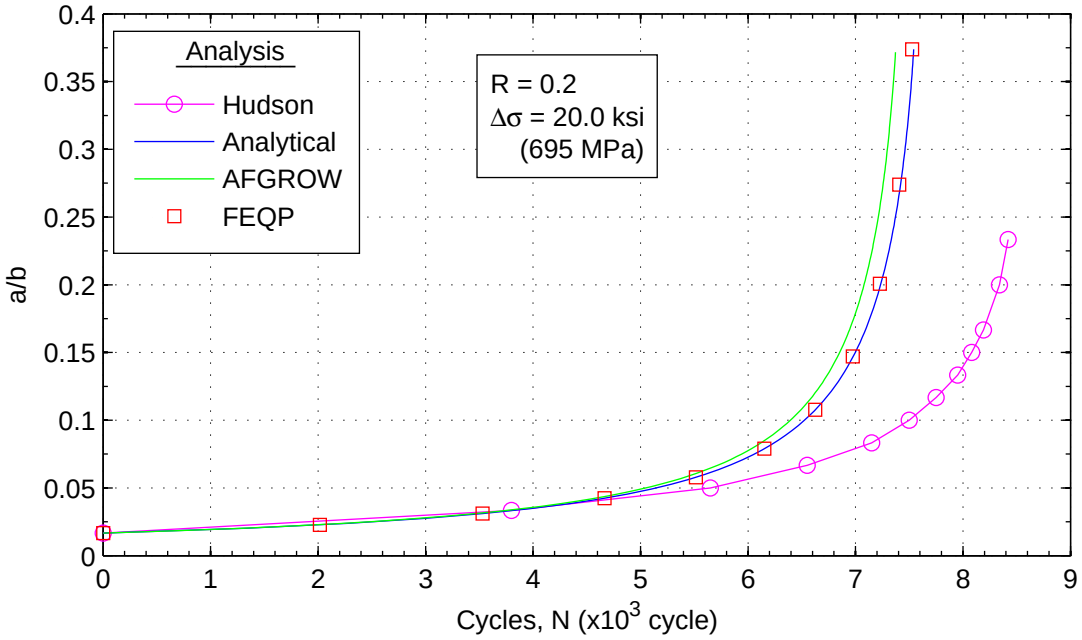

Fig. 4.9: Comparison of cycle count during crack propagation for the analytical and FEQP methods ($R = 0.2$).

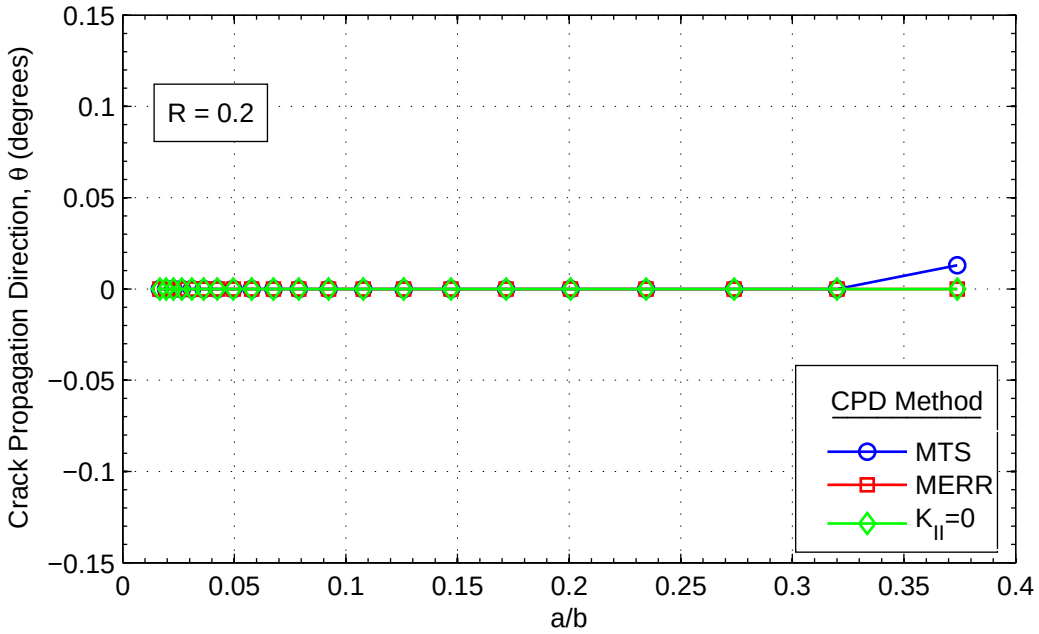

Fig. 4.10: Comparison of crack propagation direction for the MTS, MERR, and $K_{II} = 0$ criterion in the FEQP analysis ($R = 0.2$).

4.5 Virtual Crack Closure Technique (VCCT)

A goal of this thesis was to evaluate XFEM-PN's ability to model crack propagation. The VCCT analysis serves as a logical step to this goal since XFEM-PN uses a *modified* VCCT method which will be discussed in Section 4.6. Abaqus' VCCT low-cycle fatigue analysis allows for automated crack propagation along a predefined path using a single model with no remeshing. Principles of LEFM were used to propagate the crack which provided a parallel comparison to the previous methods.

Analyses were performed on an IBM iDataPlex system due to the increased memory requirements and solution time. Abaqus 6-13.1 was available on the system; however, the results of the first mesh were compared to those from Abaqus 6-13.3 and no differences were observed.

The following sections discuss the model setup, mesh convergence, and results.

4.5.1 Model Setup

A two-dimensional half model was developed in this analysis using two parts; similar top and bottom halves of the plate. Each part had the same material properties and section as defined in the FEQP analysis. The halves of the plate were placed in surface-to-surface contact; specifying the propagation plane, labeled in Fig. 4.11, as the initially bonded region. The unbonded portion represents the initial crack. The assembled model had the same boundary conditions as the FEQP model.

The goal of the focused meshing strategy in the VCCT model was to maximize the number of elements along the propagation plane while reducing the density elsewhere, as it was expected to require many elements to converge N_f. The partitioning scheme shown in Fig. 4.11 allows the number of elements to gradually and uniformly increase while using a structured meshing algorithm. Several of these square patterns were stacked together to build a 12:1 element transition, meaning for every one partition at the top and bottom plate edges, there are twelve partitions at the crack propagation plane.

Two-dimensional plane stress, linear quadrilateral elements (CPS4) were selected since these elements are the only quadrilateral elements supported by VCCT in a low-cycle fatigue analysis [5]. Tetrahedron elements are supported; however, these elements tend to be too stiff. The resulting course mesh is shown in Fig. 4.12.

Two steps were required in this analysis: a static general, and direct cyclic step. The static general step serves two purposes: first, it initializes the crack prior to the direct cyclic step, and second, it loads the structure up to σ_{max} so that the cyclic load can oscillate between σ_{max} and σ_{min}. An example of the loading amplitude curve is shown in Fig. 4.13 for $R = 0.2$. Defining a sinusoidal curve tended to reduce the stabilization ratios, discussed next, by an order or two of magnitude compared to a triangular shaped curve. Assisting stabilization of each cycle reduced the computation times.

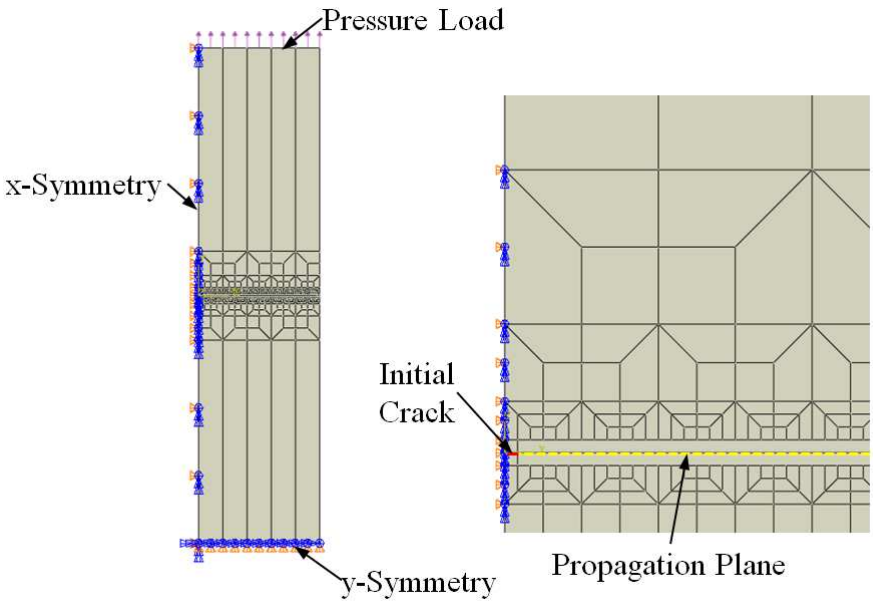

Fig. 4.11: Partitioning, loading, and boundary conditions used in the VCCT model with an inset zoomed in at the initial crack.

Application of a cyclic load requires Abaqus' direct cyclic solver, which uses a truncated Fourier series to solve for the displacement at each node at all time increments. Differences between the applied load and the internal forces, known as the residuals, are also calculated at each node and time increment. Abaqus' representations for nodal displacement and residuals based on a truncated Fourier Series are presented in Eqs. (4.15) and (4.16), respectively. Displacement coefficients are corrected after each iteration until a stabilized solution is found. The number of Fourier terms, n, can increase between iterations. Stabilized solutions are defined by four ratios which must be below a set tolerance. Stabilization ratios are defined as: (1) the ratios of the largest correction to the displacement coefficients over the largest displacement coefficient for the constant, CU_0, and periodic, CU_n, terms and (2) the ratios of the largest residual coefficient to the time average force for the constant, CR_0, and periodic, CR_n, terms. [5, 31]

$$\overline{u}(t) = u_0 + \sum_{k=1}^{n} \left(u_k^s \sin k\omega t + u_k^c \cos k\omega t \right) \tag{4.15}$$

$$\overline{R}(t) = R_0 + \sum_{k=1}^{n} \left(R_k^s \sin k\omega t + R_k^c \cos k\omega t \right) \tag{4.16}$$

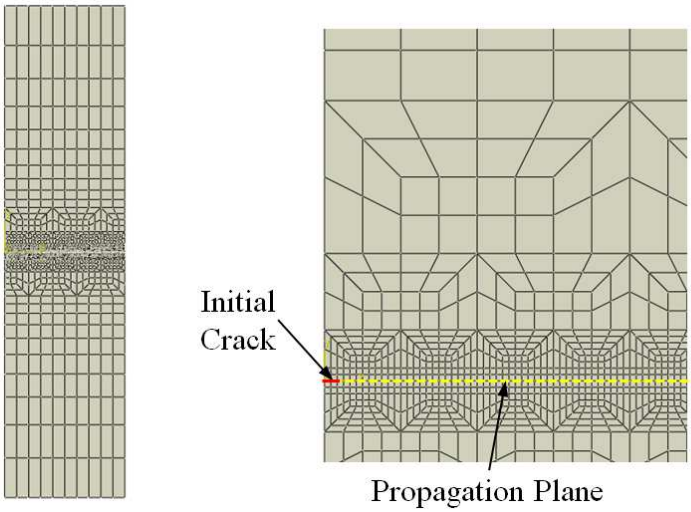

Fig. 4.12: Coarse meshing scheme for the VCCT model with an inset zoomed in at the initial crack.

where: $\overline{u}$ is the nodal displacement
u_0 is an unknown constant displacement coefficient
$u_k^s,\ u_k^c$ are unknown periodic displacement coefficients
$\overline{R}$ is the nodal residual
R_0 is an unknown constant residual coefficient
$R_k^s,\ R_k^c$ are unknown periodic residual coefficients
ω is angular frequency
t is time.

Several steps were taken to aid in the stabilization of the VCCT low-cycle fatigue analysis. For a propagating crack analysis, Abaqus recommends increasing the parameter I_0, which is the frequency at which residuals of successive iterations are checked to determine if they are increasing. I_0 was increased from a default value of 4 to 8 using the discontinuous analysis option. The discontinuous analysis option also increases I_R, the starting point for the logarithmic rate of convergence checks, from a default of 8 to 10. However, this convergence check was not performed in the direct cyclic step since a fixed time increment was used. I_A was increased from the default value of 5 to a recommended value of 10 in the static step and 20 in the direct cyclic step. The I_A parameter determines the maximum number of cutbacks allowed in an increment. [5, 31, 75]

Processing time and solution accuracy were used as metrics in a tradeoff study to determine the number of Fourier terms, values for stabilization ratios, and time increment size needed for a stabilized cycle. The number of Fourier terms was limited to 12 since the default value of 25 resulted in an increase in processing time while not converging to the stabilization

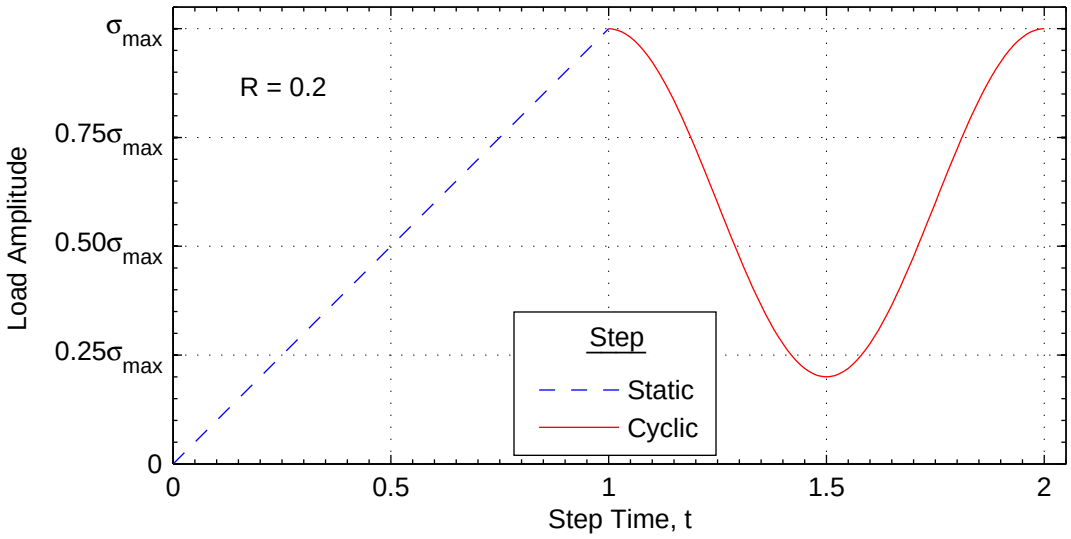

Fig. 4.13: Example load amplitude curve for the static and cyclic steps with $R = 0.2$.

tolerances. Ratios CU_0 and CU_n were left at default values of 0.005 and CR_0 and CR_n where both increased to values between 0.15 to 0.30 depending on mesh density. Additional results for CR_0 can be found in Appendix C. Since no temperature effects were analyzed, Abaqus used a fixed time increment set to 0.02. Larger time increments resulted in destabilized cycles.

Fracture and fatigue criterion were needed for the static and direct cyclic steps, respectively. The Benzeggagh-Kenane, or BK, model [15] was selected to describe the equivalent critical energy release rate, $\mathcal{G}_c$, for a mixed mode fracture analysis. Eq. (4.17) presents the BK law as interpreted by Abaqus [5]. This form of $\mathcal{G}_c$ was selected because failure was assumed to occur from pure Mode I fracture as seen in the FEQP analysis and the following order of magnitude study showed that $\mathcal{G}_c \approx \mathcal{G}_{Ic}$. In a two dimensional problem $\mathcal{G}_{III} = 0$. It was assumed that the CCP in this thesis failed by pure Mode I fracture, so $\mathcal{G}_{II} \ll \mathcal{G}_I$. There was no material data to empirically fit the exponent, η, so a value of unity was assumed. No data was available to determine the pure Mode II critical energy release rate, $\mathcal{G}_{IIc}$. To ensure that $\mathcal{G}_c \approx \mathcal{G}_{Ic}$, it was assumed that $\mathcal{G}_{IIc} \approx \mathcal{G}_{Ic}$. Applying these assumptions to Eq. (4.17) results in $\mathcal{G}_c \approx \mathcal{G}_{Ic}$ as needed for this analysis.

$$\mathcal{G}_c = \mathcal{G}_{Ic} + (\mathcal{G}_{IIc} - \mathcal{G}_{Ic}) \left(\frac{\mathcal{G}_{II} + \mathcal{G}_{III}}{\mathcal{G}_I + \mathcal{G}_{II} + \mathcal{G}_{III}} \right)^{\eta} \tag{4.17}$$

The fatigue crack propagation portion of the analysis also needed a definition for crack growth. Abaqus has defined the onset of fracture with Eq. (4.18) and growth defined by the Paris equation, modified for the energy release rate range, $\Delta\mathcal{G}$, in Eq. (4.19). Constants c_1, c_2, c_3, and c_4 are empirically determined based on material and geometry; and the energy release rate range is defined as $\Delta\mathcal{G} = \mathcal{G}_{max} - \mathcal{G}_{min}$ [5]. Constants c_1 and c_2 were both set to zero in this study since no data was available to fit them and this would assume growth to

start immediately. To solve for constants c_3 and c_4, Eq. (4.19) was set equal to Eq. (4.1). Using $\Delta\mathcal{G} = \mathcal{G}_{max}(1 - R^2)$ and the relation between $\mathcal{G}$ and K, the relationship between Abaqus' constants and Walker equation constants was formed in Eq. (4.20). Values of c_3 and c_4 for each loading condition are listed in Table 4.9.

$$N = c_1 \left(\Delta\mathcal{G}\right)^{c_2} \tag{4.18}$$

$$\frac{da}{dN} = c_3 \left(\Delta\mathcal{G}\right)^{c_4} \tag{4.19}$$

$$c_3 = C_0(1 - R)^{\gamma m} \left(\frac{E}{1 - R^2}\right)^{m/2} \tag{4.20a}$$

$$c_4 = \frac{m}{2} \tag{4.20b}$$

Table 4.9: Abaqus Crack Growth Constants.

Loading Condition	R	c_3 $\frac{\text{in./cycle}}{(\text{ksi in.})^{c_4}}$	c_3 $\frac{\text{mm/cycle}}{(\text{MPa mm})^{c_4}}$	c_4
1	0	4.19×10^{-2}	5.26×10^{-5}	1.92
2	0.2	2.79×10^{-2}	3.51×10^{-5}	1.92
3	0.5	1.62×10^{-2}	2.04×10^{-5}	1.92
4	0.67	1.20×10^{-2}	1.51×10^{-5}	1.92

Abaqus requires bounds on the range of $\Delta\mathcal{G}$ applied to the Paris equation since the equation is only valid in the Paris region of crack growth. The energy release rate threshold, $\mathcal{G}_{th}$, defines the lower bound where no crack growth occurs and the Paris limit energy release rate, $\mathcal{G}_{pl}$, defines the upper bound where the crack growth rate enters the unstable growth region [5]. Dowling [8] recommends using the critical stress intensity as the upper bound to this region; thus, $\mathcal{G}_{pl}/\mathcal{G}_c$ was set to 0.999, allowing growth to occur up until fracture at $\mathcal{G}_{pl} \approx \mathcal{G}_{Ic}$. Since data provided in Hudson's report did not exhibit significantly slower growth rates approaching a threshold value, the ratio of $\mathcal{G}_{th}/\mathcal{G}_c$ was set to 0.001, generally allowing crack growth to occur at all applied loads.

During Abaqus' VCCT fatigue crack growth analysis, values of $\Delta\mathcal{G}$ are calculated as nodal quantities at the crack tip with Eq. (4.21), restated here and defined in Section 2.2.2. Inserting $\Delta\mathcal{G}$ into Eq. (4.19) and using the characteristic element length of the model, the number of cycles required to fracture the element ahead of the crack tip are calculated. In a three-dimensional model, the element, or elements, with the least number of cycles remaining within a given tolerance are fractured [5]. For a VCCT analysis, this means the contacting nodes on the propagation plane are released. In a two-dimensional model, there is only one

element ahead of the crack tip, thus only one element can be fractured during each cycle. Once the nodes have been released, the calculated number of cycles is added to the total cycle count and the analysis repeats, redistributing the load over the reduced cross section. This process continues until the fracture ratio, $\mathcal{G}_{pl}/\mathcal{G}_c$, is reached, at which point propagation occurs in one cycle increments until reaching the user defined number of cycles to analyze [5].

$$\mathcal{G}_I = \frac{1}{2\Delta a} F_{y,cd} u_{y,ab} \tag{4.21a}$$

$$\mathcal{G}_{II} = \frac{1}{2\Delta a} F_{x,cd} u_{x,ab} \tag{4.21b}$$

$$\mathcal{G}_{III} = \frac{1}{2\Delta a} F_{z,cd} u_{z,ab} \tag{4.21c}$$

One of the objectives of this thesis is to compare the estimated life of different analysis procedures. The final cycle count, as calculated by Abaqus, was unknown, and these analyses were computationally expensive with some of the final mesh refinements taking several days to complete. Having an analysis run after it had reached $\mathcal{G}_{pl}/\mathcal{G}_c$ was undesirable. Therefore, a Python code was written to run in parallel with Abaqus to monitor the analysis's status (STA) file, looking for consecutive cycles to increase by one. If this occurred, it was assumed that $\mathcal{G}_{Ic}$ had been reached, fracture had occurred, and the analysis could be terminated. This code is provided in Appendix G.

Fatigue crack propagation was achieved by coupling VCCT, the direct cyclic solver, and the fatigue and fracture criterion; however, this operation is not directly supported by Abaqus CAE. All definitions made in the direct cyclic step were manually added to the input (INP) file.

4.5.2 Convergence Analysis

Two convergence criteria were used for the VCCT analysis: the stress intensity range, ΔK_I, as the crack propagated, and the final cycles at failure, N_f. Abaqus reports the energy release rate range, $\Delta\mathcal{G}$, and Eq. (4.22) was used to convert to ΔK based on Abaqus' definition of $\Delta\mathcal{G} = \mathcal{G}_{max} - \mathcal{G}_{min}$. Loading condition #1 with $R = 0$ was used, similar to the FEQP convergence analysis.

$$\Delta K = \sqrt{\frac{1-R}{1+R} E \Delta\mathcal{G}} \tag{4.22}$$

Table 4.10 lists the RMSE and maximum error of ΔK_I, calculated with Eqs. (4.9) and (4.10), as the crack propagates from a_i through a_c where $K_{max} = K_c$. Abaqus does not terminate the analysis when $K_{max} = K_c$ so results were truncated to meet this condition. Error calculations are based on the analytical method. As shown in Fig. 4.14, the maximum

errors occur with small a/b ratios. By the third mesh, nodal errors are less than 2% for $a/b \geq 0.03$, and the RMSE including all nodes is 1.0%. The ΔK_I RMSE is within an acceptable range compared to Tada's Eq. (3.3) which was only expected to be accurate within 1% of experimental data. Fig. 4.14 indicates the importance of a fine mesh with short crack length ratios.

Percent differences in Table 4.10 were calculated between consecutive meshes at nodal locations consistent with Mesh #1. The Root Mean Square Difference (RMSD) calculation is presented in Eq. (4.23), where n is the total number of nodes in Mesh #1. Maximum differences were calculated with Eq. (4.14). RMSD were 0.5% between Mesh #3 and #4. RMSE and RMSD for ΔK_I in Mesh #3 indicate that this mesh would be sufficient for the first metric of this analysis.

$$\text{RMSD} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\left(\frac{\Delta K_{Current\,Mesh}}{\Delta K_{Previous\,Mesh}} - 1\right)_k^2} \times 100\% \tag{4.23}$$

Table 4.10: Stress Intensity Range Convergence Analysis.

Mesh	Elements Along Propagation Plane	ΔK_I % Error		ΔK_I % Difference	
		Max.	RMSE	Max.	RMSD
1	120	-17	4.5	-	-
2	240	-7.3	2.0	11	2.7
3	480	-3.5	1.0	4.1	1.0
4	960	-1.8	0.9	1.8	0.5

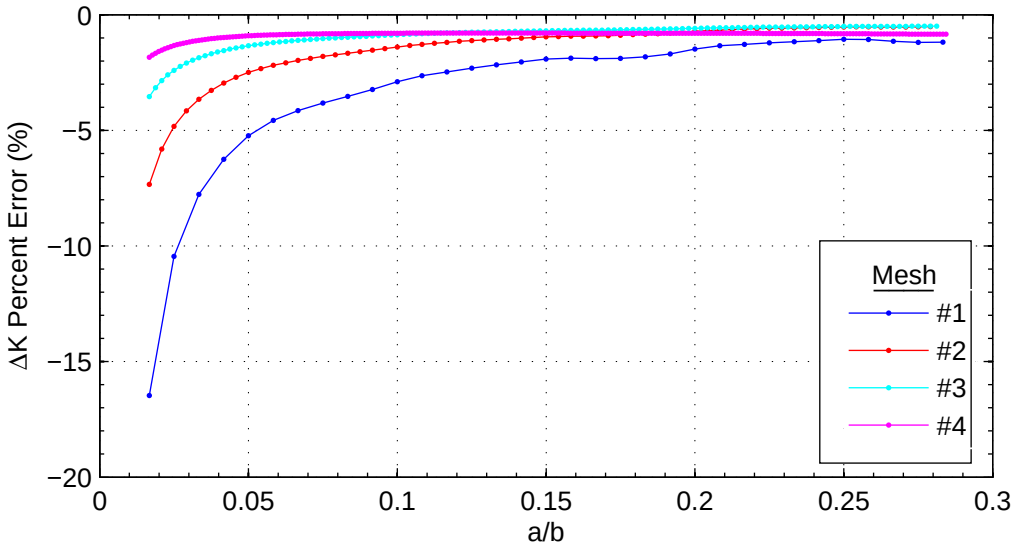

Fig. 4.14: Error in stress intensity range, ΔK_I, during crack propagation in VCCT analysis.

Fatigue fracture life estimates were the second metric for convergence, and results of the analysis are listed in Table 4.11. Percent error was based on the analytical method, and percent difference was between consecutive meshes. Two methods of computing life estimates were used in this analysis. The first estimate is the fatigue cycle as reported by Abaqus when $K_{max} = K_c$. The second method, the integrated polynomial method, integrates the da/dN curve. This was achieved by fitting a quadratic polynomial to three consecutive points with the nodal location on the abscissa and da/dN calculated from Eq. (4.19) on the ordinate. The incremental curves were integrated and the results summed over the crack length, resulting in N_f. This method is similar to the previously described modified Simpson's rule used in the FEQP analysis; however, in general this method does not restrict the location of middle point since it's location is mesh dependent. This was performed in MATLAB with the `polyfit()`, `polyint()`, and `polyval()` functions. Code can be found in Appendix F.

It is obvious from Table 4.11 that the estimated fatigue fracture life is highly mesh dependent, with errors in the Abaqus estimate relative to the analytical results greater than 97% for the first mesh and reducing to 9% by the fourth mesh. Using the integrated polynomial approach shows a significant reduction in error. Fig. 4.15 shows that the integrated polynomial method gives life estimates approximately one mesh refinement before Abaqus' calculation, thus requiring fewer elements and less computational time for a similar answer. Mesh #3's estimate of N_f is within 2.8% of the analytical results. Further refining the mesh shows a 3% change in the answer, while there is a significant increase in processing time.

Table 4.11: Estimated Fatigue Fracture Life Convergence Analysis.

Mesh	Elements Along Propagation Plane	Abaqus, N_f		Integrated Polynomial, N_f		Processing Time (hrs)
		% Error	% Diff.	% Error	% Diff.	
1	120	97	-	38	-	0.5
2	240	34	-32	11	-19	3.0
3	480	15	-14	2.8	-7.4	21
4	960	9.2	-5.3	-0.2	-3.0	184

Mesh #3 was considered converged since RMSE was within 1% for ΔK_I and 3% for N_f using the integrated polynomial method. RMSD's for both metrics do not show a significant improvement by proceeding to Mesh #4. Processing time was also taken into consideration with Mesh #4 taking 184 hours to run, which was not feasible to run multiple models.

4.5.3 Results

Results from the VCCT analysis are shown in Table 4.12. Critical crack lengths for the VCCT analysis compare well with the analytical method with errors less than 0.5% attributed to the discretization of the mesh. Fatigue fracture life estimates, using the integrated polynomial method, are approximately 8% higher than the analytical results, which is a result of a slower growth rate from under estimating ΔK_I at the shorter crack lengths where the most cycles

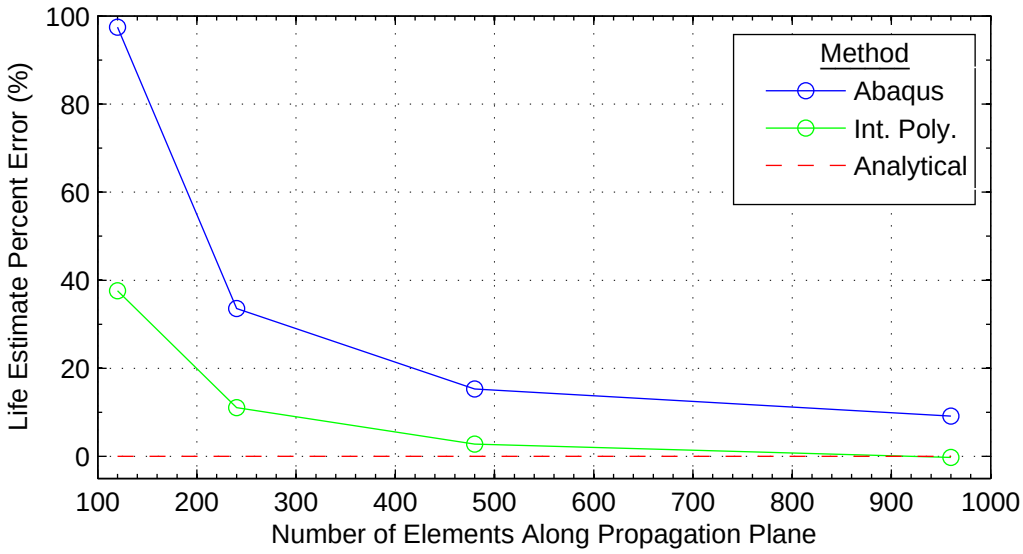

Fig. 4.15: VCCT Convergence plot of estimated fatigue fracture life.

are accumulated. Normalized ΔK_I values are plotted in Fig. 4.16, where the error in shorter crack lengths is obvious. This is a result of the converged mesh selection, where a similar curved shape is seen for Mesh #3 in Fig. 4.14. The RMSE for the $R = 0.2$ load was 0.9% with a maximum error of -3.6% found at the initial crack length. The ΔK_I RMSE is within an acceptable range compared to Tada's dimensionless geometry factor error.

Table 4.12: Results from the FEQP Fatigue Fracture Analysis

Loading Condition	R	Critical Crack Length, in. (mm)						Fatigue Fracture Life, cycles		
		Experimental		Analytical		VCCT		Experimental	Analytical	VCCT
1	0	0.8	(20.3)	1.68	(42.7)	1.69	(42.9)	3 050	2 280	2 460
2	0.2	1.4	(35.6)	2.24	(56.9)	2.25	(57.2)	8 420	7 540	8 150
3	0.5	1.4	(35.6)	2.99	(75.9)	3.00	(76.3)	42 500	49 600	53 630
4	0.67	1.8	(45.7)	3.34	(84.8)	3.35	(85.1)	154 000	179 200	193 900

Crack growth versus cycle count for $R = 0.2$ is shown in Fig. 4.17 with an RMSE of 8.6% and a maximum error of 13% found during the first growth cycle. Similar ΔK_I and fatigue cycle results were found for all R-ratios with figures in Appendix D. No CPD analysis was performed since the crack is restricted to growth along the plane perpendicular to plate's mid-plane. As a check on the solution, all values of $K_{II} \ll K_I$, thus pure Mode I fracture with $\theta = 0$ was expected.

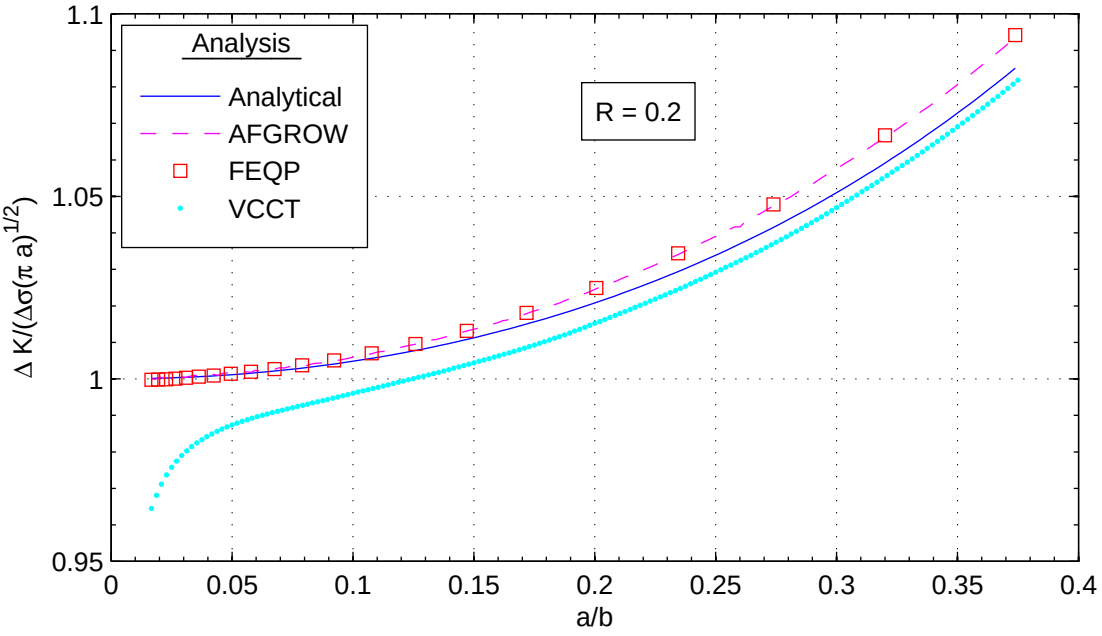

Fig. 4.16: Comparison of the normalized ΔK_I during crack propagation for the analytical and VCCT methods ($R = 0.2$).

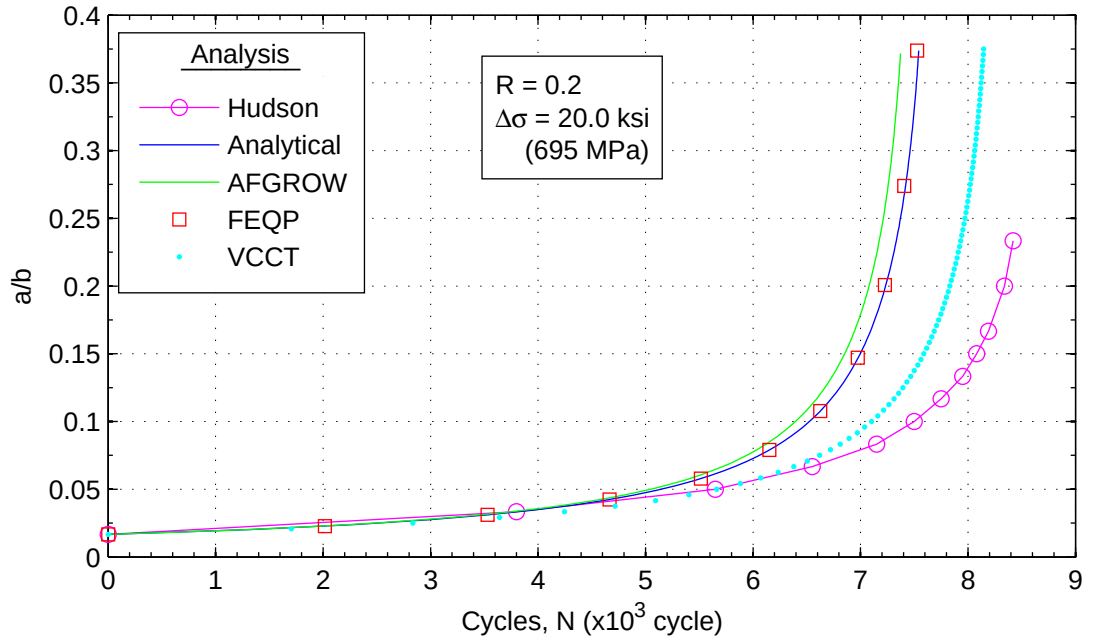

Fig. 4.17: Comparison of cycle count during crack propagation for the analytical and VCCT methods ($R = 0.2$).

4.6 eXtended Finite Element Method with Phantom Nodes (XFEM-PN)

XFEM-PN was the final FEM explored which allows a crack to propagate along a solution dependent path without remeshing. This method is advantageous when analyzing complex loading and geometry where no analytical solution or experimental data are available. In the present analysis, analytical methods and experimental data are available so a direct comparison was made.

XFEM with crack tip enrichment functions are only available in Abaqus 6-13.3 with a non-propagating crack and only in three dimensions. Thus, a comparison to this method is not possible within the scope of this thesis.

Analyses were performed with Abaqus 6-13.3 on a 64-bit personal computer with 20.0 Gb of RAM. Abaqus 6-13.1 on the IBM iDataPlex system was initially used; however, a problem with the Abaqus 6.-13.1 code was found that generated incorrect values for $\Delta\mathcal{G}$. The model setup for XFEM-PN low-cycle fatigue analysis used a similar process to the VCCT analysis in Section 4.5. The model setup, convergence analysis, and a comparison of results to the previous analyses are presented below.

4.6.1 Methods

Two parts were needed to define a model with an initial crack: (1) a two-dimensional half model of the plate, and (2) a two-dimensional wire representing the initial crack. The plate had the same section and boundary conditions as defined in the FEQP and VCCT analyses. Traction separation and damage evolution parameters, discussed later, were added to the material definition along with the elastic properties used in the previous models.

A focused meshing strategy similar to that used in the VCCT model was developed here with the same 12:1 element transition. The model was assembled and meshed such that the initial crack was centered on several elements within the enrichment region. Placing initial cracks on element boundaries was avoided as it tended to give varying and incorrect results which was attributed to a misrepresentation of the initial crack. Two-dimensional plane stress, linear quadrilateral elements (CPS4) were selected since it is the only quadrilateral element supported in an XFEM-PN low-cycle fatigue analysis [5]. Linear and quadratic triangular elements were avoided since they produced fluctuation in$\Delta\mathcal{G}$ values and the triangular elements made the structure too stiff. Partitioning and meshing results are show in Figs. 4.18 and 4.19, respectively.

The main difference between VCCT and XFEM-PN analyses was the representation of the crack. XFEM-PN cracks are placed in an enrichment region where elements have coincident phantom nodes and phantom degrees of freedom, similar to the process development by Song, Areias, and Belytschko [4] discussed in Section 2.2.2. Once fatigue or fracture requirements were met in an element, Abaqus fractures the element, releasing the real node from its coincident phantom node. The two nodes are then free move independently of each other.

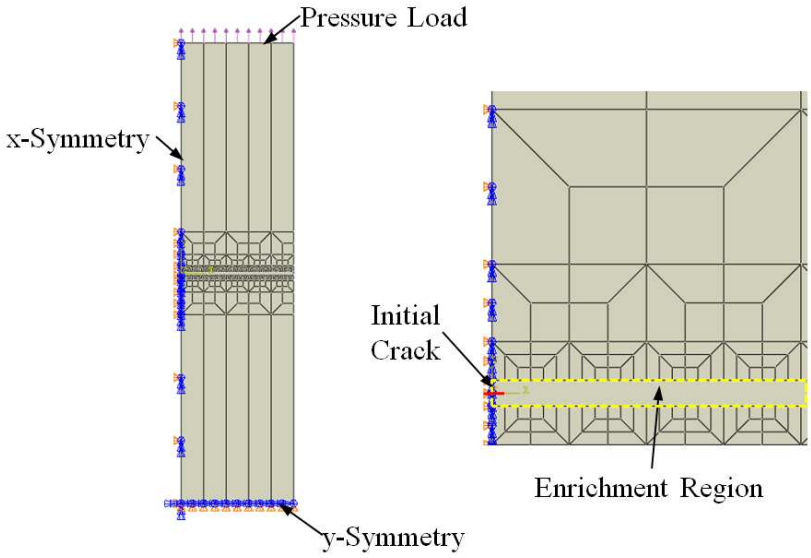

Fig. 4.18: Partitioning, loading, and boundary conditions used in the XFEM model with an inset zoomed in at the initial crack.

XFEM-PN assumes that the stress singularity at the crack tip is not needed in the analysis and defines failure on $\mathcal{G}_c$ criterion [5].

A Level Set Method (LSM), similar to the one proposed by Osher and Sethian [56], is used to locate the discontinuity in the displacement field of fractured elements. Abaqus' representation uses two nearly orthogonal signed distance functions, ϕ and ψ, to locate the crack. The intersection of the distance to the crack surface, ϕ, and the distance to the initial crack tip, ψ, defines the crack surface [5]. Only one crack tip is allowed in an element, and the crack cannot bifurcate or coalesce. Once propagation begins, Abaqus assumes that the crack will completely fracture the element; therefore, only ϕ values need to be stored. Both ψ and ϕ are returned as nodal values in the ODB file for the initial cycle associated with cracking the element, and only ϕ for all subsequent cycles.

Two steps were used for the XFEM-PN low-cycle fatigue analysis in this study: (1) the static general step loaded the plate up to σ_{max}, and (2) the cyclic load was applied in the direct cyclic step to propagate the crack. Similar loading profiles to Fig. 4.13 where used. The direct cyclic solver uses the truncated Fourier series discussed in Section 4.5.1 using Eqs. (4.15) and (4.16) with control parameters I_0, I_R, and I_A adjusted as in the VCCT analysis.

Processing time and solution accuracy were used as metrics in a similar tradeoff study to determine the number of Fourier terms, values for stabilization ratios, and time increment size needed for a stabilized cycle. Abaqus' XFEM-PN low-cycle fatigue analysis allows for parallel computations, thus dramatically reducing processing time compared to VCCT which solves on a single processor. Each iteration solved quickly with an increment size of 0.05 and within the default ratio tolerance of 0.005 for CR_0, CR_n, CU_0, and CU_n. Fourier terms were limited to 12; however, never more than 2 were needed in all XFEM-PN analyses.

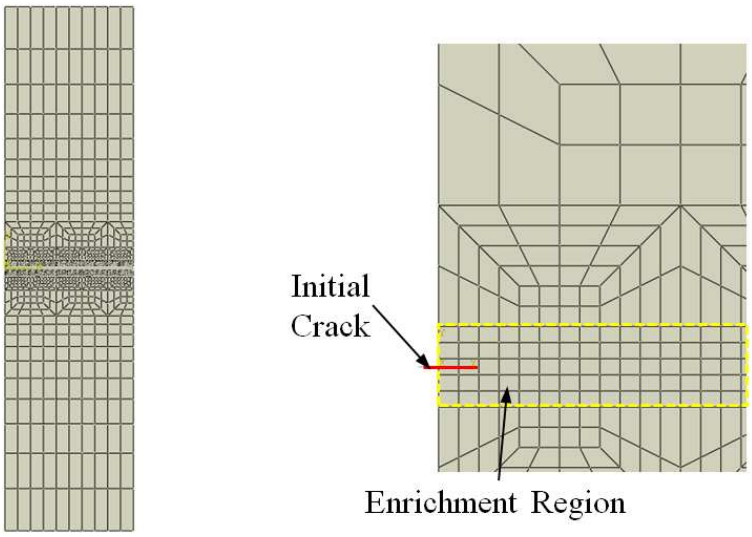

Fig. 4.19: Coarse meshing scheme for the XFEM model with an inset zoomed in at the initial crack.

Similar fracture and fatigue criterion were used for the constants of Eqs. (4.17), (4.18), and (4.19), and bounding ratios $\mathcal{G}_{th}/\mathcal{G}_c$ and $\mathcal{G}_{pl}/\mathcal{G}_c$. Traction separation and damage evolution are required by Abaqus for an XFEM-PN analysis; however, these parameters are primarily used for static load applications, so they did not affect the fatigue crack growth analysis. For completeness, the maximum principal stress criterion was selected using the ultimate stress, $\sigma_{ult} = 83.2\,\text{ksi}$ (574 MPa). Damage evolution was selected to be a mode-independent fracture energy equal to $\mathcal{G}_{Ic}$.

Energy release rate ranges were calculated based on a *modified* VCCT method similar to the method described above; however, nodal forces and displacements are not available at the crack tip since the crack tip is internal to the element. Abaqus' methods for calculating the values needed for Eq. (4.21) are proprietary information, not available to the public. A similar method was mentioned in the paper by Shi et al. [58] where the authors developed a user defined subroutine for Abaqus, but no insight was provided on the *modified* VCCT. Once $\Delta\mathcal{G}$ was known, cycles were calculated from Eq. (4.19), and the crack was propagated entirely through the element ahead of the crack tip. This was continued until $\mathcal{G}_{pl}/\mathcal{G}_c$ was reached. Python code was run in parallel to terminate the analysis should cycles increment by one, the limit case for critical crack growth.

Abaqus calculates CPD based one on three criteria: MTS, normal to the element 1-direction, or normal to the element 2-direction. The latter two methods grow the crack in a mesh dependent fashion, which removes the need for crack extension models. For comparison to the FEQP CPD analysis, only the MTS criterion was selected.

All parameters were entered manually into the INP file since Abaqus CAE does not directly support an XFEM-PN low-cycle fatigue analysis.

4.6.2 Convergence Analysis

The stress intensity ranges and cycles to failure were used as convergence criteria with $R = 0$, similar to the VCCT analysis. Table 4.13 shows that the RMSE, computed relative to the analytical method with Eq. (4.9), holds relatively constant at approximately 4.5% for all meshes. Fig. 4.20 shows that the percent error in ΔK_I at the nodes remains relatively constant throughout crack growth in Meshes #2 through #4. The fluctuations in error seen in Mesh #1 are attributed to the coarse meshing at the borders of the enriched region. RMSD in ΔK_I, calculated with Eq. (4.23), is 0.2% between Meshes #2 and #3. While the RMSE is larger than the error reported in Tada's Eq. (3.3) for F, proceeding to Mesh #3 does not provided significantly better results. For these reasons, Mesh #2 was considered converged for the first convergence criteria.

Table 4.13: XFEM-PN Stress Intensity Range Convergence Analysis.

Mesh	Elements Along Propagation Plane	ΔK_I % Error		ΔK_I % Difference	
		Max.	RMSE	Max.	RMSD
1	120	-6.4	4.4	-	-
2	240	-4.6	4.5	2.0	0.4
3	480	-4.8	4.7	-0.3	0.2
4	960	-5.0	4.9	-0.3	0.3

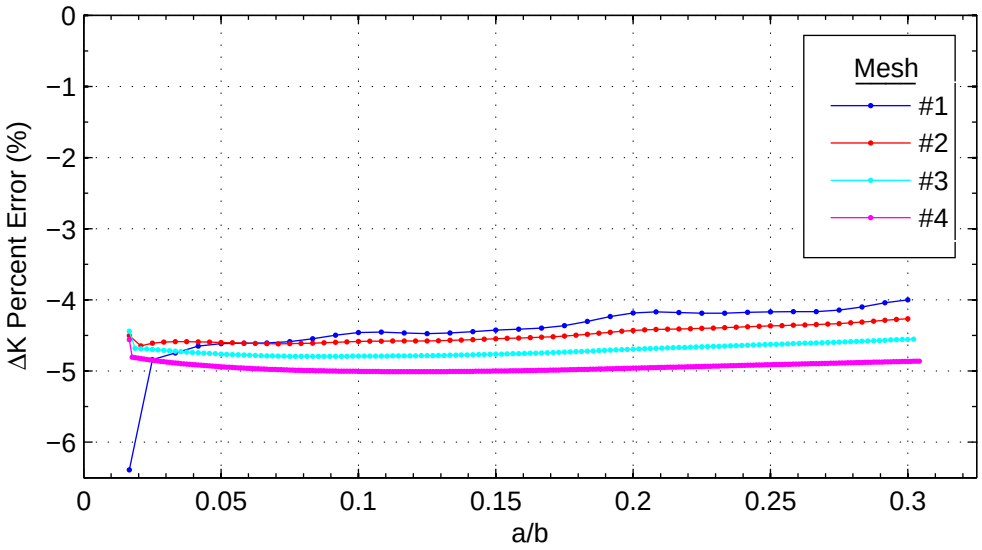

Fig. 4.20: Error in stress intensity range, ΔK_I, during crack propagation in XFEM-PN analysis.

The second convergence criterion was the estimated N_f calculated by Abaqus and the integrated polynomial method. Results are listed in Table 4.14. The integrated polynomial approach has differences less than 1% between the second and third meshes. Errors for the

integrated polynomial method are similar for each mesh, with Mesh #2 21% higher than the analytical result. Fig. 4.21 shows that Abaqus' results are converging towards the results from integrated polynomial method.

Table 4.14: XFEM-PN Estimated Fatigue Fracture Life Convergence Analysis.

Mesh	Elements Along Propagation Plane	Abaqus, N_f		Integrated Polynomial, N_f		Processing Time (hrs)
		% Error	% Diff.	% Error	% Diff.	
1	120	60	-	23	-	0.25
2	240	37	-15	21	-2.0	1.0
3	480	29	-5.8	22	0.5	5.0
4	960	27	-1.6	22	0.7	37

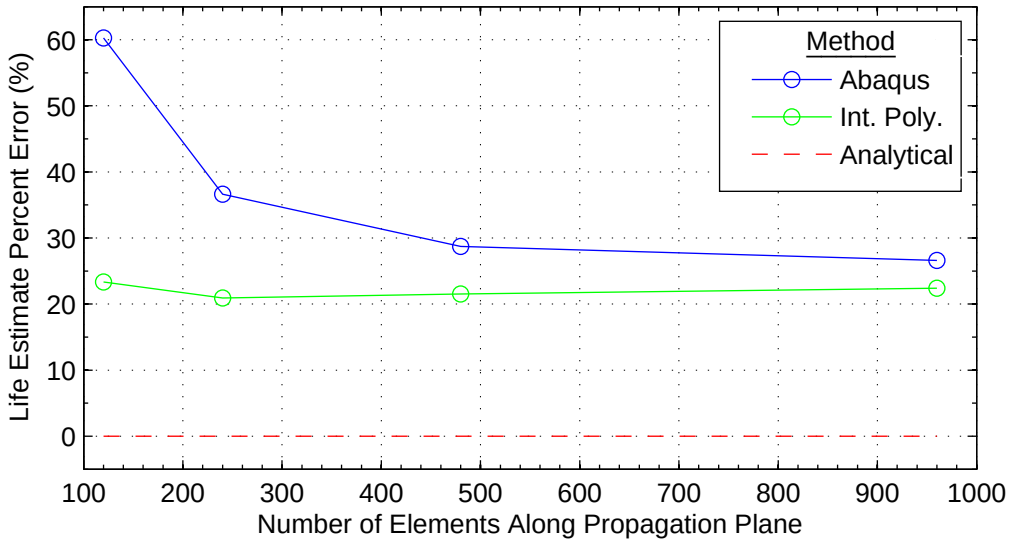

Fig. 4.21: XFEM-PN Convergence plot of estimated fatigue fracture life.

Mesh #2 was considered converged by both convergence criteria since progressing to a more refined mesh showed less than a 1% change in results. Processing time was significantly lower compared to the VCCT analysis due to parallel computing.

4.6.3 Results

Results from the XFEM-PN analysis of the four loading conditions are listed in Table 4.15. Critical crack lengths are 4-7% longer than those calculated with the analytical method. The longer crack length is a result of low SIFs values. Fig. 4.22 shows the normalized ΔK_I for $R = 0.2$ which has an RMSE of 4.4% and a maximum error of -4.6% at the initial crack length relative to the analytical equations. Low SIFs increased the estimated cycles to failure with errors of approximately 20% for each R-ratio using the integrated polynomial

method. Fig. 4.23 shows that XFEM-PN has the largest estimate of fatigue fracture life of the analyses performed in this thesis with a RMSE of 20% and a maximum error of 22% for the first growth cycle.

Table 4.15: Results from the XFEM-PN Fatigue Fracture Analysis

Loading Condition	R	Critical Crack Length, a_c, in. (mm)			Fatigue Fracture Life, N_f, cycles		
		Experimental	Analytical	XFEM-PN	Experimental	Analytical	XFEM-PN
1	0	0.8 (20.3)	1.68 (42.7)	1.80 (45.7)	3 050	2 280	2 750
2	0.2	1.4 (35.6)	2.24 (56.9)	2.38 (60.3)	8 420	7 540	9 090
3	0.5	1.4 (35.6)	2.99 (75.9)	3.13 (79.4)	42 500	49 600	59 680
4	0.67	1.8 (45.7)	3.34 (84.8)	3.48 (88.3)	154 000	179 200	215 600

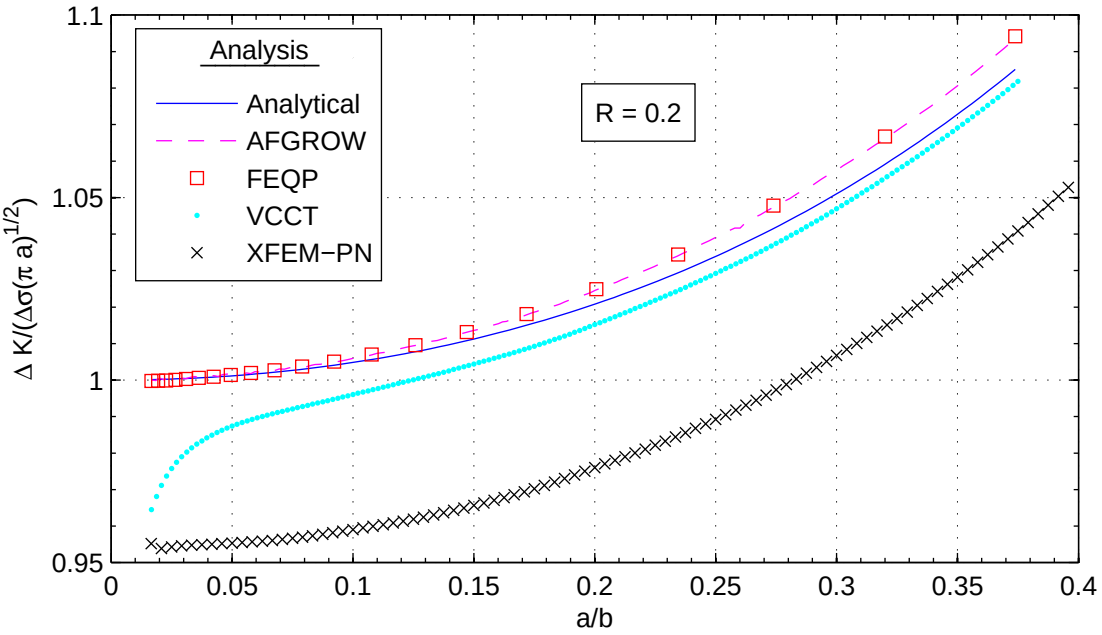

Fig. 4.22: Comparison of the normalized ΔK_I during crack propagation for the analytical and XFEM-PN methods ($R = 0.2$).

A CPD analysis showed that the MTS method predicted $\theta = 0$ for all crack lengths, as seen in Fig. 4.24 and as expected with pure Mode I fracture. Abaqus does not return a value for CPD, so it was post-processed with Eq. (4.24), where x and y coordinates are based on ϕ level set values, i is the current fractured element, and $i + 1$ is the fractured element in the next cycle. Similar CPD results were found for the other loading conditions are provided in Appendix D.

$$\theta = \tan^{-1} \frac{y_i - y_{i+1}}{x_i - x_{i+1}} \tag{4.24}$$

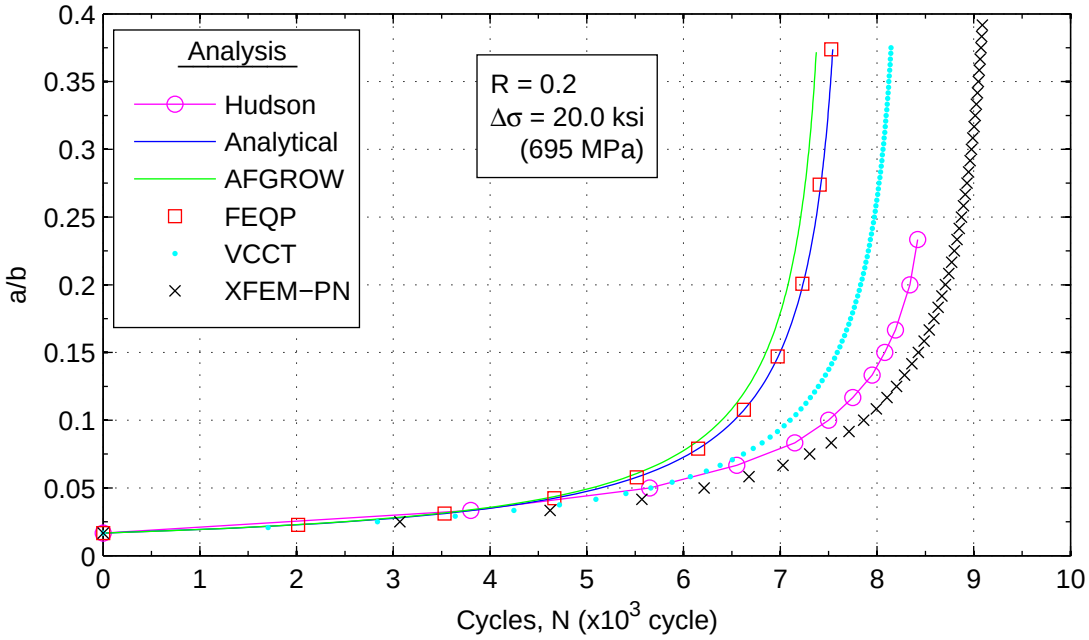

Fig. 4.23: Comparison of cycle count during crack propagation for the analytical and XFEM-PN methods ($R = 0.2$).

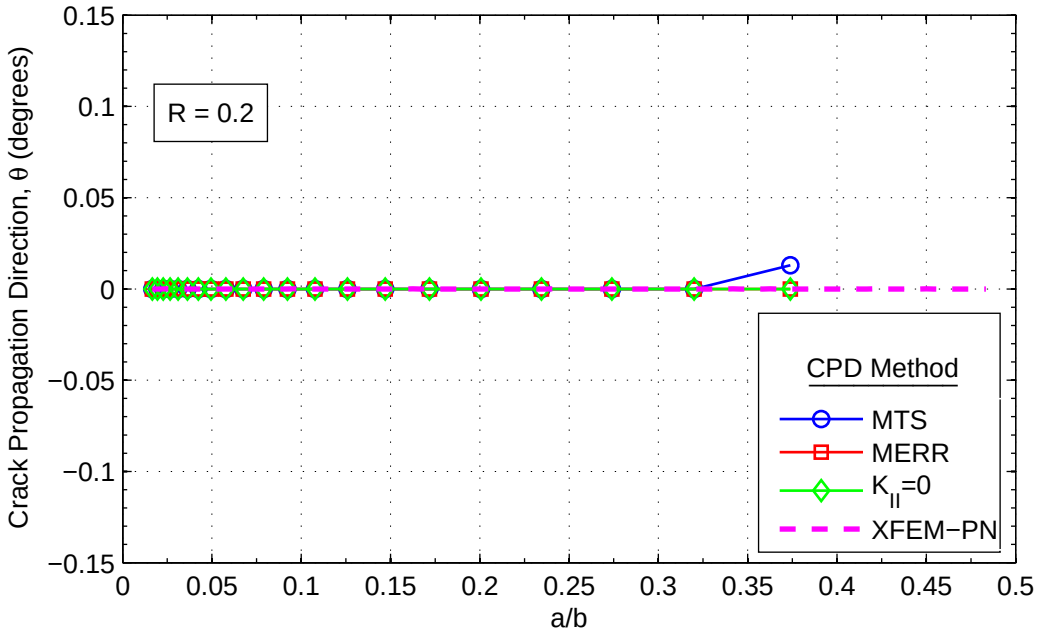

Fig. 4.24: Comparison of crack propagation direction from FEQP and XFEM-PN analyses ($R = 0.2$).

4.7 Conclusion

Five methods were used to estimate the fatigue crack growth in a center crack plate: (1) analytical equations, (2) AFGROW, (3) FEQP, (4) VCCT, and (5) XFEM-PN. Stress intensity ranges from a_i to a_c for all methods compared well to each other with the largest discrepancy seen at short crack lengths for the VCCT, and a constant offset of -5% throughout crack growth for XFEM-PN. Critical crack lengths for all of the analyses are summarized in Table 4.16. XFEM-PN had the longest critical crack lengths that were 4-7% larger than the analytical results, which was attributed to under estimating the SIFs.

Table 4.16: Comparison of Critical Crack Lengths

Loading Condition	R	Critical Crack length, a_c, in. (mm)											
		Experimental		Analytical		AFGROW		FEQP		VCCT		XFEM-PN	
1	0	0.8	(20.3)	1.68	(42.7)	1.69	(42.7)	1.68	(42.7)	1.69	(42.9)	1.80	(45.7)
2	0.2	1.4	(35.6)	2.24	(56.9)	2.23	(56.9)	2.24	(56.9)	2.25	(57.2)	2.38	(60.3)
3	0.5	1.4	(35.6)	2.99	(75.9)	3.00	(76.3)	2.99	(75.9)	3.00	(76.3)	3.13	(79.4)
4	0.67	1.8	(45.7)	3.34	(84.8)	3.33	(85.1)	3.34	(84.8)	3.35	(85.1)	3.48	(88.3)

A crack propagation direction analysis showed good agreement between MTS, MERR, and $K_{II} = 0$ in the FEQP analysis and MTS in the XFEM-PN analysis. Each analysis indicated pure Mode I growth with $\theta = 0$ as seen in Fig. 4.24. θ's that did deviate from zero where considered numerical noise and were ignored.

One of the main goals of this thesis is to estimate fatigue fracture life. Table 4.17 summarizes the results from each of the analyses. Resulting cycles to failure from the analytical, AFGROW, and FEQP analyses compared well to each other. AFGROW and the FEQP analyses had errors of 0.5% and 2.5%, respectively, relative to the analytical results. An integrated polynomial method for calculating life based on the results from Abaqus was introduced to aid convergence of the VCCT and XFEM-PN models. Compared to the analytical method, VCCT and XFEM-PN overestimated life by 8% and 20%, respectively, which is a result of the under estimating the SIFs.

None of the methods compare well with the experimental results from Hudson's report. Several factors are not accounted for with the Walker equation; however, discrepancies in cycles to failure are most likely a result of only having two tests run at each loading condition. Large variability in growth rates is common as described in [63], and many more tests would need to be run to completely quantify the variation associated with the fracture variables studied.

Methods (1)-(4) offer a good baseline for comparison as they have been fully developed in the literature. Analytical equations do not offer much insight into how the crack interacts with the geometry and boundary conditions as the crack propagates through the material. AFGROW allows for a quick analysis and check of classical problems as well as some custom problems. Numerical solutions such as the finite element method are required for complex structures, load transfer and other complex loading and boundary conditions.

Table 4.17: Comparison of Life Estimates

Loading Condition	R	Fatigue Fracture Life, N_f, cycles					
		Experimental	Analytical	AFGROW	FEQP	VCCT	XFEM
1	0	3 050	2 280	2 220	2 270	2 460	2 750
2	0.2	8 420	7 540	7 370	7 530	8 150	9 090
3	0.5	42 500	49 600	48 510	49 490	53 630	59 680
4	0.67	154 000	179 200	175 300	178 800	193 900	215 600

FEQP is the most accurate of the finite element methods explored here relative to the analytical solution studied. The FEQP method can be extended to the generalized 2-D and 3-D problems with good expected results. However, cracks must be grown manually and remeshed at each increment. This requires either significant user interaction and time, or an external program which may have to be customized for each problem.

The significant user interaction required with manually or scripted remeshing for each increment associated with crack propagation led to a VCCT low-cycle fatigue analysis which propagated a crack along a predefined path without remeshing. Predefined paths are acceptable, but in complicated geometry and loading the path is often unknown.

Finally, the XFEM-PN low-cycle fatigue analysis was examined. Using the XFEM-PN allows cracks to grow along a solution dependent path, independent of the mesh given it is sufficiently refined. The XFEM-PN gives engineers the resource to evaluate different crack locations and orientations with minimal remodeling. XFEM-PN will be used in the next analysis which looks at a Monte Carlo simulation to bound the life estimate.

Chapter 5

Monte Carlo Analysis

Analyzing the variability in fatigue fracture life of engineering structures can be difficult when the structures include complex geometry and complex loading and boundary conditions combined with fatigue crack growth models fit to experimental data. Each of these aspects, including many others, has a statistical distribution, and investigating how these affect the structure's life in the presence of a crack may lead to safer, more robust designs. If no known analytical solution exists for combining these components, one method to analyze the distribution on life is the Monte Carlo method. In a Monte Carlo simulation, constants such as loading magnitude and direction, geometric dimensions, or material properties are replaced by Probability Distribution Functions (PDF). If the PDF is developed from a normal distribution, the PDF is defined by two parameters, the mean, μ, and standard deviation, σ, of the sample. Random values are selected from each PDF and the analysis is run as many times as needed to satisfy the law of large numbers and central limit theorem [64] mentioned in Section 2.3.

The analysis of a Center Cracked Plate (CCP) has a known solution so it stands to be a good validation tool between the analytical equations and the eXtended Finite Element Method with Phantom Nodes (XFEM-PN) coupled with a low-cycle fatigue analysis in Abaqus 6-13.3. To limit the scope of the Monte Carlo analysis, distributions were only used for the Walker equation coefficients, C_0, m, and γ fit in Section 3.2.2. Additionally, due to the number of runs required to get a representative distribution on the fatigue fracture life estimate, N_f, only the $R = 0.2$ loading condition was analyzed. The following sections describe the random numbers selection process, the analytical and XFEM-PN Monte Carlo simulations, and results.

5.1 Random Numbers Selection

Annis concluded in [65] that growth constants cannot be independently selected at random since they are jointly distributed based on the linear regression fit. Three constants are present in the Walker equation and each is assumed to be normally distributed about their

mean in log-log space where the coefficients were fit. n-dimensional, jointly distributed PDFs are described by a multivariate normal distribution presented in Eq. (5.1), where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the non-singular symmetric covariance matrix describing the interaction between constants. Several approaches exist for selecting multivariate normally distributed values. One way is to select x_1 based on a normal distribution of μ_1 and σ_1, then find x_2 conditionally on x_1, μ_2, and σ_2, and so on until x_n is found conditionally on x_1, x_2, ..., x_{n-1}, μ_n, and σ_n [76]. MATLAB has a standard function for generating random numbers based on a multivariate normal distributions called `mvnrnd()` which was used in this analysis.

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{n/2}\sqrt{|\boldsymbol{\Sigma}|}} \exp\left(-\frac{\left[(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right]}{2}\right) \tag{5.1}$$

Eq. (5.2) was transformed into log-log space as shown in Eq. (5.3), which was fit with a least-squares multiple linear regression in MATLAB using the form of Eq. (5.4).

$$\frac{da}{dN} = \frac{C_0}{(1-R)^{m(1-\gamma)}} (\Delta K)^m \tag{5.2}$$

$$\log_{10}\left(\frac{da}{dN}\right) = \log_{10}(C_0) + m\log_{10}(\Delta K) - m(1-\gamma)\log_{10}(1-R) + \varepsilon \tag{5.3}$$

$$y = b + m_1 x_1 + m_2 x_2 + \varepsilon \tag{5.4}$$

$$\begin{aligned} \text{where:} \quad y &= \log_{10}\left(\tfrac{da}{dN}\right) \\ b &= \log_{10}(C_0) \\ m_1 &= m \\ m_2 &= -m(1-\gamma) \\ x_1 &= \log_{10}(\Delta K) \\ x_2 &= \log_{10}(1-R) \\ \varepsilon &= \text{random error} \end{aligned}$$

The resulting mean vector and covariance matrix are shown in Eq. (5.5) in $\log_{10}$ units. If all off diagonal values of $\boldsymbol{\Sigma}$ were zero, it would imply no interrelation between the variances on the diagonal; however, the converse is not necessarily true [76]. To determine interdependency, the correlation coefficient matrix, $\boldsymbol{\rho}$, was extracted, where a value of ± 1 indicates a direct correlation and zero no correlation. Positive values imply both variances change in a similar direction, while negative values imply variances change in inverse directions. It is obvious from Eq. (5.5c) that all coefficients in the correlation matrix are correlated since the off-diagonal terms are significant relative to 1.0 and must be modeled as such, confirming Annis' previous conclusion.

$$\boldsymbol{\mu} = \begin{bmatrix} b \\ m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} -9.064 \\ 3.839 \\ -1.674 \end{bmatrix} \tag{5.5a}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_b^2 & \rho_{12}\sigma_b\sigma_{m_1} & \rho_{13}\sigma_b\sigma_{m_2} \\ \rho_{12}\sigma_b\sigma_{m_1} & \sigma_{m_1}^2 & \rho_{23}\sigma_{m_1}\sigma_{m_2} \\ \rho_{13}\sigma_b\sigma_{m_2} & \rho_{23}\sigma_{m_1}\sigma_{m_2} & \sigma_{m_2}^2 \end{bmatrix} = \begin{bmatrix} 5.680 & -3.777 & 3.916 \\ -3.777 & 2.625 & -2.365 \\ 3.916 & -2.365 & 3.812 \end{bmatrix} \times 10^{-3} \tag{5.5b}$$

$$\boldsymbol{\rho} = \begin{bmatrix} \rho_{11} & \rho_{12} & \rho_{13} \\ \rho_{12} & \rho_{22} & \rho_{23} \\ \rho_{13} & \rho_{23} & \rho_{33} \end{bmatrix} = \begin{bmatrix} 1 & -0.978 & 0.842 \\ -0.978 & 1 & -0.748 \\ 0.842 & -0.748 & 1 \end{bmatrix} \tag{5.5c}$$

Least-squares linear regression assumes a normally distributed random error term. To describe the distribution of the random error term, ε, in Eq. (5.4), a histogram of the $\log_{10}(da/dN)$ is shown in Fig. 5.1 with a normal PDF superimposed. The histogram shows a decent comparison with the normal PDF. To confirm that the $\log_{10}(da/dN)$ and the normal PDF have similar distributions, a quantile-quantile, or Q-Q, plot is presented in Fig. 5.2 comparing the two distributions. The reference line in Fig. 5.2 represents both samples originating from the same distribution. The bulk of the $\log_{10}(da/dN)$ data lie near the reference line. Data at the left and right ends of the plot deviate from the reference line. Points that deviate from the reference line on the left end are attributed to the non-linearity of the crack growth rate approaching ΔK_{th}. Points that deviate from the reference line on the right end are explained by the non-linearity of the crack growth rate when approaching $K_{max} = K_{Ic}$. From the Q-Q plot, it can be concluded that the $\log_{10}(da/dN)$ data, and consequently the random error is normally distributed in log-log space.

In the Monte Carlo analyses, random variables used in this study were selected in log-log space using $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ in Eqs. (5.5a) and (5.5), then converted back to C_0, m, and γ with the inverse of Eq. (5.4) and used to estimate N_f in the analytical and XFEM-PN simulations described next.

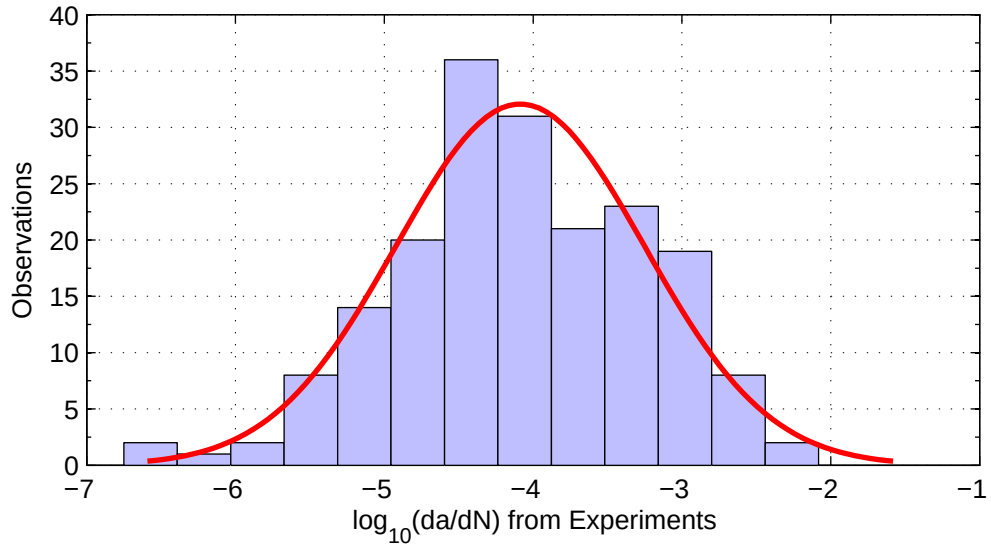

Fig. 5.1: Histogram of $\log_{10}(da/dN)$ from experimental data.

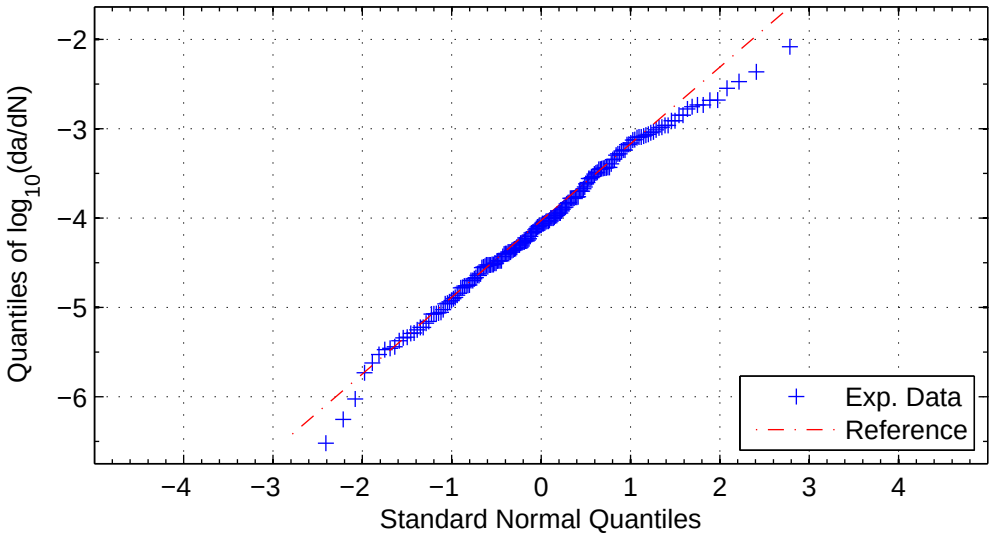

Fig. 5.2: Q-Q plot of $\log_{10}(da/dN)$ from experimental data with a normal PDF.

5.2 Analytical Simulation

The mean vector and covariance matrix were used to generate random numbers for the variables in the Walker equation, which were inserted into Eq. (5.7) for integration from a_i to a_c. A convergence study was performed to determine an adequate sample size for the Monte Carlo simulation where the number of samples was increased in increments of 1000 until the mean and standard deviation on N_f did not change by more than 0.5% between iterations. This process was repeated five times resulting in sample sizes ranging from 2000-7000 samples. A total number of 10,000 samples were selected since the increase in the computational time was minimal.

$$\overline{\Delta K} = \frac{\Delta K}{(1-R)^{1-\gamma}} \tag{5.6}$$

$$N_f = \int_{N_i}^{N_c} dN = \frac{1}{C_0} \int_{a_i}^{a_c} \frac{da}{\left(\overline{\Delta K\,(a)}\right)^m} \tag{5.7}$$

Results from the analytical Monte Carlo simulation are shown in Fig. 5.3 with a mean value of $\mu = 7550\,\text{cycles}$ and a standard deviation of $\sigma = \pm 240\,\text{cycles}$. The coefficient of variation (CV) was used to normalize the standard deviation such that CV $= \sigma/\mu \times 100\%$. A 3.2% variation in estimated fatigue fracture life is excellent. Results from Hudson's report [6] are also shown by the vertical line at the recorded average data point, emphasizing the fact that only two experimental runs at $R = 0.2$ and $\sigma_m = 15.0\,\text{ksi}$ is not an adequate sample size to describe the system. A normal PDF using μ and σ fits very well to the analysis which was confirmed by the Q-Q plot in Fig. 5.4. Distributions of the 10,000 samples of C_0, m, and γ are found in Appendix D.

For a comparison on the correlation effects of the covariance matrix, Fig. 5.5 illustrates what happens to the standard deviation when no correlation is assumed between the Walker constants. The CV for the uncorrelated analysis increased to 23%, significantly larger than the correlated simulation above, thus highlighting the importance of correctly selecting the random variables in a Monte Carlo simulation. Additionally, the distribution no longer appears to be described by a normal PDF, and may more closely be compared to a log-normal PDF.

Crack length and cycle count are plotted in Fig. 5.6 where cycles were calculated from the 10,000 sampled Walker coefficients. Life was calculated at several crack increments by integrating Eq. (5.7) from a_i to the incremental crack length. Mean life and standard deviation were determined at each crack incremented based on a normal distribution. Recorded data from Hudson's report and XFEM-PN results from Section 4.6 were included. Neither the experimental data nor XFEM-PN results fall within three standard deviations of the mean analytical values. This figure further demonstrates that the two experiments from Hudson's report are not adequate to realize the variability in crack growth. The overestimated XFEM-PN fatigue cycles are a result of the underestimated energy release rates as described in Section 4.6.

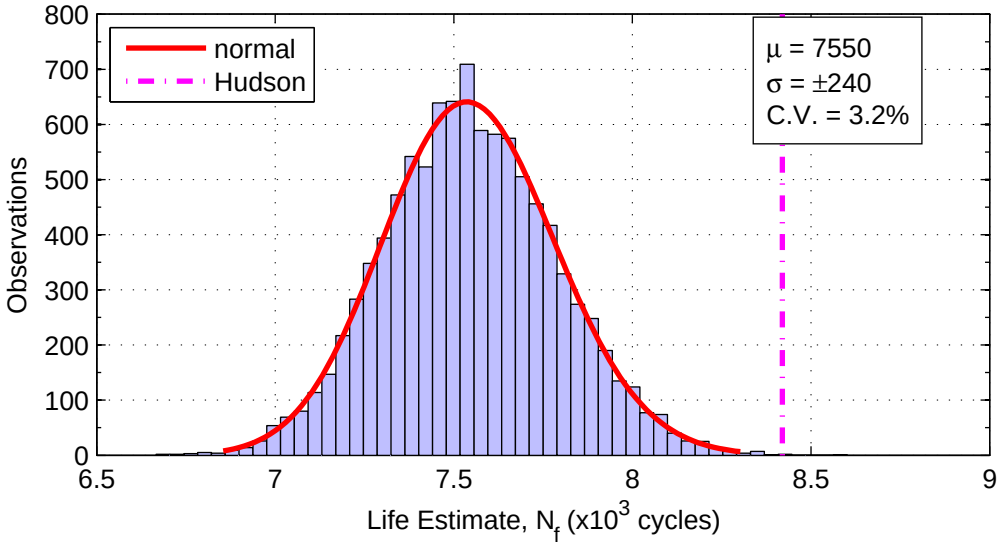

Fig. 5.3: Monte Carlo simulation of fatigue fracture life with 10,000 samples ($R = 0.2$).

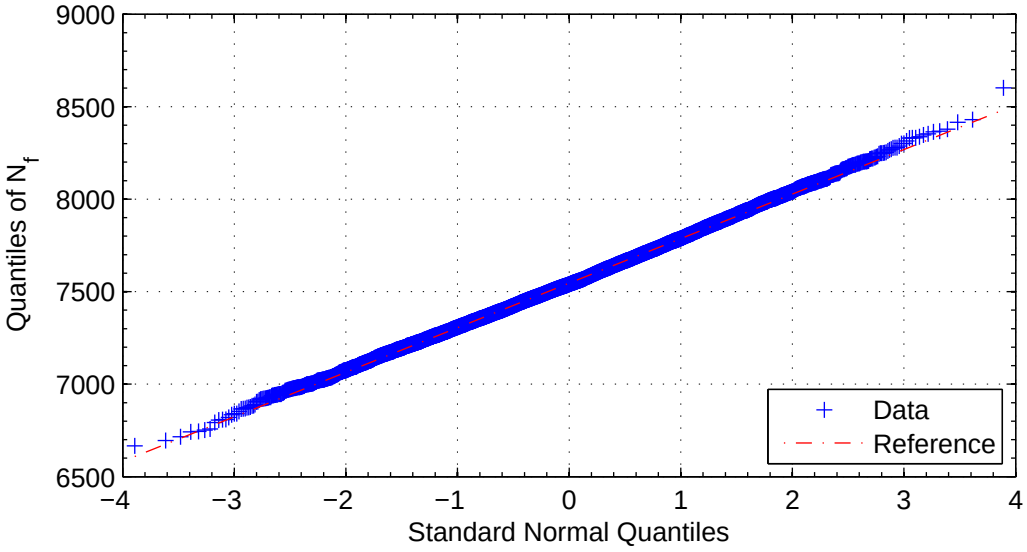

Fig. 5.4: Q-Q plot of the distribution of Monte Carlo simulation compared to a normal PDF.

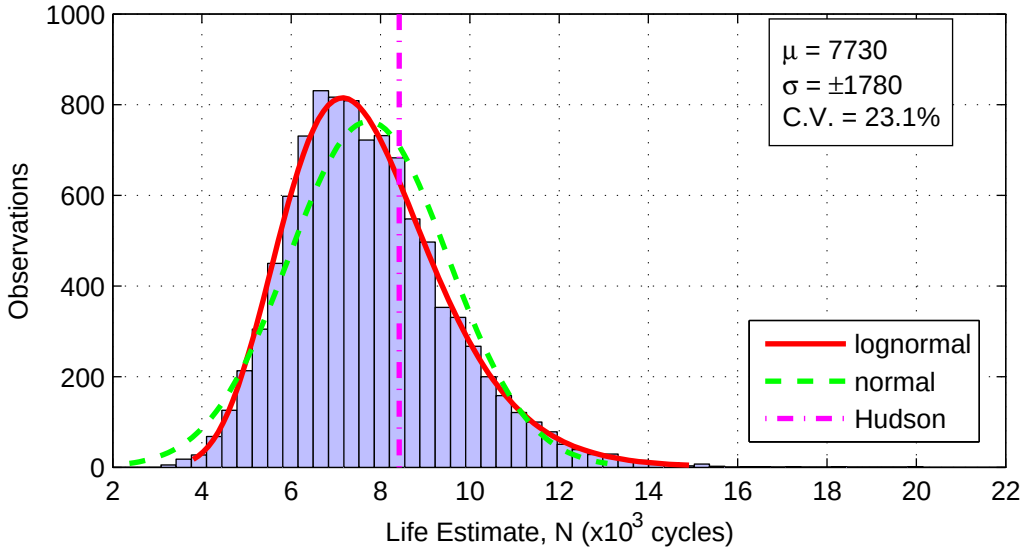

Fig. 5.5: Monte Carlo simulation of fatigue fracture life with 10,000 samples with no correlation between Walker constants ($R = 0.2$).

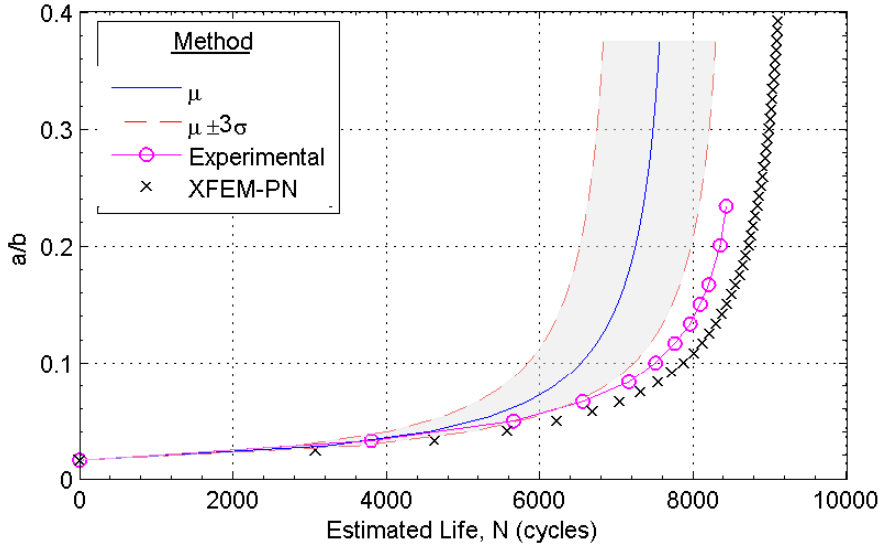

Fig. 5.6: Monte Carlo simulation of fatigue crack growth with 10,000 samples ($R = 0.2$)

5.3 XFEM-PN Simulation

A similar method was followed for the XFEM-PN Monte Carlo simulation, except running 10,000 samples was not feasible with each finite element analysis requiring approximately 1.0-1.5 hours to complete. A total of 500 runs were completed, each one with a randomly selected, correlated C_0, m, and γ with the methods described in Section 5.1. Values were converted to c_3 and c_4 with Eq. (5.8) and inserted into a standard Abaqus INP file. Python code was developed to handle the multiple runs, minimizing user interaction.

$$c_3 = C_0(1-R)^{\gamma m}\left(\frac{E}{1-R^2}\right)^{m/2} \tag{5.8a}$$

$$c_4 = \frac{m}{2} \tag{5.8b}$$

Results from the XFEM-PN Monte Carlo simulation are shown in Fig. 5.7 with fatigue fracture life estimated with the integrated polynomial method. A mean fatigue fracture life of $\mu = 9090\,\text{cycles}$ was found with CV = 3.3%. Mean life values for XFEM-PN are 20% higher than the analytical estimation as a result of Abaqus' predicted ΔK_I's for the CCP being about 5% too low, as explained in Section 4.6. A Q-Q plot comparing the results from the XFEM-PN simulation with a normal PDF is provided in Fig. 5.8 showing good correlation between the two distributions. More importantly, the normal distribution of the XFEM-PN estimate was similar to the distribution from the analytical method as seen with the Q-Q plot in Fig. 5.9, verifying that XFEM-PN is an acceptable method for analyzing the variability in fatigue fracture life. Probability distributions for the Abaqus coefficients c_3 and c_4 used in this analysis are found in Appendix D.

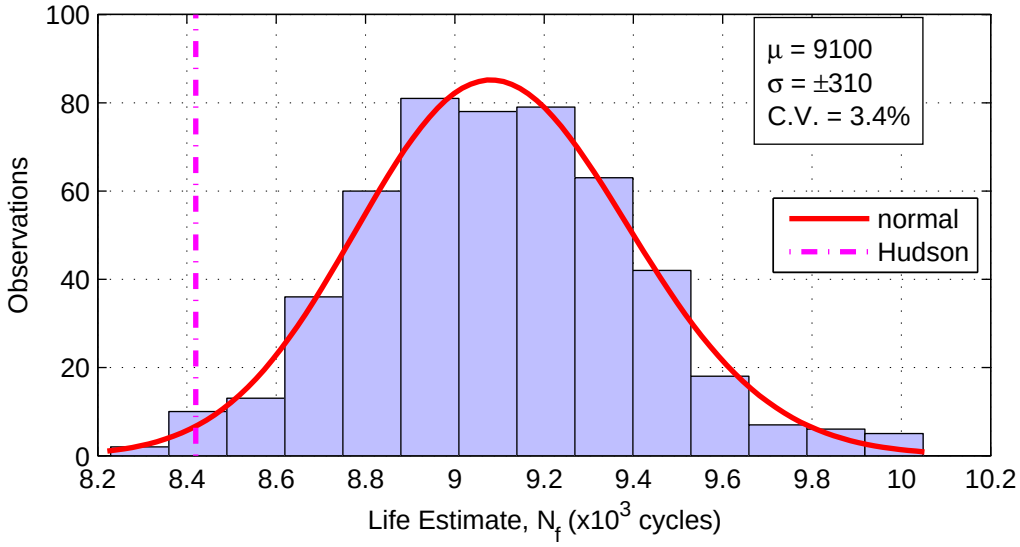

Fig. 5.7: Monte Carlo simulation of fatigue fracture life estimated with XFEM-PN and the integrated polynomial method with 500 samples ($R = 0.2$).

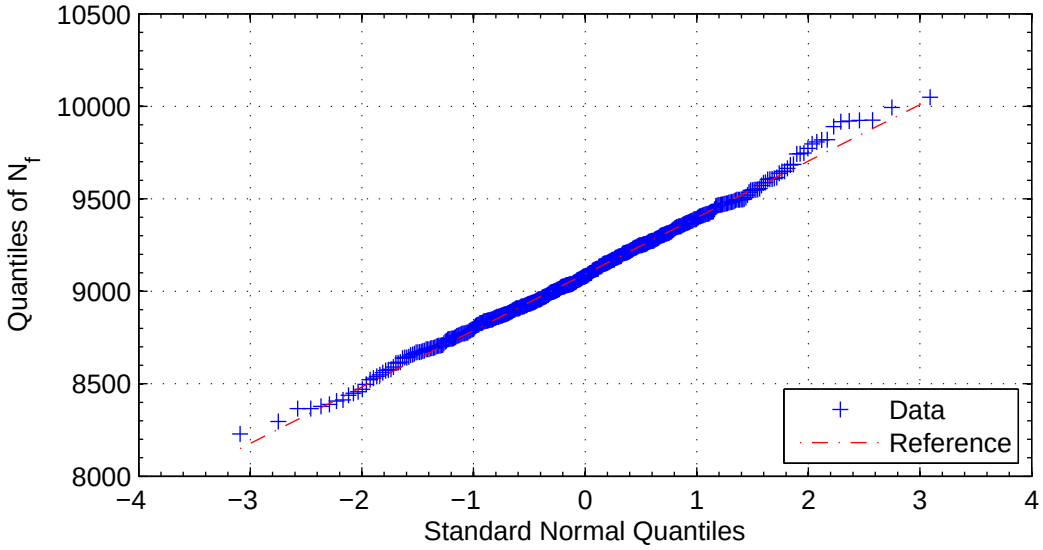

Fig. 5.8: Q-Q plot of the distribution of the XFEM-PN Monte Carlo simulation compared to a normal PDF.

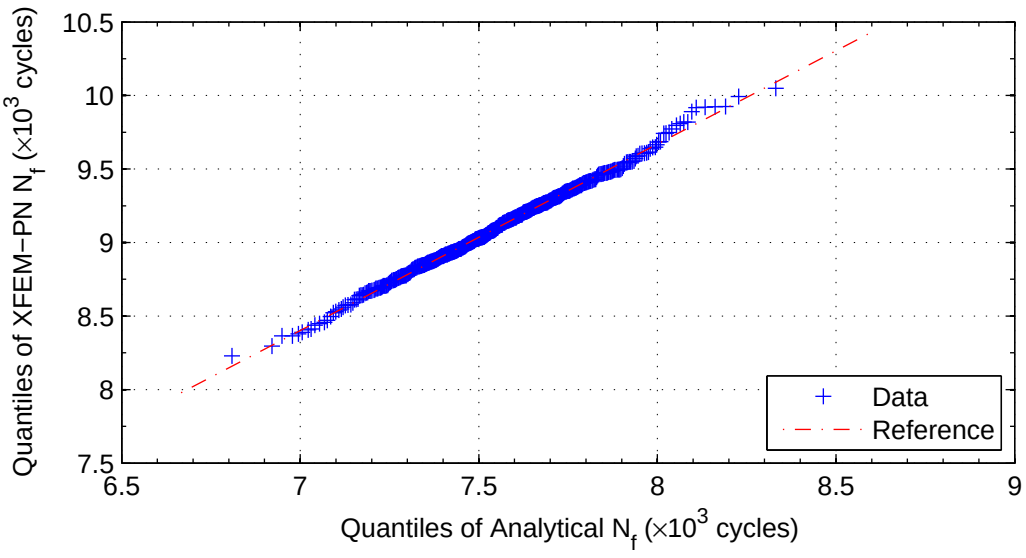

Fig. 5.9: Q-Q plot of the distribution of the XFEM-PN Monte Carlo simulation compared to the distribution of the Analytical Method simulation.

5.4 Conclusion

This chapter looked at two Monte Carlo simulations: the first following the traditional analytical approach allowing for a large number of samples, and a second approach based on crack growth estimates using the Abaqus XFEM-PN low-cycle fatigue solver, the proposed method in this thesis of estimating the variability in fatigue fracture life. It was shown that a multivariate normal distribution was needed to appropriately select random numbers from the distributions for the random variables in the Walker equation. Crack growth constants C_0, m, and γ which were all found to be strongly correlated. Both simulation methods gave similar results, with comparable coefficients of variation of approximately 3%. Distributions of the results appeared to be normal by analyzing Q-Q plots.

An XFEM-PN Monte Carlo simulation has proven to be a viable method for analyzing the variability in fatigue fracture life estimates for a center cracked plate. It is expected that the Abaqus XFEM-PN low-cycle fatigue solver could be used as the basis for extending the life predictions to more complex geometries, loading and boundary conditions.

Chapter 6

Conclusions and Future Work

The final chapter of this thesis summarizes the work performed and conclusions found throughout the analyses. Recommendations are provided for future work in three-dimensional fatigue crack propagation analyses. First, the goals of the thesis are reviewed.

6.1 Research Goals and Objectives

Research goals were stated in Chapter 1 with the primary goal to investigate the capabilities of the eXtended Finite Element Method with Phantom Nodes (XFEM-PN) for modeling crack propagation and estimating fatigue fracture life. Then, use XFEM-PN to quantify the distribution of fatigue fracture life. If this is possible with a simple model, confidence will be gained when using this method with complex models that have not been studied in the literature, and may not have an analytical description. The objectives that were met to realize these goals include:

1. Fit fatigue crack growth rate models to experimental data.
2. Analyze crack growth using methods established in the literature.
3. Analyze crack growth using XFEM-PN and compare results to the experimental data and established methods.
4. Combine the variability in the fitted crack growth rate models with the XFEM-PN to determine the distribution on fatigue fracture life estimates.

6.2 Conclusions

The Walker crack growth model was selected to estimate the fatigue crack growth rates. Unlike the Paris equation, the Walker equation addresses the effects of the R-ratio, thus

the entire data set was used to fit the crack growth model. The Walker equation was also selected because Hudson's data set did not contain an adequate number of samples in the unstable and threshold regions of crack growth to model the Forman equation and Broek and Rice equation. The Walker equation was easily modified for implementation into Abaqus' low-cycle fatigue analysis tool used for fatigue crack growth.

Four models were selected to estimate the fatigue fracture life of the Center Cracked Plate (CCP) for comparison to XFEM-PN: (1) analytical equations, (2) AFGROW, (3) FEQP, and (4) VCCT. First, integration of the Walker equation was performed from an initial crack length to the critical crack length, a_c, computed from the critical stress intensity factor, K_c. The dimensionless geometry factor, F, was found in Tada's handbook [2]. This method provided a solid reference for comparison as these equations have been thoroughly developed in the literature.

AFGROW provided another baseline for comparison. The program, developed by The Air Force Research Laboratory and LexTech, Inc. [34], uses a database of equations to calculate Stress Intensity Factors (SIFs) for numerous crack and specimen geometry. Cracks are incrementally grown until reaching K_c, and life is summed between crack increments. The CCP is a predefined model in AFGROW and required only dimensions, material data, and a load spectrum. Stress intensity factors (SIFs) from the AFGROW analysis had a Root Mean Square Error (RMSE) less than 0.5% relative to the analytical analysis. Difference in the SIFs was attributed to different equations for F. AFGROW's estimate of cycles to failure for the four loading conditions were approximate 2.5% low.

A Linear Elastic Fracture Mechanics (LEFM) Finite Element formulation using Quarter Points (FEQP) was developed in the mid-1970's by Henshell and Shaw [45] and Barsoum [46] as one of the first finite element methods that did not require the use of specialized elements and could adequately model the $1/\sqrt{r}$ stress singularity at the crack tip. The method requires the mid-side nodes of quadratic elements adjacent to the crack tip to be moved to the quarter point location, and the edges adjacent to the crack tip to be collapsed to a single point. The Abaqus finite element code only allows quasi-static analyses with quarter point method, thus the crack must be manually grown and the model remeshed between increments. Dowling's [8] modified Simpson's method was used to reduce the number of models needed to calculate the fatigue fracture life. Twenty models were selected and a Python code was developed to grow the crack, remesh the model, and extract pertinent results from the Abaqus Output Database (ODB) file. The FEQP analyses provided excellent agreement with the analytical analyses. ΔK's throughout propagation had an RMSE of less than 0.5% and fracture life was within 0.5% of the analytical result. A Crack Propagation Direction (CPD) analysis with the three criteria available in Abaqus showed that crack growth would occur perpendicular to the mid-plane of plate in a pure Mode I fashion, as expected.

Rybicki and Kanninen [50] developed the Virtual Crack Closure Technique (VCCT) in the late 1970's based on Irwin's assertion that the energy required to grow a crack is equal to the energy required to close a crack of the same length. VCCT is naturally accommodated in a finite element framework as the force required to hold two nodes together at a crack tip and the displacement behind the crack tip are readily available. This method is easier

to use than FEQP since it does not require a focused meshed around the crack tip and can be easily applied to crack growth models if the crack propagation plane is known. Stress singularities are not modeled in this analysis, and only the energy release rate, $\mathcal{G}$, is needed for crack propagation. SIFs, converted from $\mathcal{G}$ in the ODB file, compared well with the analytical analysis with nodal errors were less than 2% for $a/b \geq 0.03$, but were too low when $a/b < 0.03$. The RMSE for all SIFs was within 1% of the analytical values. The low SIFs at the beginning of crack growth led to overestimating the fatigue fracture life by approximately 8%. The integrated polynomial approach was developed to calculate life based on Abaqus' results extracted from the ODB file. As a result, a less refined mesh was required to estimate life, saving significant computational time.

XFEM-PN was developed by Song, Areias, and Belytschko [4] in 2006 as a method to arbitrarily define a crack, independent of the mesh, given the mesh is sufficiently refined. A Level Set Method (LSM) is used to define the crack location using signed distance functions stored as nodal values. Abaqus combines XFEM-PN, LSM, and a direct cyclic solver propagate fatigue cracks along a solution dependent path [5]. A *modified* VCCT method was used to calculate $\Delta\mathcal{G}$, and a modified Paris equation determines the cycles between crack lengths. SIFs had a RMSE 5% lower than the analytical analysis which led to a 20% overestimation of expected life. The integrated polynomial method was used in this analysis as well. A CPD analysis showed crack propagation perpendicular to the mid-plane of the plate under pure Mode I growth, the same as the FEQP solution.

A Monte Carlo simulation was selected as the method to model variability in the fatigue fracture life estimate. This study focused on random numbers selected from the distributions of the random variables representing the constants C_0, m, and γ of the Walker equation. As suggested by Annis [65] and confirmed in this thesis, the growth constants are highly correlated and cannot be selected independent of each other, thus a multivariate normal PDF was used. Two simulations were performed: (1) an analytical simulation allowed the selection of 10,000 samples to ensure that the law of large numbers and central limit theorem were satisfied, and (2) an XFEM-PN simulation. Both the analytical simulation and XFEM-PN simulations resulted in normal distributions on life and Coefficients of Variation (CV) of approximately 3%.

These results show that for the analysis of a CCP the XFEM-PN provides a viable alternative to estimating the fatigue fracture life while providing an understanding of the variability in that solution. It is expected that the methods used in the study can be extended to other crack propagation problems for estimating life under more complex geometry, loading scenarios and boundary conditions. Applying the methods developed in this thesis has great potential, leading to safer, more robust design decisions if fatigue fracture life can be accounted for earlier in the design phase.

6.3 Future Work

Work performed in this thesis was limited to a center cracked plate made from a 7075-T6 Aluminum alloy in two dimensions because experimental data on crack growth and

an analytic solution for the LEFM analysis existed in various forms to provide a baseline for evaluating finite-element-based crack propagation methods. An obvious first extension would be modeling crack propagation using XFEM-PN in three dimensions. Some XFEM-PN three-dimensional crack growth analyses have already been performed by the author for other research projects, including a compact specimen used in a crack growth test. Material properties used in the model are listed in Table E.1 in Appendix E for standard rail steel. Compact specimen dimensions are provided in Fig. E.1 in Appendix E. The partitioning scheme and coarse structured mesh of a half-model with symmetry about the mid-plane of the specimen are shown in Fig. 6.1.

Compact specimens are loaded by two pins through the top and bottom holes of the specimen. Loading and boundary conditions at the pins were modeled as reference points at the center of the pins. The reference points used kinematic coupling constraints for all active degrees-of-freedom to distribute the load and boundary conditions to surfaces on the compact specimen. The surfaces were defined by the top and bottom 45° of the top and bottom pin holes, respectively. Loading and boundary conditions of each reference point are listed in Table E.2. Contact between the pins and the compact specimen was explored; however, Abaqus 6-13.3 currently contains a programming error where the direct cyclic solver would not stabilize if surface-to-surface contact was present. The programming error resulted in incorrect results, thus analyzing contact was abandoned. Dassault Systèmes Simulia Corporation has been notified and verified the error. The error is to be corrected in a future release.

The finite element model was built with three dimensional, linear continuum elements (C3D8) as these are the only brick element types allowed for an XFEM-PN analysis in Abaqus. The incompatible mode option (C3D8I) was used for the elements outside the enrichment region, since this option adds degrees of freedom for curvature which aids in convergence when bending is present.

Before a convergence analysis was performed, a post-processing routine was needed to determine how to extract and analyze the output data from the model. Three dimensional fatigue crack growth analyses have varying stress intensity factor ranges, ΔK, across the crack front. Fig. 6.2 shows an example of the variation in the normalized ΔK values for four randomly selected cycles from the second mesh refinement. The reduction in ΔK on the right side of the figure is a result of the free surface at $z/b = 0.5$, where z is the through-thickness coordinate and b is the specimen thickness. The mid-plane of the specimen is at $z/b = 0$. For the purposes of this analysis, the median value of the ΔK profile was used to represent the total ΔK used to calculate the crack growth rate, da/dN.

The varying ΔK values also resulted in different crack growth rates across the crack front, and thus non-uniform crack propagation was seen. The non-uniform crack growth was observed as some fraction of the elements adjacent to the crack front releasing since the elements met the criteria for fracture. It often took another solution cycle for remaining elements to release and to become a uniform crack front again. The primary issue with non-uniform crack fronts, is that the sharp change in the crack geometry resulted in energy release rates three to four times higher for elements not fractured compared to the other elements on the crack front. Once the crack front was uniform, the energy release rates for all elements on

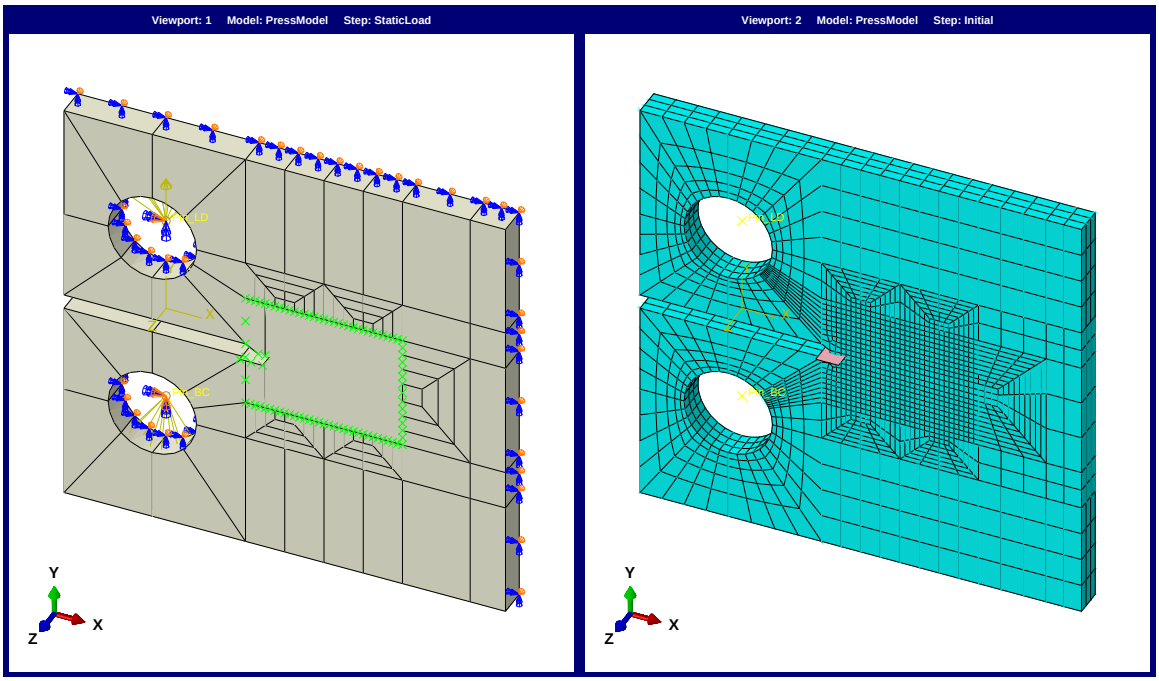

Fig. 6.1: Compact specimen partitioning scheme and coarse mesh.

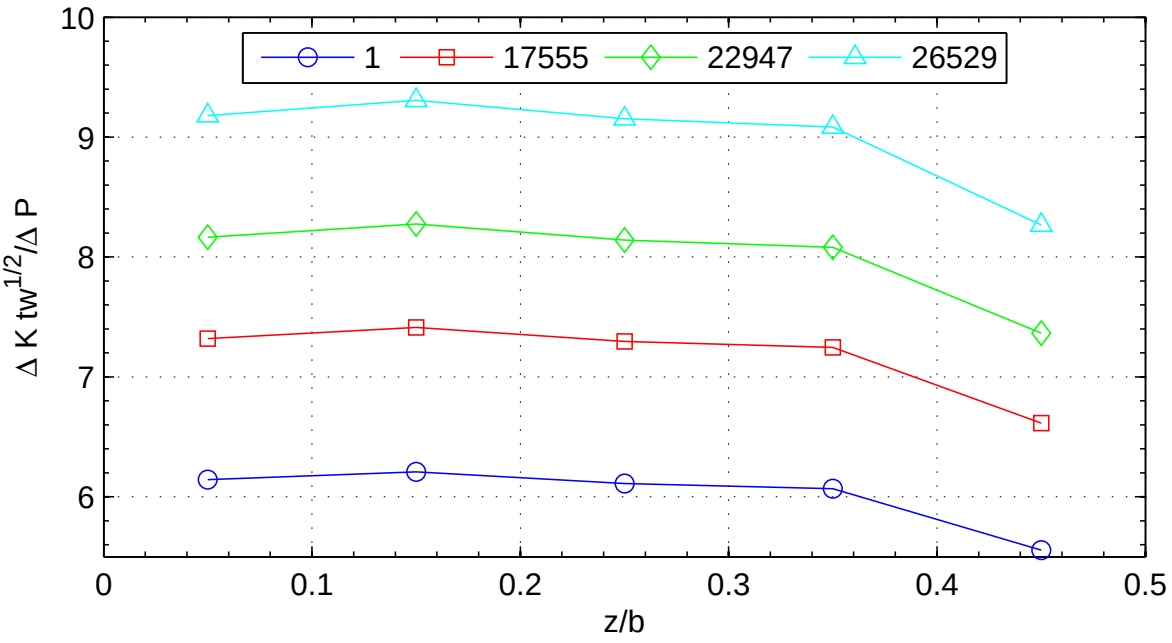

Fig. 6.2: Variation in the normalized ΔK across a compact specimen half model.

the crack front were similar. Throughout this analysis, results from non-uniform crack fronts were discarded.

An XFEM-PN convergence analysis was performed and the results at listed in Table 6.1. Mesh sizes are defined by the number of elements along the crack propagation plane times the through-thickness elements. ΔK values during propagation where compared to analytical results for the convergence criteria. The equation for ΔK is provided in Eq. (6.1) from ASTM Standard E647 [66], where $\alpha = a/w$, a is the crack length, w is the distance from the center line of the pins to back of the specimen, ΔP is the load range, and b is the specimen thickness. The Root Mean Square Difference (RMSD) is 0.11% between Mesh #1 and #2 and both have a Root Mean Square Error (RMSE) of less than 1%. Models were not run to complete fracture due to time constraints which resulted in different final crack lengths for each model. Since the final crack lengths were not the same, the fatigue cycle counts of the models were not comparable to each other, thus it was not used for measuring convergence. Mesh #2 was considered converged for this analysis.

$$\Delta K = \frac{\Delta P}{b\sqrt{w}} \frac{(2+\alpha)}{(1-\alpha)^{3/2}} \left(0.886 + 4.64\alpha - 13.32\alpha^2 + 14.72\alpha^3 - 5.6\alpha^4\right) \tag{6.1}$$

Fig. 6.3 shows the median normalized ΔK values at each crack increment during propagation and the analytical results. The XFEM-PN results compare well with an RMSE of 0.67% and a maximum error of -1.2% occurring at the final crack increment. Fig. 6.4 shows the cycle count during crack growth for the analytical analysis, Abaqus' results from the 3-D compact specimen, and the integrated polynomial method of calculating life from Abaqus' results. The integrated polynomial method has an RMSE of 1.1% and a maximum error of 1.7% occurring at the final crack length, while Abaqus' results have a larger error with an RMSE of 11% and a maximum error of 27% at the final crack increment. The lower RMSE again shows that the integrated polynomial method can be applied to various crack growth analyses. Deformation modes and principal strain fields for the compact specimen are found in Appendix E.

Results from the compact specimen analysis show that XFEM-PN and the methods discussed in this thesis can be applied to models other than the center cracked plate, including models in three dimensions. Some of the issues for future work include optimizing Abaqus' damage tolerance setting to possibly remove the non-uniform crack growth. Also, other methods for analyzing the ΔK through-thickness profile should be explored, including looking at the

Table 6.1: Convergence Analysis for the Compact Specimen

Mesh	Elements On 2D Propagation Plane	ΔK % Error		ΔK % Difference	
		Max.	RMSE	Max.	RMSD
1	96	-1.6	0.92	-	-
2	320	-1.2	0.67	0.21	0.11
3	768	-0.26	0.14	0.20	0.12

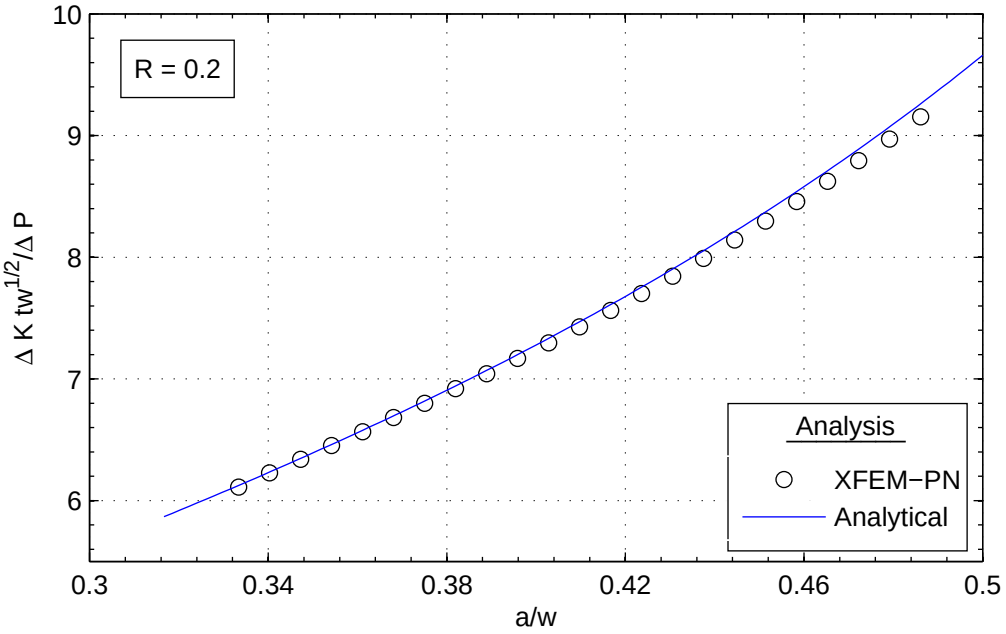

Fig. 6.3: Comparison of the median normalized ΔK_I during crack propagation in a 3-D compact specimen for the analytical and XFEM-PN methods ($R = 0.2$).

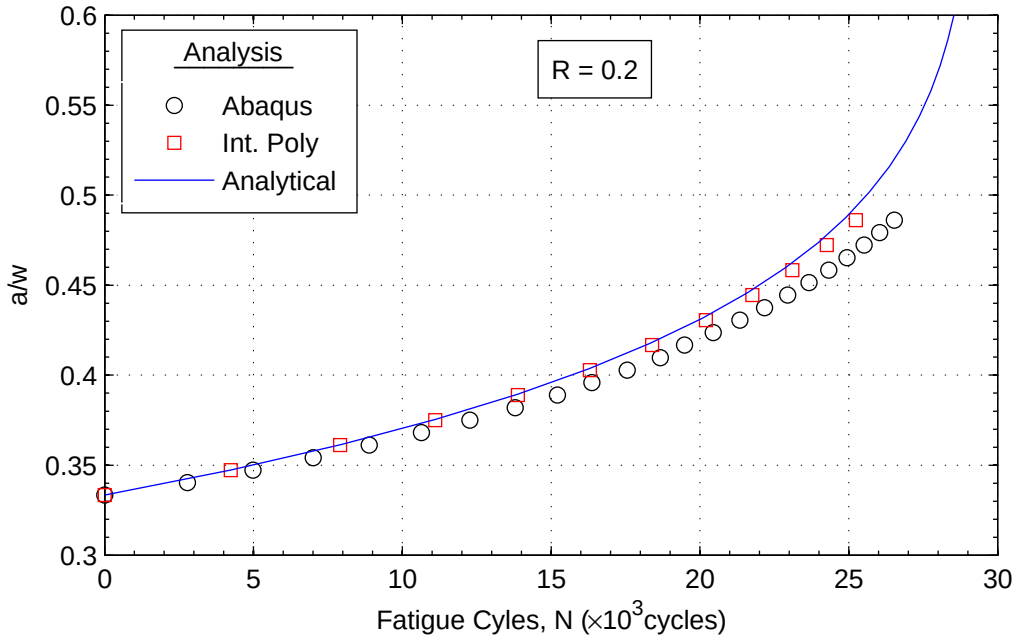

Fig. 6.4: Comparison of cycle count during crack propagation in a 3-D compact specimen for the analytical and XFEM-PN methods ($R = 0.2$).

effects the profile has on crack growth projections along the crack front. These analyses should be coupled with experimental data to determine if the crack growth patterns are similar to those seen in the test specimens.

This thesis used a Monte Carlo simulation to quantify the distribution on fatigue fracture life for the CCP specimen. Extension of the Monte Carlo method into three-dimensional crack propagation may be difficult due to the computational time and file size requirements. For instance, the second mesh refinement had 28 900 nodes and 86 700 degrees-of-freedom and only propagated a crack 0.45 in. (11.4 mm), taking 16 hours on 64-bit personal computer with 20.0 Gb of RAM, resulting a ODB file size of 3 Gb. Other methods should be explored for analyzing fatigue life distributions in 3-D including the advanced Uncertainty Quantification method proposed by Bigoni, Engsig-Karup, and True [77]. The advanced Uncertainty Quantification method required fewer simulations than the traditional Monte Carlo method, which would be advantageous for 3-D crack propagation analyses.

Bibliography

[1] Broek, D., 1988. *The Practical Use of Fracture Mechanics*. Kluwer Academic Publishers, Norwell, MA.

[2] Tada, H., Paris, P., and Irwin, G. R., eds., 1973. *The Stress Analysis of Cracks Handbook*, 3^{rd} ed. Del Research Corp., Hellertown, Pa, ch. 2, pp. 101–305. See also URL `http://www.abc.edu`.

[3] Belytschko, T., and Black, T., 1999. "Elastic Crack Growth in Finite Elements with Minimal Remeshing". *International Journal for Numerical Methods in Engineering,* **45**, pp. 601–620.

[4] Song, J.-H., Areias, P. M. A., and Belytschko, T., 2006. "A Method for Dynamic Crack and Shear Band Propagation with Phantom Nodes". *International Journal for Numerical Methods in Engineering,* **67**(6), pp. 868–893.

[5] Dassault Systèmes Simulia Corp., 2013. *Abaqus 6.13: Abaqus Analysis User's Guide*. Providence, RI.

[6] Hudson, C. M., 1969. Effect of Stress Ratio on Fatigue-Crack Growth in 7075-T6 and 2024-T3 Aluminum-Alloy Specimens. NASA Technical Note NASA TN D-5390, Langley Research Center, Langley Station, Hampton, VA, August.

[7] Broek, D., 1986. *Elementary Engineering Fracture Mechanics*. Dordrecht, Hingham, MA.

[8] Dowling, N. E., 2007. *Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue*, 3^{rd} ed. Pearson Education, Inc., Upper Saddle River, NJ.

[9] Griffith, A., 1920. "The Phenomena of Rupture and Flow in Solids". Philosophical Transactions, Vol. 221 of *A*, pp. 163–198.

[10] Irwin, G., 1956. "Onset of Fast Crack Propagation in High Strength Steel and Aluminum Alloys". *Sagamore Research Conference Proceedings,* **2**, pp. 289–305.

[11] Anderson, T., 1995. *Fracture Mechanics: Fundamentals and Applications*, 2^{nd} ed. CRC Press, Inc., Boca Raton, NY.

[12] Rice, J., 1968. "A Path Independent Integral and the Approximate Analysis of Strain Concentration by Notches and Cracks". *Journal of Applied Mechanics,* **35**(2), pp. 379–386.

[13] Shih, C., and Asaro, R., 1988. "Elastic-Plastic Analysis of Cracks on Bimaterial Interfaces: Part I-Small Scale Yielding". *Journal of Applied Mechanics*, pp. 299–316.

[14] Whitcomb, J. D., 1984. Analysis of Instability-Related Growth of a Through-Width Delamination. Report NASA TM-86301, NASA Langley Research Center.

[15] Benzeggagh, M., and Kenane, M., 1996. "Measurement of Mixed-Mode Delamination Fracture Toughness of Unidirectional Glass/Epoxy Composites with Mixed-Mode Bending Apparatus". *Composites Science and Technology,* **56**(4), pp. 439–449.

[16] Reeder, J. R., 2006. 3D Mixed-Mode Delamination Fracture Criteria - An Experimentalist's Perspective. Report No. 2006004860, NASA Langley Research Center.

[17] Paris, P., Gomez, M., and Anderson, W., 1961. "A Rational Analytic Theory of Fatigue". *The Trend in Engineering,* **13**, pp. 9–14.

[18] Paris, P., and F.Erdogan, 1963. "A Critical Analysis of Crack Propagation Laws". *Journal of Basic Engineering,* **85**(4), December, pp. 528–533.

[19] Walker, K., 1970. "The Effects of Stress Ratio During Crack Propagation and Fatigue for 2024-T3 and 7075-T6 Aluminum". Effects of Environment and Complex Load History on Fatigue Life, ASTM STP 462, American Society for Testing and Materials, pp. 1–14.

[20] Forman, R., Kearney, V., and Engle, R., 1967. "Numerical Analysis of Crack Propagation in Cyclic-Loaded Structures". *Journal of Basic Engineering,* **89**(3), September, pp. 459–463.

[21] Broek, D., and Rice, R., 2003. Fatigue Crack Growth Properties of Rail Steels. Report No. DOT-TSC-1076, Battelle Columbus Laboratory for U.S. DOT, FRA.

[22] Farris, T., Keer, L., and Steele, R., 1990. "Life Prediction for Unstable Shell Growth in Rails". *Journal of Engineering for Industry,* **112**, pp. 175–180.

[23] Hartman, A., and Schijve, J., 1970. "The Effects of Environment and Load Frequency on the Crack Propagation Law for Macro Fatigue Crack Growth in Aluminium Alloys". *Engineering Fracture Mechanics,* **1**(4), pp. 615–631.

[24] Takeo, Y., and Kiyoshi, S., 1976. "The Effect of Frequency on Fatigue Crack Propagation Rate and Striation Spacing in 2024-T3 Aluminium Alloy and SM-50 Steel". *Engineering Fracture Mechanics,* **8**(1), pp. 82–88.

[25] Yokobori, T., and Aizawa, T., 1973. "The Influence of Temperature and Stress Intensity Factor Upon the Striation Spacing and Fatigue Crack Propagation Rate of Aluminum Alloy". *International Journal of Fracture,* **9**, pp. 489–491.

[26] Murri, G., Salpekar, S., and O'Brien, T., 1991. "Fatigue Delamination Onset Prediction in Unidirectional Tapered Laminates". In *Composite Materials Fatigue and Fracture, ASTM STP 1110*, T. O'Brien, ed., Vol. 3. American Society for Testing and Materials, Philadelphia, PA, pp. 312–399.

[27] Murri, G., O'Brien, T., and Rousseau, C., 1997. "Fatigue Life Methodology for Tapered Composite Flexbeam Laminates". *American Helicopter Society* 53^{rd} *Annual Forum*, May.

[28] Giannis, S., 2013. "Utilising Fracture Mechanics Principles for Predicting the Mixed-Mode Delamination Onset and Growth in Tapered Composite Laminates". *Composite Structures,* **102**, pp. 294 – 305.

[29] ASTM Standard D6115-97, 2013. *Standard Test Method for Mode I Fatigue Delamination Growth Onset of Unidirectional Fiber-Reinforced Polymer Matrix Composites.* ASTM International, West Conshohocken, PA.

[30] Erdogan, F., and Sih, G., 1963. "On the Crack Extension in Plates Under Plane Loading and Transverse Shear". *Journal of Basic Engineering,* **85**(4), pp. 519–525.

[31] Dassault Systèmes Simulia Corp., 2013. *Abaqus 6.13: Abaqus Theory Guide.* Providence, RI.

[32] Hussain, M., Pu, S., and Underwood, J., 1974. "Strain Energy Release Rate for a Crack Under Combined Mode I and Mode II". Fracture Analysis, ASTM STP 560, American Society for Testing and Materials, pp. 2–28.

[33] Cotterell, B., and Rice, J., 1980. "Slightly Curved or Kinked Cracks". pp. 155–169.

[34] AFGROW, 2014. Fracture Mechanics and Fatigue Crack Growth Analysis software tool. On the WWW, January. URL`http://www.afgrow.net/`.

[35] NASGRO, 2014. Fracture Mechanics and Fatigue Crack Growth Analysis Software. On the WWW, January. URL`http://www.swri.org/4org/d18/mateng/matint/nasgro/`.

[36] Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J., 2002. *Concepts and Applications of Finite Element Analysis*, 4^{th} ed. John Wiley and Sons, Inc., Hoboken, NJ.

[37] Jr., V. W., 1970. "The Finite Element Method for Prediction of Crack Behavior". *Nuclear Engineering and Design,* **11**(2), pp. 323–332.

[38] Anderson, G. P., Ruggles, V. L., and Stibor, G. S., 1971. "Use of Finite Element Computer Programs in Fracture Mechanics". *International Journal of Fracture Mechanics,* **7**(1), pp. 63–76.

[39] Chan, S., Tuba, I., and Wilson, W., 1970. "On the Finite Element Method in Linear Fracture Mechanics". *Engineering Fracture Mechanics,* **2**(1), pp. 1–17.

[40] Byskov, E., 1970. "The Calculation of Stress Intensity Factors Using the Finite Element Method with Cracked Elements". *International Journal of Fracture Mechanics,* **6**(2), pp. 159–167.

[41] Wilson, W., 1973. "Finite Element Methods for Elastic Bodies Containing Cracks". In *Mechanics of Fracture*, G. Sih, ed. Noordhoff International Publishing, Leyden.

[42] Hardy, R. H., 1974. "A High-Order Finite Element for Two-Dimensional Crack Problems". PhD Thesis, Georgia Institute of Technology, Atlanta, GA, June.

[43] Holston, A., J., 1976. "A Mixed Mode Crack Tip Finite Element". *International Journal of Fracture,* **12**(6), pp. 887–899.

[44] Jiang, C., and Cheung, Y., 1995. "A Special Bending Crack Tip Finite Element". *International Journal of Fracture,* **71**(1), pp. 57–69.

[45] Henshell, R., and Shaw, K., 1975. "Crack Tip Finite Elements are Unnecessary". *International Journal for Numerical Methods,* **9**(3), pp. 495–507.

[46] Barsoum, R., 1976. "On the Use of Isoparametric Finite Elements in Linear Elastic Fracture Mechanics". *International Journal for Numerical Methods,* **10**(1), pp. 25–37.

[47] Barsoum, R., 1977. "Triangular Quarter Point Elements as Elastic and Perfectly Plastic Crack Tip Element". *International Journal for Numerical Methods,* **11**(1), pp. 85–98.

[48] Hojfeldt, E., and Ostervig, C. B., 1986. "Fatigue Crack Propagation in Shafts with Shoulder Fillets". *Engineering Fracture Mechanics,* **25**(4), pp. 421 – 427.

[49] Alegre, J., and Cuesta, I., 2010. "Some Aspects about the Crack Growth FEM Simulations Under Mixed-Mode Loading". *International Journal of Fatigue,* **32**(7), pp. 1090 – 1095.

[50] Rybicki, E., and Kanninen, M., 1977. "A Finite Element Calculation of Stress Intensity Factors by a Modified Crack Closure Integral". *Engineering Fracture Mechanics,* **9**(4), pp. 931 – 938.

[51] Shivakumar, K., Tan, P., and J.C. Newman, J., 1988. "A Virtual Crack-Closure Technique for Calculating Stress Intensity Factors for Cracked Three Dimensional Bodies". *International Journal of Fracture,* **36**, pp. 43 – 50.

[52] Mabson, G., Deobald, L., Dopker, B., Hoyt, D., Baylor, J., and Graesser, D., 2007. "Fracture Interface Elements for Static and Fatigue Analysis". 16^{th} *International Conference on Composite Materials.*

[53] Hosseini-Toudeshky, H., Ghaffari, M. A., and Mohammadi, B., 2013. "Mixed-Mode Crack Propagation of Stiffened Curved Panels Repaired by Composite Patch Under Combined Tension and Shear Cyclic Loading". *Aerospace Science and Technology,* **28**(1), pp. 344 – 363.

[54] Melenk, J., and Babuška, I. "The Partition of Unity Finite Element Method: Basic Theory and Applications". *Computer Methods in Applied Mechanics and Engineering,* **139**.

[55] Moës, N., Dolbow, J., and Belytschko, T., 1999. "A Finite Element Method for Crack Growth without Remeshing". *International Journal for Numerical Methods in Engineering,* **46**, pp. 131–150.

[56] Osher, S., and Sethian, J. A., 1988. "Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations". *Journal of Computational Physics,* **79**(1), pp. 12 – 49.

[57] Stolarska, M., Chopp, D. L., Mëos, N., and Belytschko, T., 2001. "Modelling Crack Growth by Level Sets in the eXtended Finite Element Method". *International Journal for Numerical Methods in Engineering,* **51**(8), pp. 943–960.

[58] Shi, J., Chopp, D., Lua, J., Sukumar, N., and Belytschko, T., 2010. "Abaqus Implementation of eXtended Finite Element Method Using a Level Set Representation for Three-Dimensional Fatigue Crack Growth and Life Predictions". *Engineering Fracture Mechanics,* **77**(14), pp. 2840 – 2863.

[59] Singh, I., Mishra, B., Bhattacharya, S., and Patil, R., 2012. "The Numerical Simulation of Fatigue Crack Growth Using eXtended Finite Element Method". *International Journal of Fatigue,* **36**(1), pp. 109 – 119.

[60] Pathak, H., Singh, A., and Singh, I. V., 2013. "Fatigue Crack Growth Simulations of 3-D Problems Using XFEM". *International Journal of Mechanical Sciences,* **76**, pp. 112 – 131.

[61] Hansbo, A., and Hansbo, P. "A Finite Element Method for the Simulation of Strong and Weak Discontinuities in Solid Mechanics". *Computer Methods in Applied Mechanics and Engineering,* **193**.

[62] Kucharczyk, P., Sharaf, M., and Münstermann, S., 2012. "On the Influence of Steel Microstructure on Short Crack Growth Under Cyclic Loading". *International Journal of Fatigue,* **41**, pp. 83 – 89.

[63] Virkler, D., Hillberry, B., and Goel, P., 1979. "The Statistical Nature of Fatigue Crack Propagation". *Journal of Engineering Materials and Technology,* **101**, pp. 148 – 153.

[64] Dunn, W., and Shultis, J., 2012. *Exploring Monte Carlo Methods.* Academic Press, San Diego, CA.

[65] Annis, C., 2004. "Probabilistic Life Prediction Isn't as Easy as It Looks". *Journal of ASTM International,* **1**(2).

[66] ASTM Standard E647-13e1, 2013. *Standard Test Method for Measurement of Fatigue Crack Growth Rates.* ASTM International, West Conshohocken, PA.

[67] ASTM Standard E399-12, 2013. *Standard Test Method for Linear-Elastic Plane-Strain Fracture Toughness KIc of Metallic Materials*. ASTM International, West Conshohocken, PA.

[68] Liaw, P., Leax, T., and Donald, J., 1987. "Fatigue Crack Growth Behavior of 4340 Steels". *Acta Metallurgica,* **35**(7), pp. 1415–1432.

[69] Frost, N., 1962. "Effect of Mean Stress on the Rate of Growth of Fatigue Cracks in Sheet Materials". *Journal of Mechanical Engineering Science,* **4**(1), pp. 22–35.

[70] Wang, C., and Miller, K., 1992. "The Effects of Mean and Alternating Shear Stresses on Short Fatigue Crack Growth Rates". *Fatigue I& Fracture of Engineering Materials I& Structures,* **15**(12), pp. 1223–1236.

[71] Isida, M., 1971. "Effect of Width and Length on Stress Intensity Factors of Internally Cracked Plates Under Various Boundary Conditions". *International Journal of Fracture Mechanics,* **7**(3), pp. 301–316.

[72] McGinty, R., 2011. Handbook for Damage Tolerance Design: Sensitivity of Fatigue Crack Growth Rates to Operating Conditions. Afgrow example, Mercer Engineering Research Group, Warner Robins, GA. See also URL `http://www.afgrow.net/applications/DTDHandbook/`.

[73] Rice, J., 1967. "Mechancis of Crack Tip Deformation and Extension by Fatigue". Fatigue Crack Propagation, ASTM STP 415, American Society for Testing and Materials, pp. 247–309.

[74] MatWeb, 2014. Material Property data. See also URL `http://www.matweb.com`, April.

[75] Dassault Systèmes Simulia Corp., 2013. *Abaqus 6.13: Abaqus Keywords Reference Guide*. Providence, RI.

[76] Walpole, R., Myers, R., Myers, S., and Ye, K., 2002. *Probability and Statistics for Engineers and Scientists*, 7^{th} ed. Prentice Hall, Upper Saddle River, NJ.

[77] Bigoni, D., Engsig-Karup, A., and True, H., 2013. "Modern Uncertainty Quantification Methods in Railroad Vehicle Dynamics". *Rail Transportation Division Fall Technical Conference.*

[78] Orringer, O., Tang, Y., Gordon, J., Jeong, D., Morris, J., and Perlman, A., 1988. Crack Propagation Life of Detail Fractures in Rails. Final Report DOT-TSC-FRA-88-1, Federal Railroad Administration, Washington, DC, October.

[79] Feddersen, C., and Broek, D., 1978. "Fatigue Crack Propagation in Rail Steels". Rail Steels-Developments, Processing, and Use, ASTM STP 644, D. Stone and G. Knupp, eds., American Society for Testing and Materials, pp. 414–429.

[80] Jeong, D., Tang, Y., Orringer, O., and Perlman, A., 1998. Propagation Analysis of Transverse Defects Originating at the Lower Gage Corner of Rail. Final Report DOT-VNTSC-FRA-98-14, Federal Railroad Administration, Washington, DC, December.

Appendix A

Material Properties

Table A.1 lists the elastic material properties of 7075-T6 aluminum alloy as determined in Hudson's report [6] with a standard tensile test. Poisson's ratio, ν, was not provided in Hudson's report, so a value of 0.33 listed on www.matweb.com [74] was assumed. Table A.2 lists the results from fatigue crack growth tests, listing the mean stress, σ_m, and alternating stress, σ_a, along with the cycles required to grow a crack from an initial half length of 0.1 in. to the half length listed in each column. Table A.3 shows the results from the residual static strength test. Using the final load, P_f, and the final crack length, a_f, the critical stress intensity factor can be calculated for the material and geometry. A detailed drawing of the center crack plate specimen is shown in Fig. A.1.

Table A.1: Average Tensile Properties for 7075-T6 Aluminum Alloy

σ_{ult}		σ_{yld}[(1)]		E		Percent[(2)]
(ksi)	(MPa)	(ksi)	(MPa)	(ksi)	(GPa)	Elongation
83.2	574	75.9	523	10 100	69.6	12

Source: [6]

(1) 0.2% Offset

(2) 2-inch (51-mm) gage length

Table A.2: Fatigue Crack Growth Test Results for 7075-T6 Aluminum Alloy

σ_m		σ_a		R	Freq	Average Number of Cycles Required to Propagate a Crack from a Half-Length, a, of 0.10 in. to a Half-Length of _ in. (mm)												
(ksi)	(MPa)	(ksi)	(MPa)		(Hz)	0.2 (5.08)	0.3 (7.62)	0.4 (10.2)	0.5 (12.7)	0.6 (15.2)	0.7 (17.8)	0.8 (20.3)	0.9 (22.9)	1.0 (25.4)	1.2 (30.5)	1.4 (35.6)	1.6 (40.6)	1.8 (45.7)
25.0	172	25.0	172	0	0.5	330	413											
20.0	138	20.0	138	0	0.5	700	930	1 030	1 075	1 100	1 110							
15.0	103	15.0	103	0	0.5,13.7	1 670	2 360	2 660	2 820	2 930	3 000	3 050						
10.0	69	10.0	69	0	20.0	6 140	8 800	10 370	11 300	11 800	12 300	12 570	12 800	12 950	13 150			
5.0	34	5.0	34	0	30.0	70 000	88 300	98 300	105 000	110 000	113 000	117 000	120 000	122 000	127 000	129 000	131 000	132 000
30.0	207	20.0	138	0.2	0.25	410	490											
25.0	172	16.7	115	0.2	0.5	760	990	1 100	1 150	1 180								
20.0	138	13.3	92	0.2	0.5	1 680	2 340	2 640	2 830	2 960	3 060	3 120						
15.0	103	10.0	69	0.2	0.5	3 800	5 650	6 550	7 150	7 500	7 750	7 950	8 080	8 190	8 340	8 420		
30.0	207	15.0	103	0.3	0.5,13.7	810	1 050	1 130	1 170									
25.0	172	12.5	86	0.3	0.5	1 500	2 060	2 320	2 470	2 570	2 630	2 680						
20.0	138	10.0	69	0.3	0.5,13.7	3 040	4 260	4 900	5 280	5 550	5 720	5 850	5 940					
15.0	103	7.5	52	0.3	13.7	7 350	10 100	11 700	12 800									
10.0	69	5.0	34	0.3	30.0	16 100	23 700	28 900	32 600	35 500	37 800	39 500	40 800	42 000	43 500	44 500		
30.0	207	10.0	69	0.5	0.5,13.7	2 230	3 040	3 430	3 660	3 810	3 920	3 990						
25.0	172	8.3	57	0.5	0.5	4 100	5 600	6 380	6 800	7 100	7 260	7 380	7 480	7 500				
20.0	138	6.7	46	0.5	13.7	9 500	12 400	13 820	14 730	15 400								
15.0	103	5.0	34	0.5	13.7	18 500	28 000	32 800	35 600	37 600	39 000	40 000	40 800	41 400	42 000	42 500		
10.0	69	3.0	21	0.5	30.0	105 000	140 000	160 000	175 000	185 000	192 500	197 500	202 500	205 000	212 000	215 000	217 500	
30.0	207	5.0	34	0.7	13.7	12 300	16 300	18 300	19 400	20 200	20 700	21 000						
25.0	172	4.4	30	0.7	13.7	19 800	27 000	30 900	33 000	34 200	35 200	36 000	36 500					
20.0	138	3.0	21	0.7	13.7	69 000	97 000	112 000	121 000	127 000	130 000	134 000	136 000	138 000	141 000	143 000	144 000	
15.0	103	3.0	21	0.7	13.7	74 000	98 000	115 000	125 000	132 000	137 000	141 000	144 000	147 000	150 000	152 000	153 000	154 000
10.0	69	1.7	12	0.7	13.7	780 000	1 260 000	1 540 000	1 690 000	1 780 000	1 825 000	1 850 000	1 870 000	1 885 000	1 915 000	1 930 000	1 935 000	
30.0	207	3.0	21	0.8	13.7	42 500	58 750	68 750	75 000	77 500	80 000	82 000	83 750	84 000	85 000			
25.0	172	2.8	19	0.8	13.7	71 000	94 000	106 000	111 000	116 000	120 000	122 000	123 000	124 000	126 000			

Source: [6]

Table A.3: Residual Static Strength Test Results for 7075-T6 Aluminum Alloy

Observation	a_f		P_f	
	(in.)	(mm)	(kips)	(kN)
1	1.85	46.99	29.9	133
2	1.80	45.72	30.4	135
3	1.79	45.47	30.4	135
4	2.36	59.94	27.2	121
5	1.16	29.46	38.5	171
6	1.19	30.23	38.8	173
7	0.96	24.38	43.1	192
8	1.09	27.69	41.7	185
9	1.15	29.21	37.7	168
10	0.94	23.88	44.5	198
11	1.75	44.45	32.0	142
12	2.19	55.63	25.0	111
13	0.73	18.54	50.4	224
14	1.53	38.86	35.9	160
15	2.10	53.34	30.2	134
16	1.73	43.94	32.3	144
17	1.43	36.32	36.4	162
18	0.71	18.03	49.8	222
19	1.65	41.91	39.0	173
20	2.73	69.34	24.3	108
21	2.81	71.37	22.8	101
22	2.13	54.10	27.1	121
23	0.93	23.62	47.1	210
24	0.78	19.81	48.4	215
25	1.20	30.48	39.0	173
26	1.37	34.80	36.0	160
27	0.72	18.29	47.5	211

Source: [6]

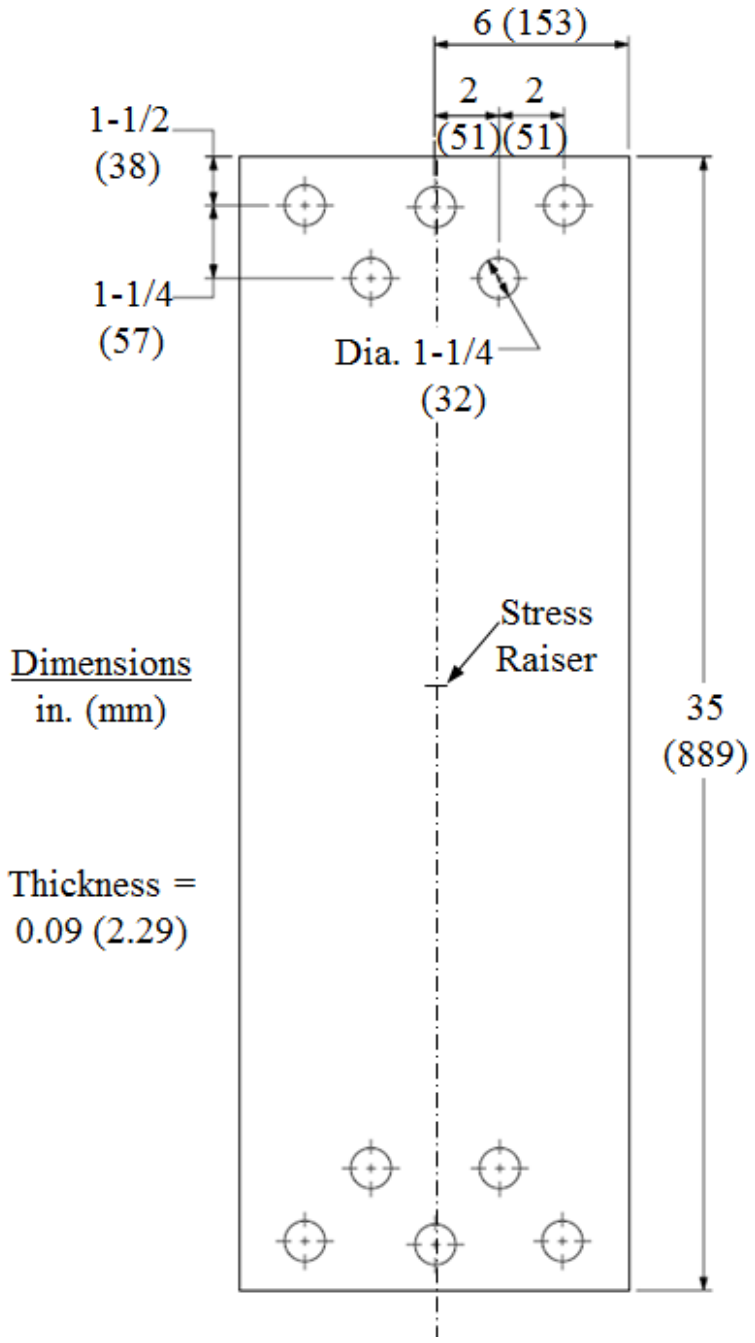

Fig. A.1: Detailed center crack plate specimen

Appendix B

Center Cracked Plate Analysis Results

The following figures show additional results from the analyses performed in Chapter 4. Figs. B.1-B.4 plot the normalized ΔK_I versus the ratio of crack half-length to specimen half-width, a/b, for the loading conditions listed in Table B.1. Root Mean Square Error (RMSE) for the normalized ΔK_I curves compared to the analytical method are provided in Table B.2. Cycle count during propagation is shown in Figs. B.5-B.8 with RMSE values for each loading condition and analysis method in Table B.3. Crack propagation direction for the three criteria in an FEQP analysis, as well as Maximum Tangential Stress (MTS) for XFEM-PN are plotted in Figs. B.9-B.12.

Resulting finite element model deformations modes and principal strains are provided for each of the finite element analyses. Deformations modes show the zero displacement boundary conditions along the axis of symmetry in the x direction and along the bottom in the y direction. Maximum and minimum principal strains, ϵ_{max} and ϵ_{min}, were provided as they are independent of coordinate system. All figures depict the critical crack length, a_c, denoting fracture is imminent. Overall specimens are shown on the left of each figure with insets at the crack tip on the right. All images have a deformation scale factor of 40 and color contours are unaveraged. Two-dimensional quadratic quadrilateral elements in the FEQP Figs. B.13-B.16 result in smooth contours, as well as significantly higher ϵ_{max} at the crack tip since mid-side nodes were moved to the quarter point location. Both VCCT in Figs. B.17-B.20 and XFEM-PN in Figs. B.21-B.24 use two-dimensional linear quadrilateral elements. Discretization of the elements is more noticeable in the XFEM-PN model since a courser mesh was used compared to the VCCT model. Neither VCCT nor XFEM-PN are able to adequately model the singularity at the crack tip as expected since these methods were not developed for this purpose.

Table B.1: Analyzed Loading Conditions

Loading Condition	σ_m (ksi)	σ_m (MPa)	σ_a (ksi)	σ_a (MPa)	σ_{max} (ksi)	σ_{max} (MPa)	$\Delta\sigma$ (ksi)	$\Delta\sigma$ (MPa)	R	Frequency (Hz)
1	15.0	103	15.0	103	30.0	207	30.0	207	0	0.5/13.7*
2	15.0	103	10.0	68.9	25.0	172	20.0	138	0.2	0.5
3	15.0	103	5.00	34.5	20.0	138	10.0	68.9	0.5	13.7
4	15.0	103	3.00	20.7	18.0	124	6.00	41.4	0.67	13.7

* Indicates multiple tests run a different frequencies.

Table B.2: RMSE Compared to the Analytical Method for the Normalized ΔK_I Curves in Figs. B.1-B.4.

Loading Condition	R	ΔK_I RMSE Compared to the Analytical Method (%)			
		AFGROW	FEQP	VCCT	XFEM
1	0	0.28	0.23	1.0	4.5
2	0.2	0.38	0.32	0.86	4.4
3	0.5	0.48	0.41	0.80	4.3
4	0.67	0.49	0.43	0.80	4.3

Table B.3: RMSE Compared to the Analytical Method for the Crack Growth Curves in Figs. B.5-B.8.

Loading Condition	R	N RMSE Compared to the Analytical Method (%)				
		Experiments	AFGROW	FEQP	VCCT	XFEM
1	0	44	2.2	0.08	8.6	20.3
2	0.0	13	2.1	0.09	8.6	20.2
3	0.0	13	2.1	0.12	8.6	20.2
4	0.00	13	2.0	0.13	8.6	20.2

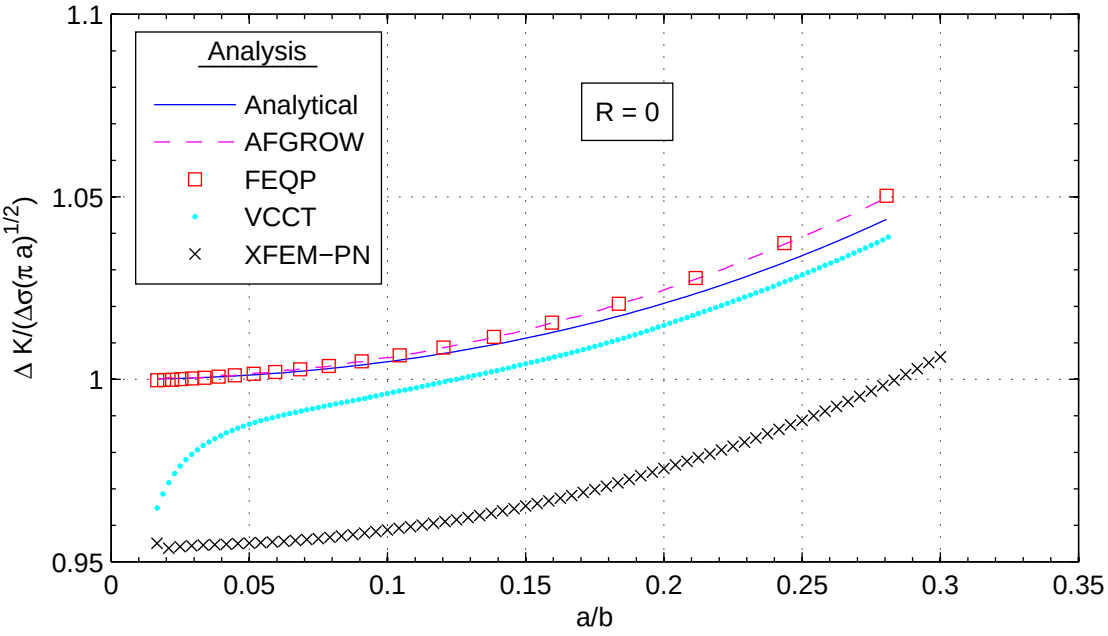

Fig. B.1: Normalized ΔK_I during crack propagation ($R = 0$).

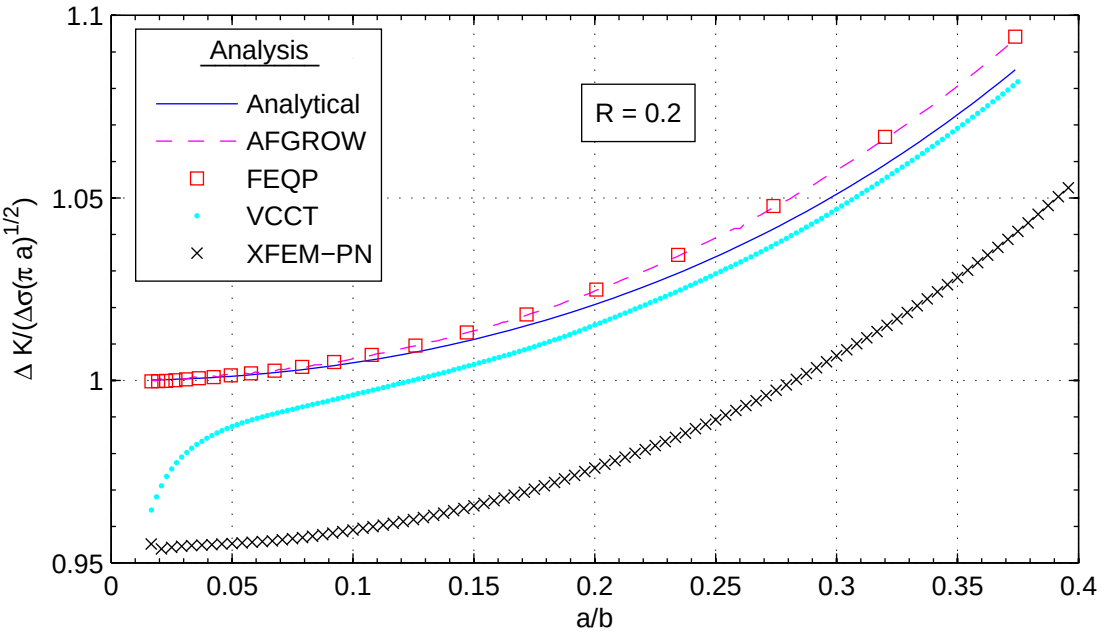

Fig. B.2: Normalized ΔK_I during crack propagation ($R = 0.2$).

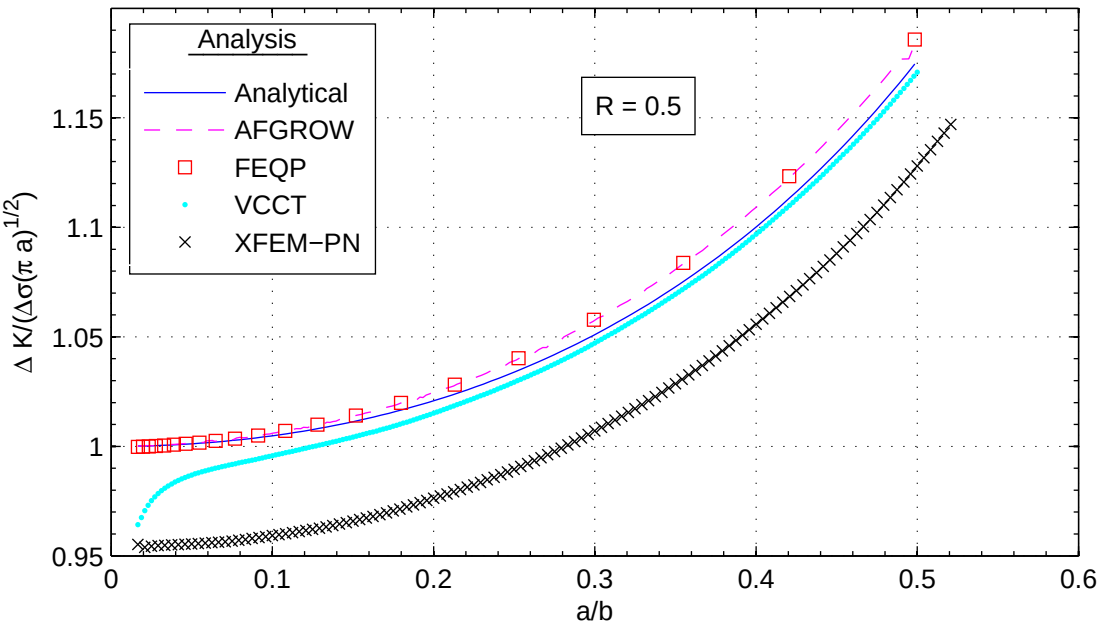

Fig. B.3: Normalized ΔK_I during crack propagation ($R = 0.5$).

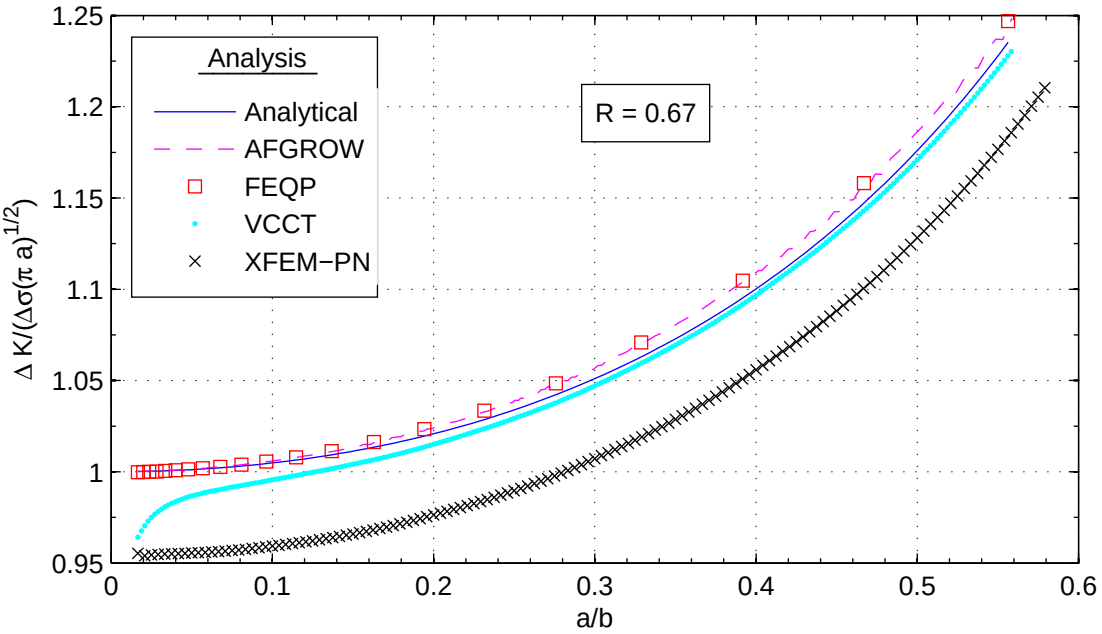

Fig. B.4: Normalized ΔK_I during crack propagation ($R = 0.67$).

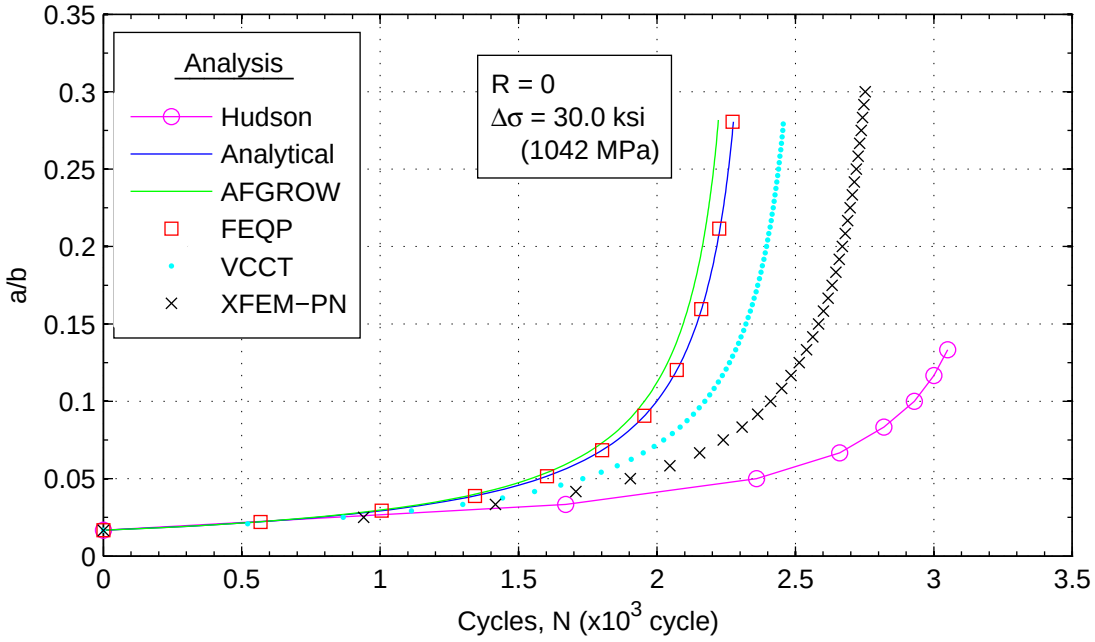

Fig. B.5: Cycle count during crack propagation ($R = 0$).

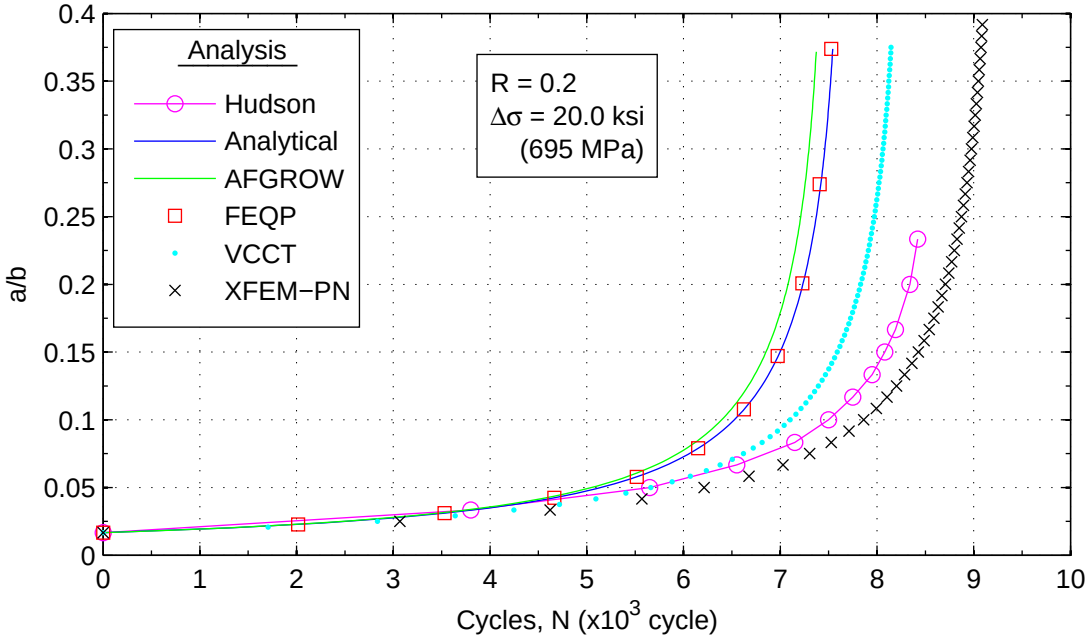

Fig. B.6: Cycle count during crack propagation ($R = 0.2$).

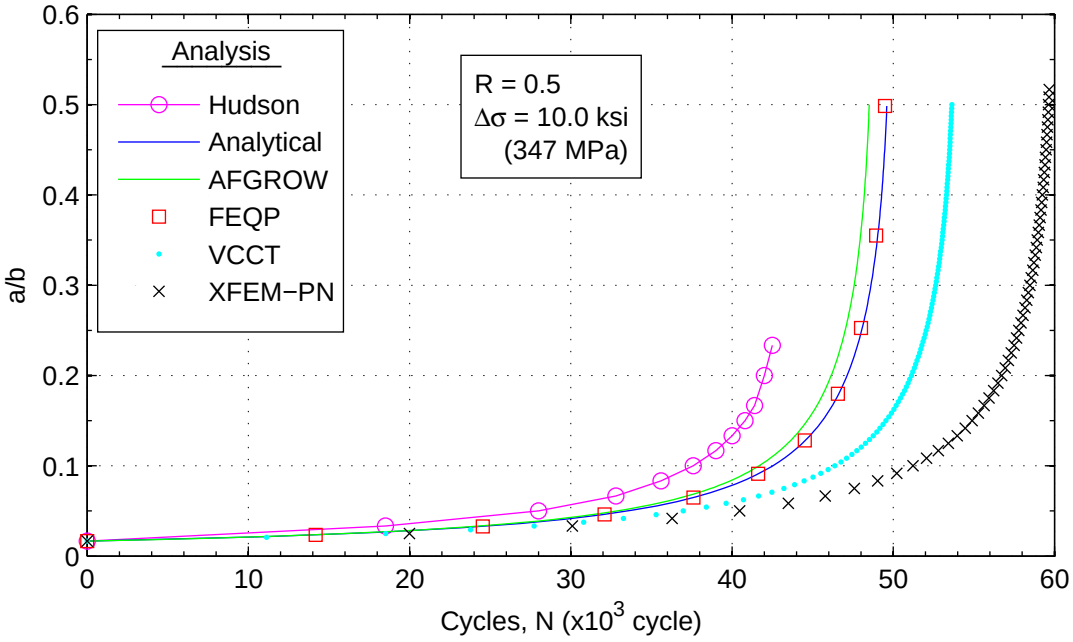

Fig. B.7: Cycle count during crack propagation ($R = 0.5$).

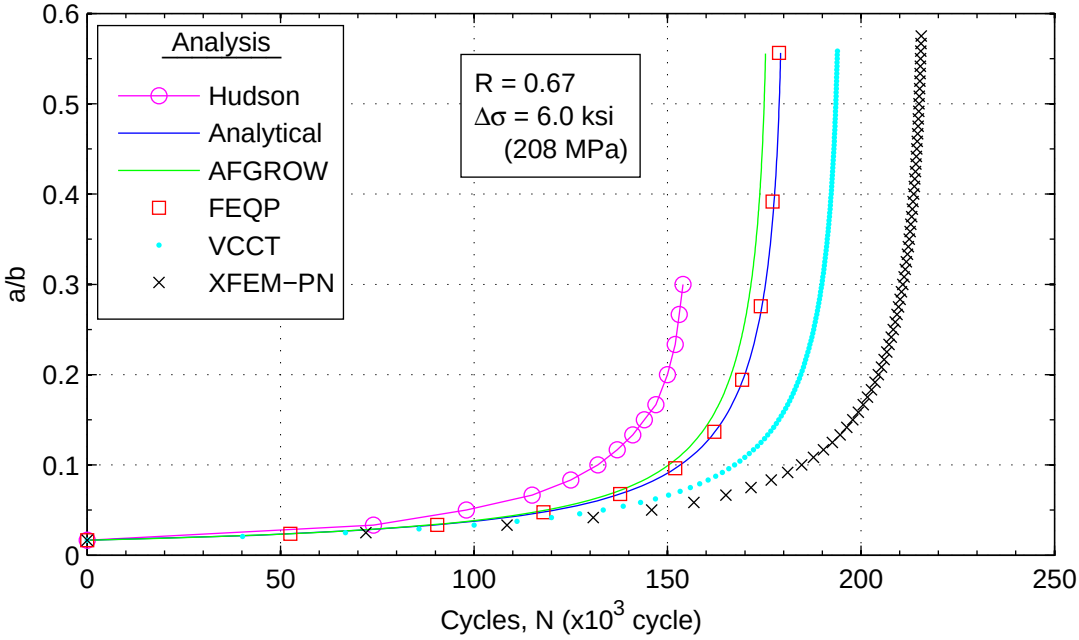

Fig. B.8: Cycle count during crack propagation ($R = 0.67$).

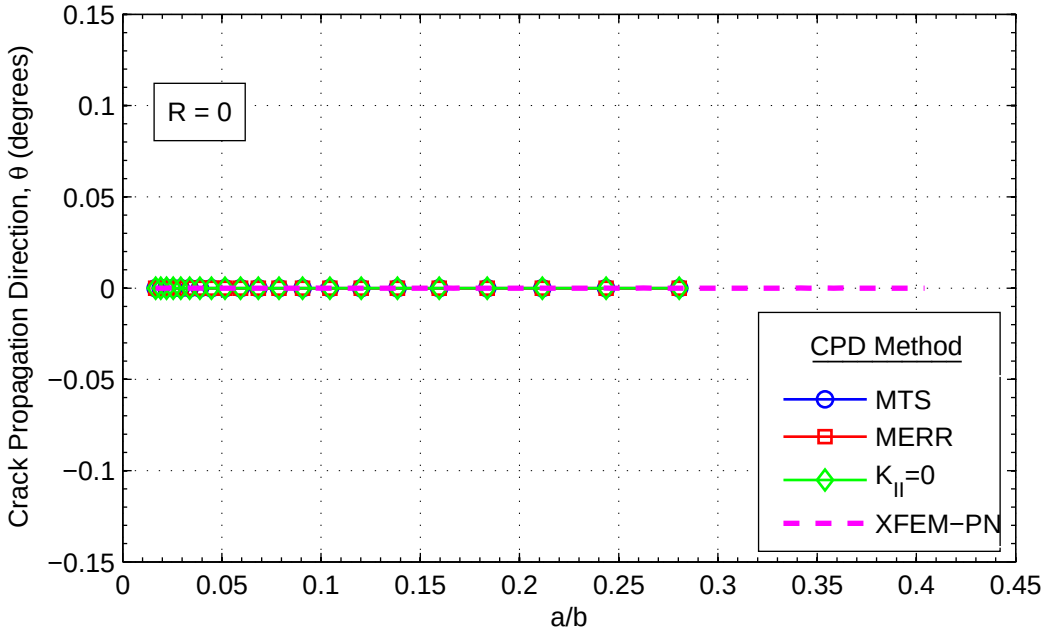

Fig. B.9: Crack extension direction during propagation ($R = 0$).

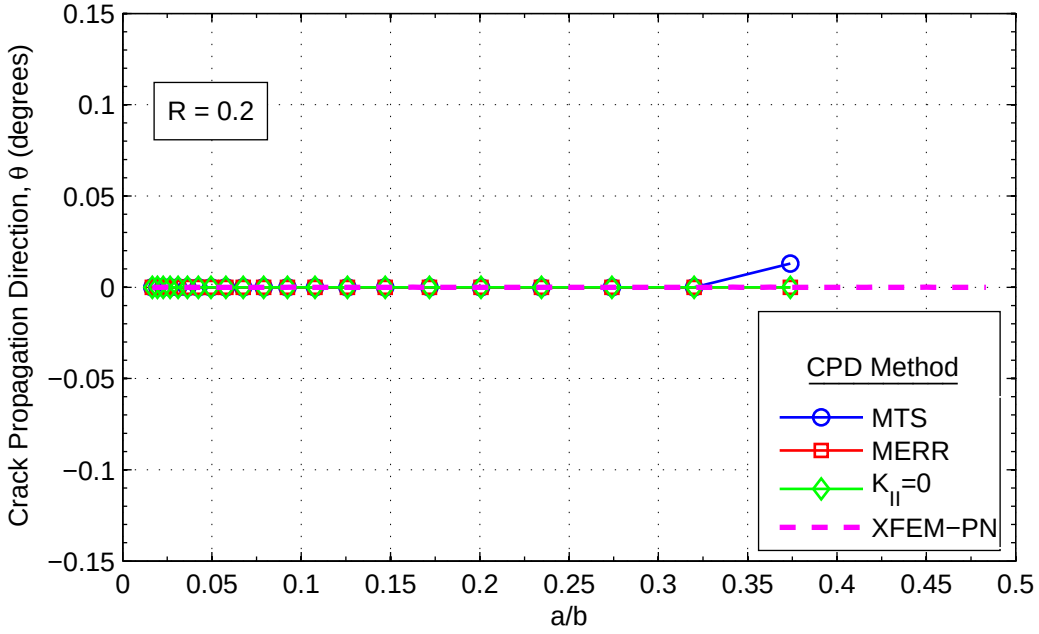

Fig. B.10: Crack extension direction during propagation ($R = 0.2$).

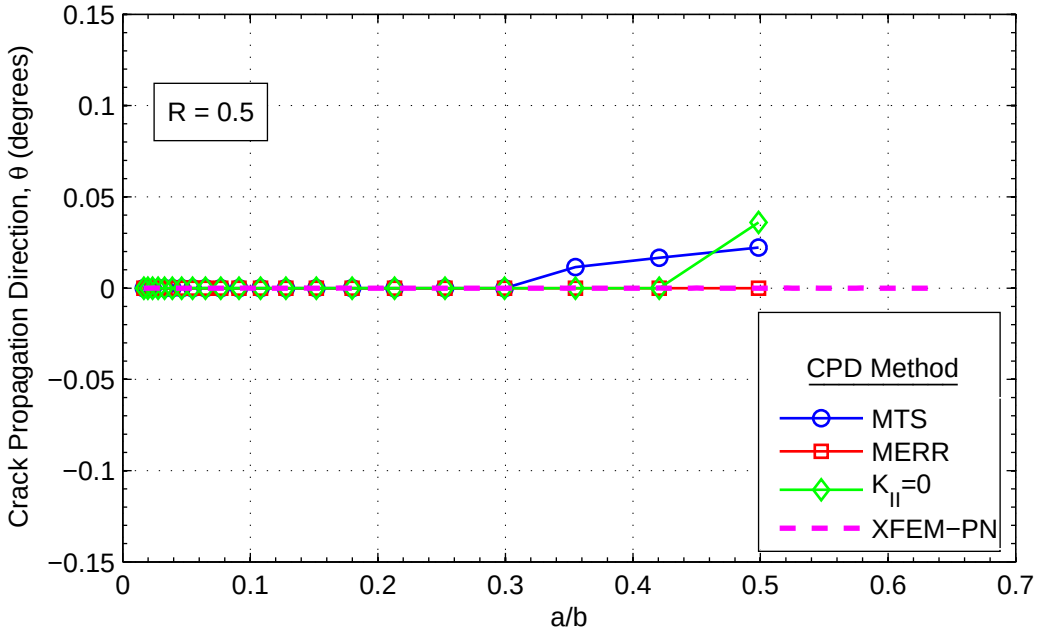

Fig. B.11: Crack extension direction during propagation ($R = 0.5$).

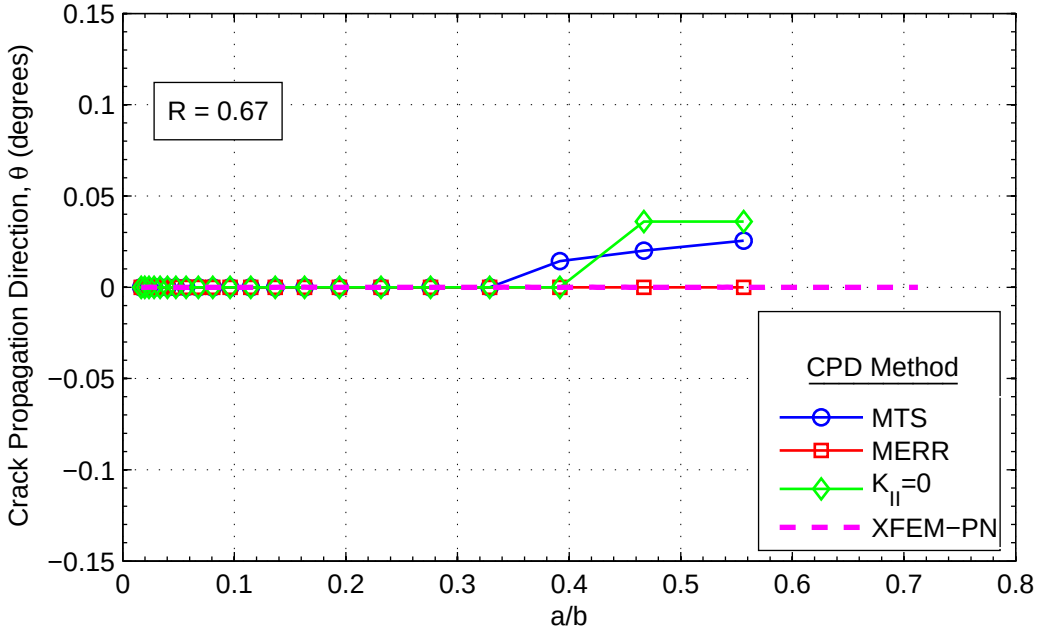

Fig. B.12: Crack extension direction during propagation ($R = 0.67$).

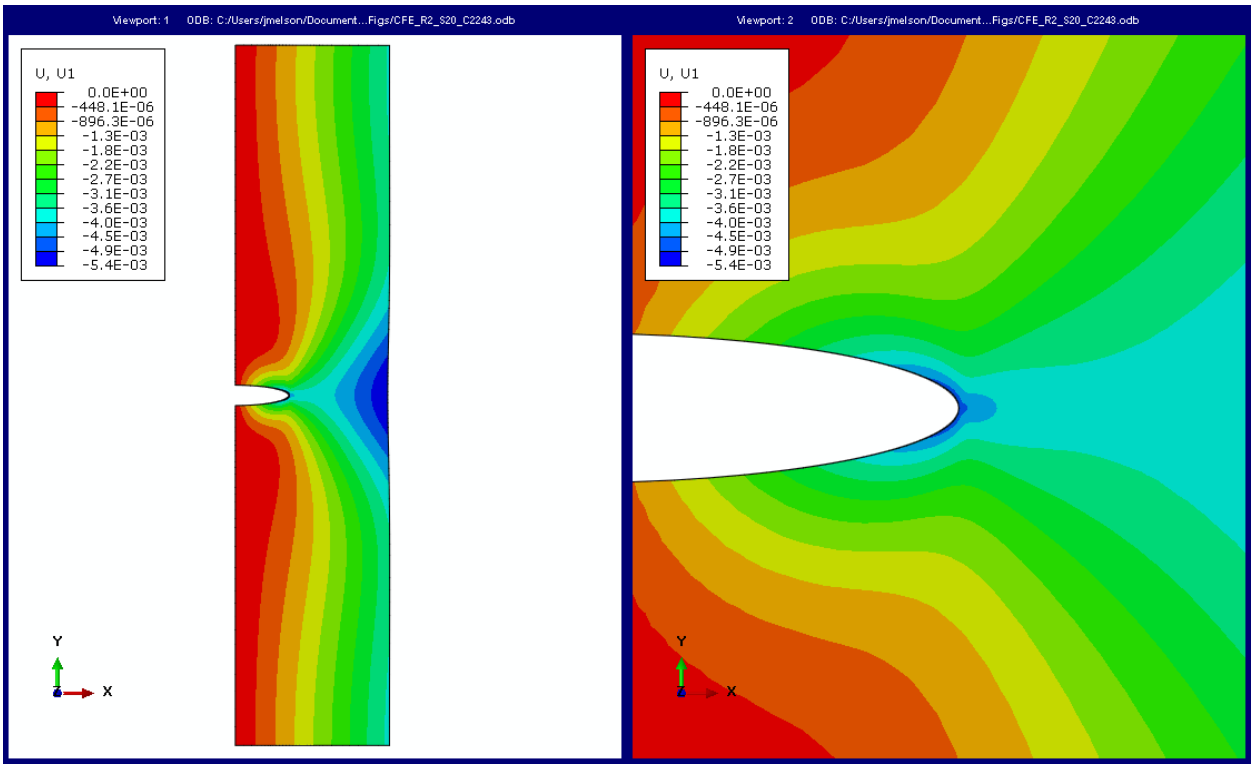

Fig. B.13: FEQP, deformation mode u_x with $a = 2.24\,\text{in.}$ $(R = 0.2)$.

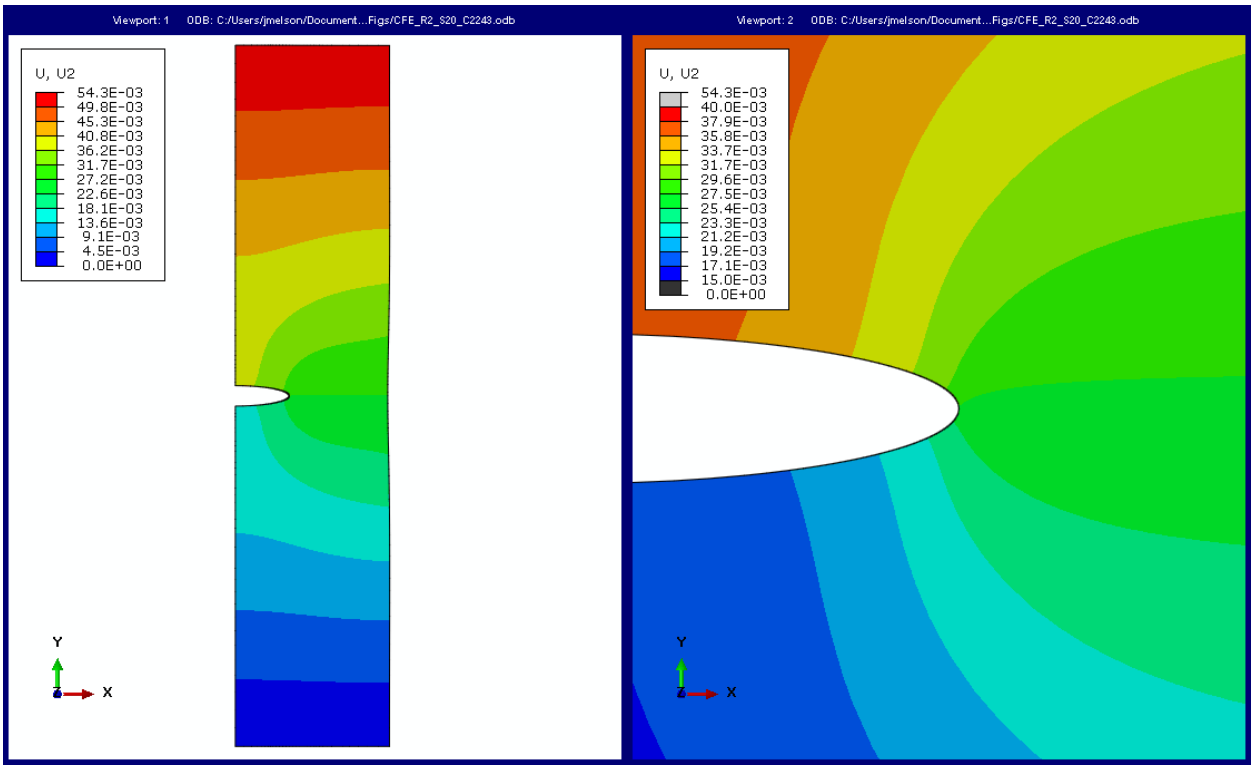

Fig. B.14: FEQP, deformation mode u_y with $a = 2.24\,\text{in.}$ $(R = 0.2)$.

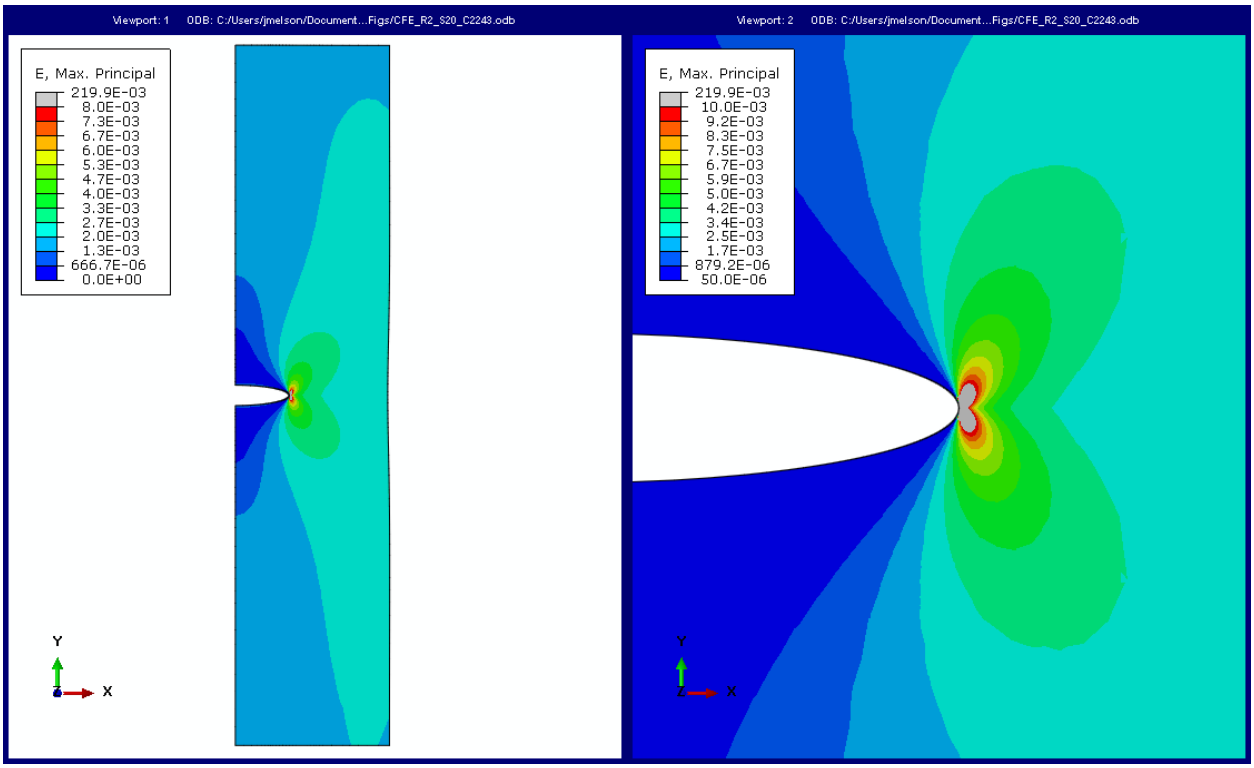

Fig. B.15: FEQP, principal strain ϵ_{max} with $a = 2.24\,\text{in.}$ $(R = 0.2)$.

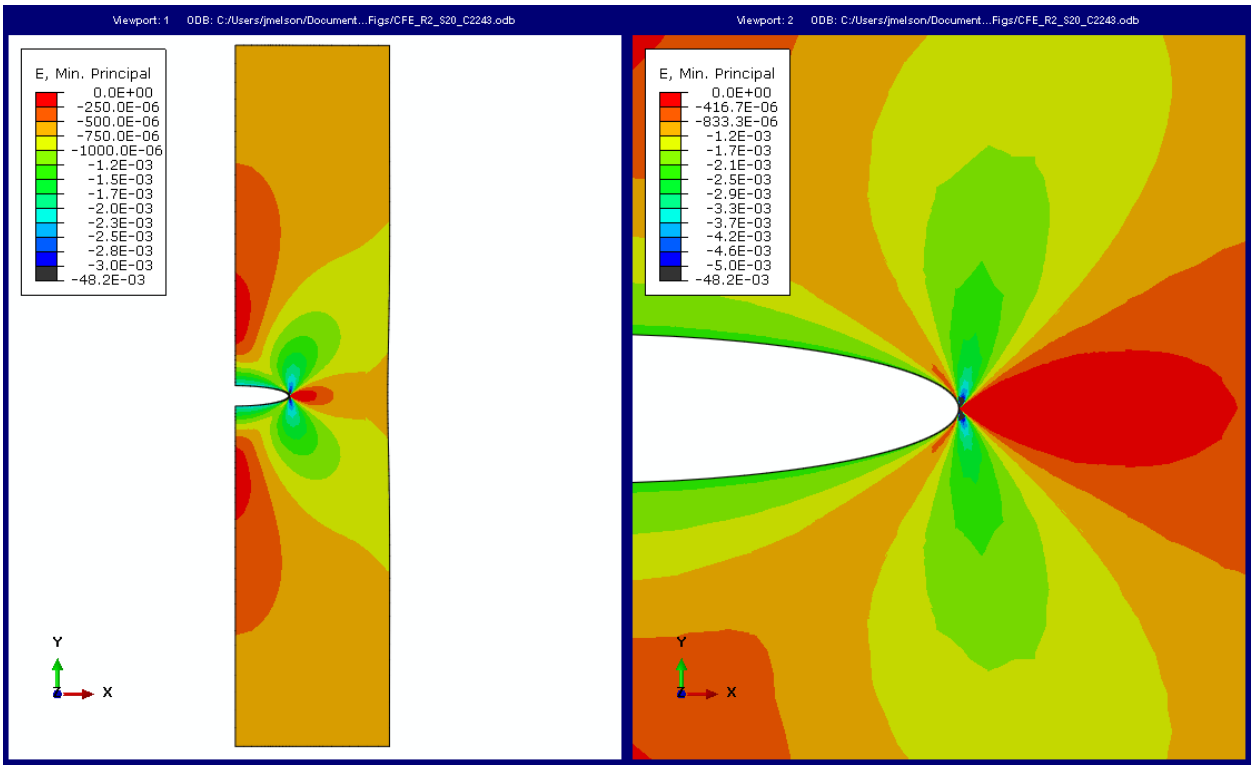

Fig. B.16: FEQP, principal strain ϵ_{min} with $a = 2.24\,\text{in.}$ $(R = 0.2)$.

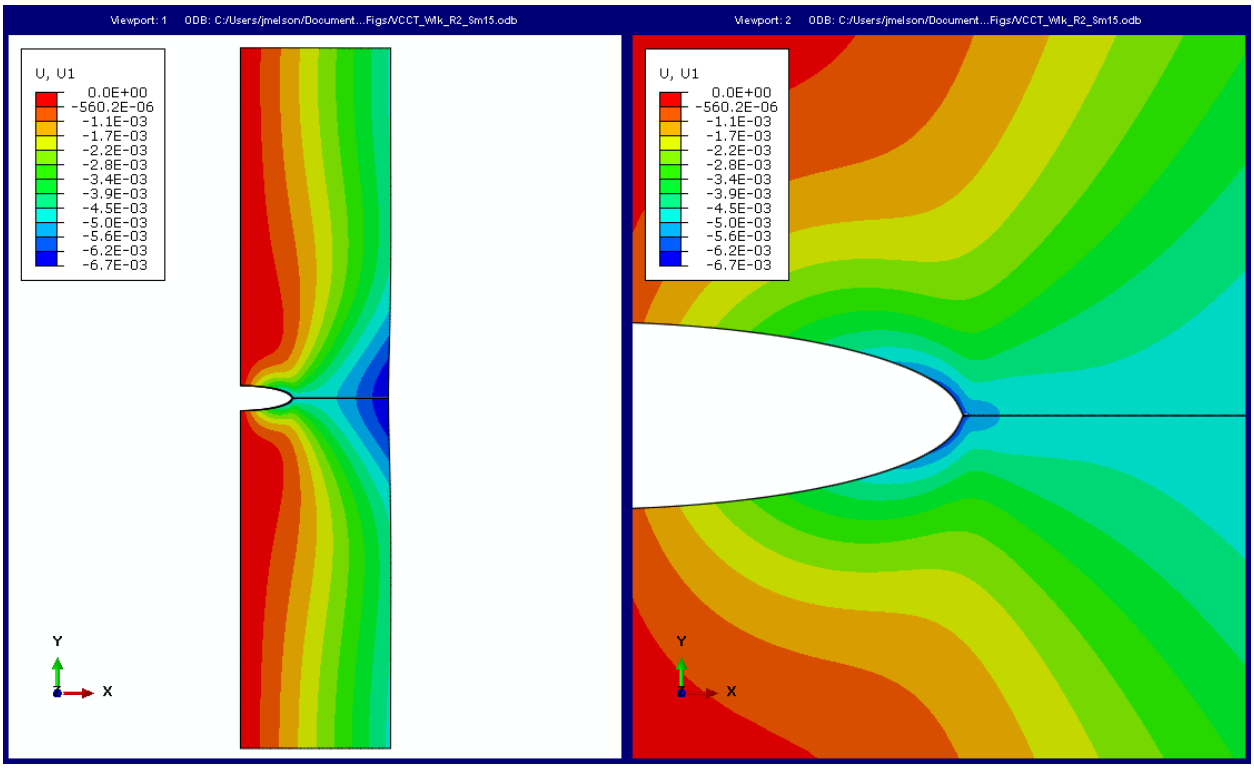

Fig. B.17: VCCT, deformation mode u_x with $a = 2.25\,\text{in.}$ $(R = 0.2)$.

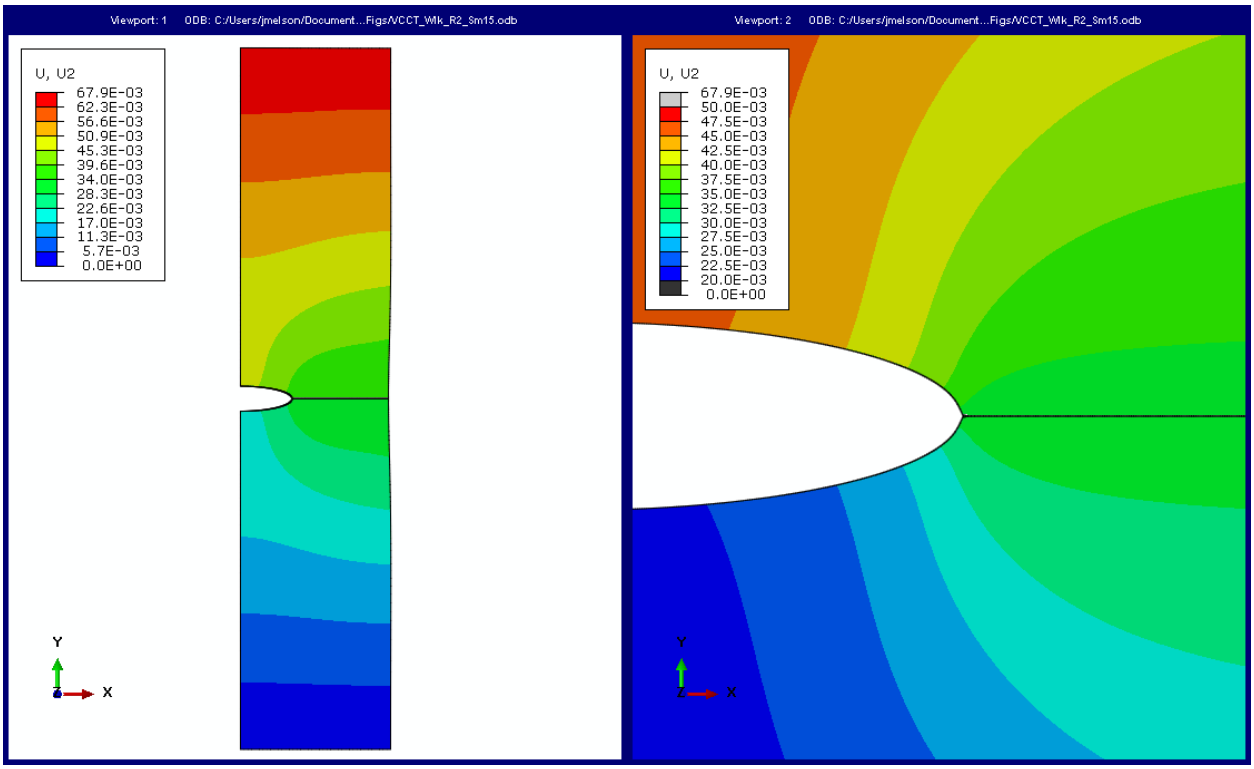

Fig. B.18: VCCT, deformation mode u_y with $a = 2.25\,\text{in.}$ $(R = 0.2)$.

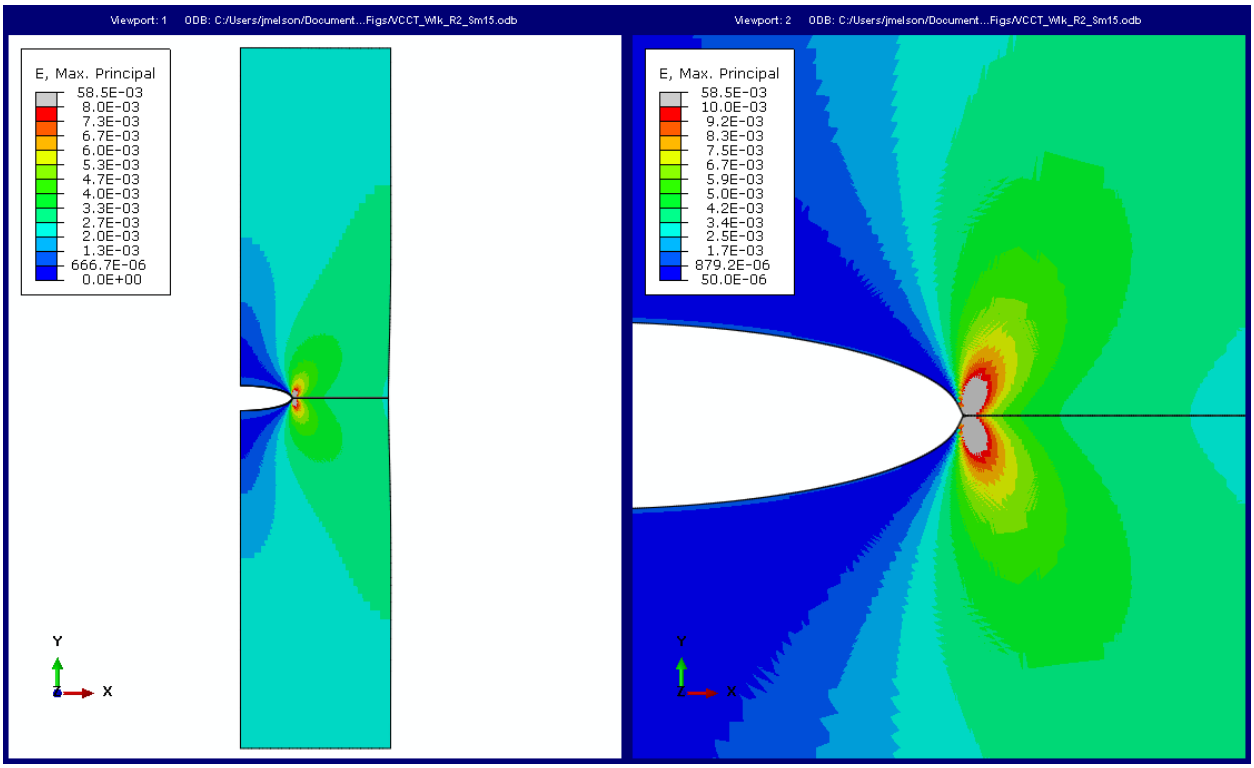

Fig. B.19: VCCT, principal strain ϵ_{max} with $a = 2.25\,\text{in.}$ $(R = 0.2)$.

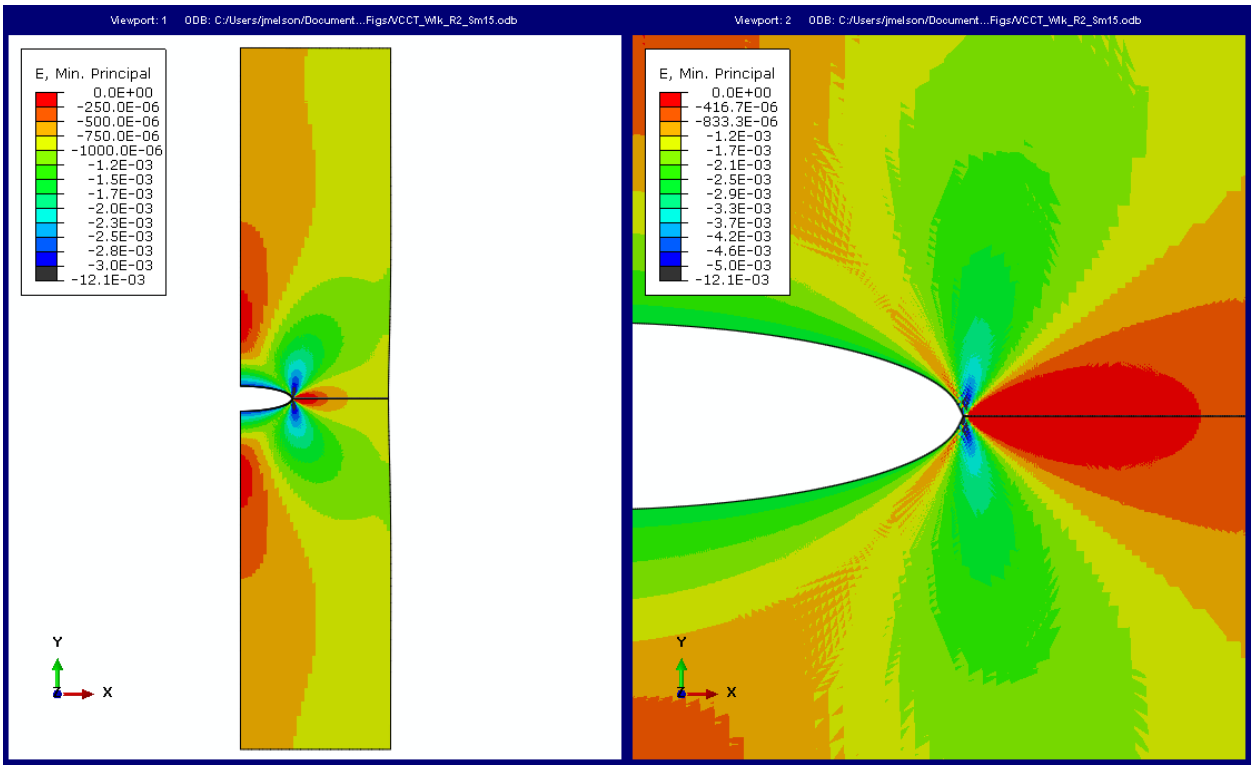

Fig. B.20: VCCT, principal strain ϵ_{min} with $a = 2.25\,\text{in.}$ $(R = 0.2)$.

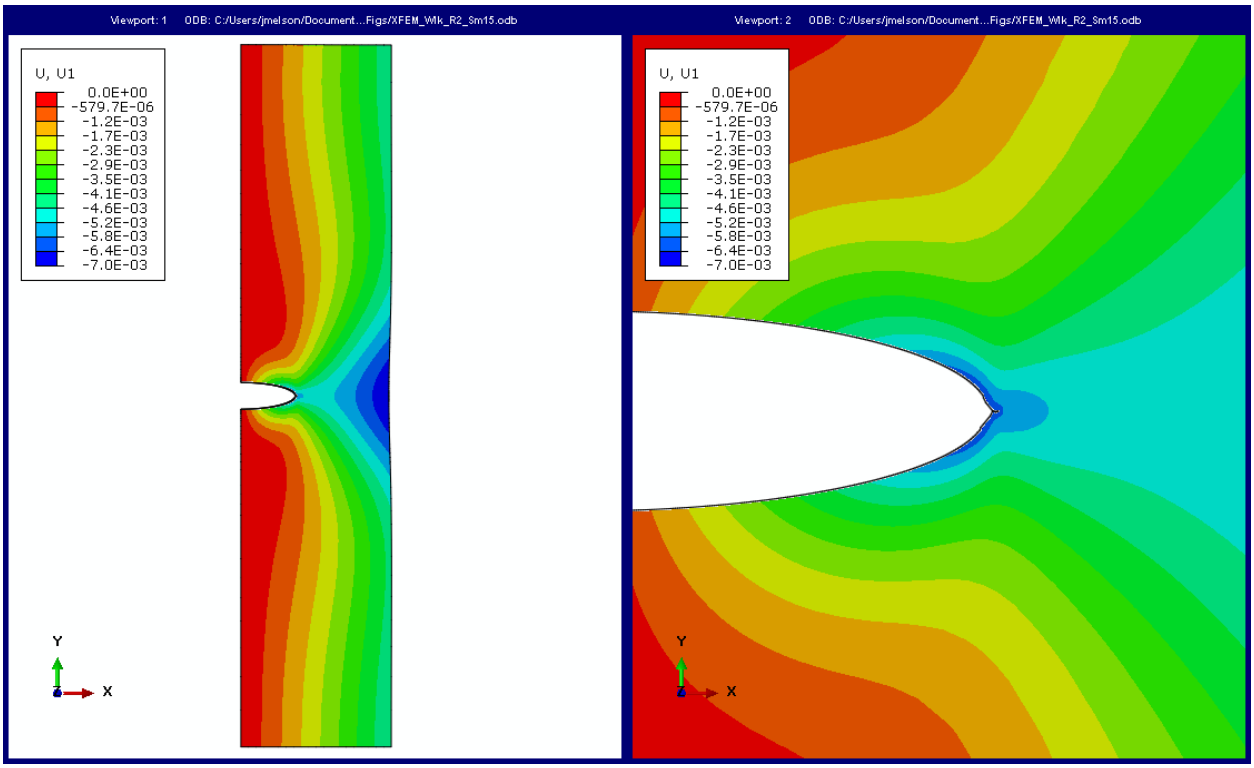

Fig. B.21: XFEM-PN, deformation mode u_x with $a = 2.38\,\text{in.}$ $(R = 0.2)$.

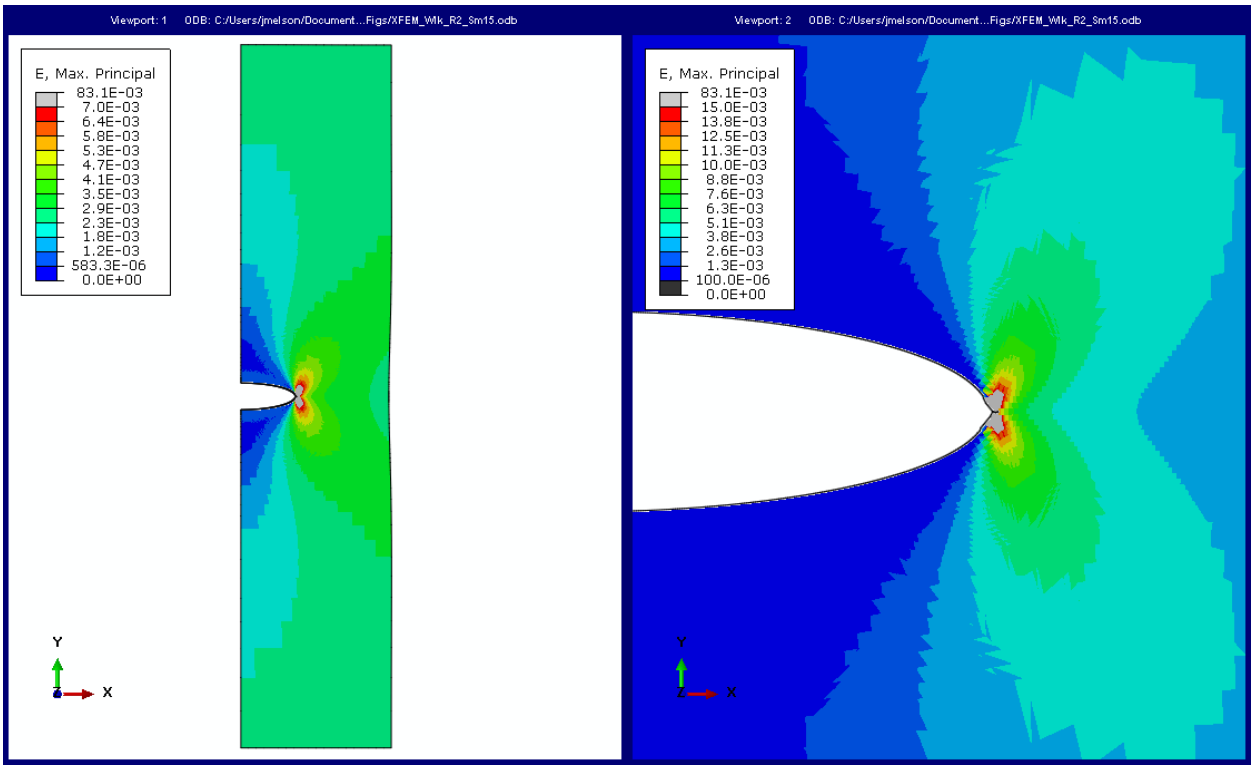

Fig. B.22: XFEM-PN, deformation mode u_y with $a = 2.38\,\text{in.}$ $(R = 0.2)$.

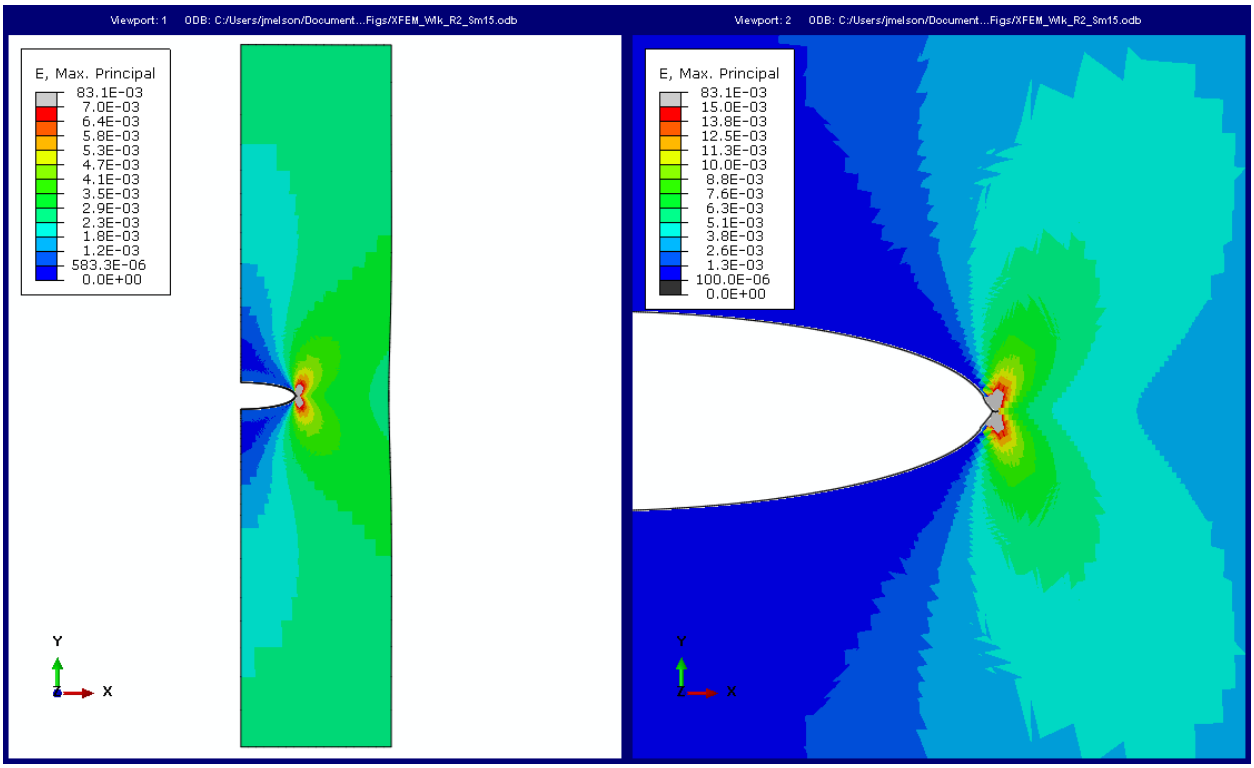

Fig. B.23: XFEM-PN, principal strain ϵ_{mAX} with $a = 2.38\,\text{in.}$ $(R = 0.2)$.

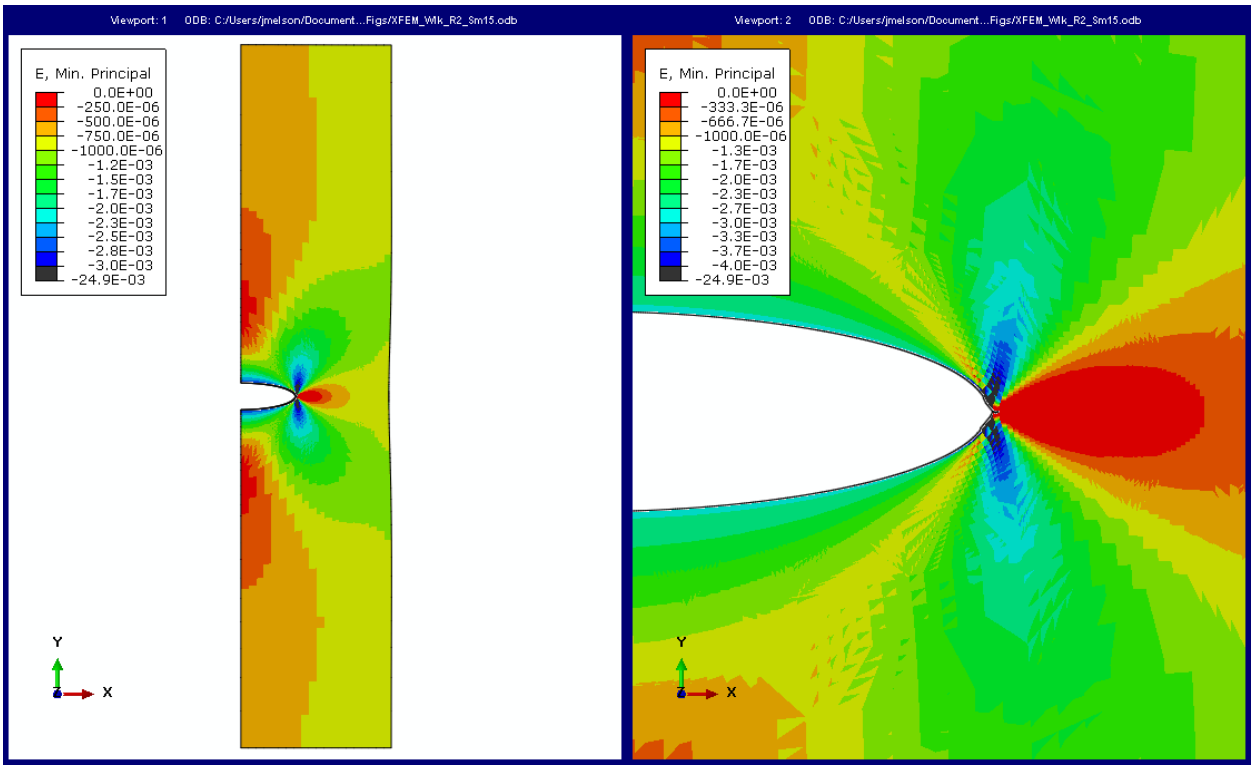

Fig. B.24: XFEM-PN, principal strain ϵ_{min} with $a = 2.38\,\text{in.}$ $(R = 0.2)$.

Appendix C

Convergence Analysis for Finite Element Models

The following figures show additional convergence analysis results for each of the finite element models. For the Finite Element with Quarter Points (FEQP) analyses, Figs. C.1-C.3 depict the normalized ΔK_I variation at contour integrals within the focused mesh region, r_e. Radial distances to the contour, r, are measured from the crack tip. Root Mean Square Error (RMSE) of all contours in each mesh are listed in Table C.1 with errors relative to the analytical solution.

Typical results of the stabilization ratios R_0, R_n, U_0, and U_n for both VCCT and XFEM-PN analyses are shown in Fig. C.4. Stabilized cycle ratios were plotted versus the half crack length, a, normalized with half width of the plate, b, for each of the mesh density. All ratios were significantly smaller than the default value of 0.005, except R_0 in the VCCT analysis. Fig. C.5 shows that R_0 increases with crack length and mesh density, requiring a user defined critical value for CR_0 as noted in Section 4.5. Spikes seen in Meshes #3 and #4 were caused when the analyses were terminated due to single cycle increments, also noted in Section 4.5. After termination, the analyses were restarted and allowed to run until the critical energy release rate was reached.

To build figures like Figs. C.4 and C.5, Fourier terms required to calculate the stabilization ratios were extracted from Abaqus' message (MSG) files for each iteration within each cycle. Fig. C.6 shows how R_0 converges to a stabilized solution for four arbitrarily chosen cycles in the Mesh #1 VCCT analysis. This behavior was typical of all stabilization ratios for both VCCT and XFEM-PN analyses. Once the cycle had stabilized, the final ratio values were selected and plotted in Figs. C.4 and C.5. For example, the final data point is marked for cycle number 2253.

Table C.1: RMSE of ΔK_I Contour Values in Convergence Analysis.

Mesh	ΔK_I Contour Value RMSE Relative to Analytical Results (%)		
	$a = 0.1$ in. (2.54 mm)	$a = 1.0$ in. (25.4 mm)	$a = 3.0$ in. (76.2 mm)
1	0.20	0.33	0.97
2	0.13	0.31	0.96
3	0.10	0.31	0.96

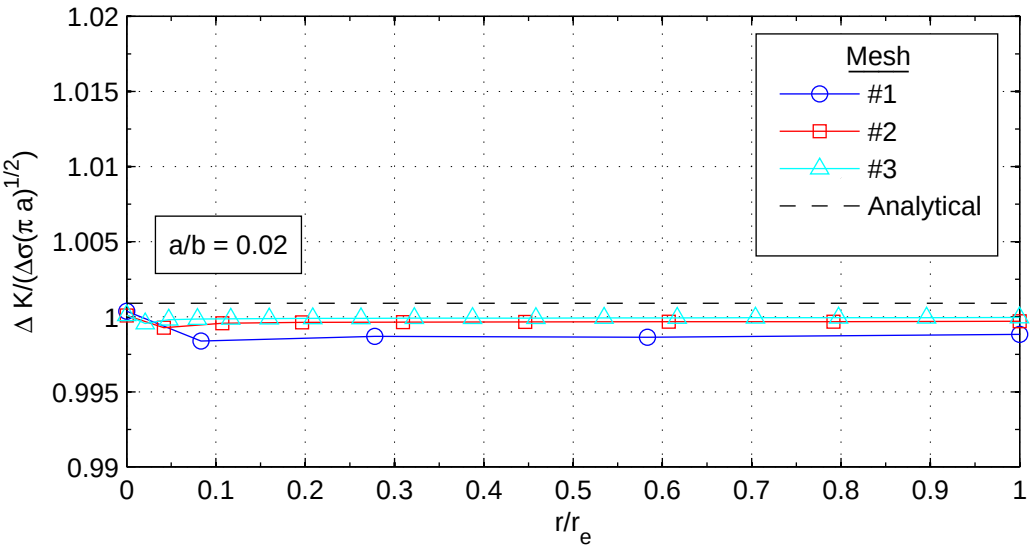

Fig. C.1: FEQP normalized ΔK_I variation with contour intervals: $a = 0.1$ in. (2.54 mm)

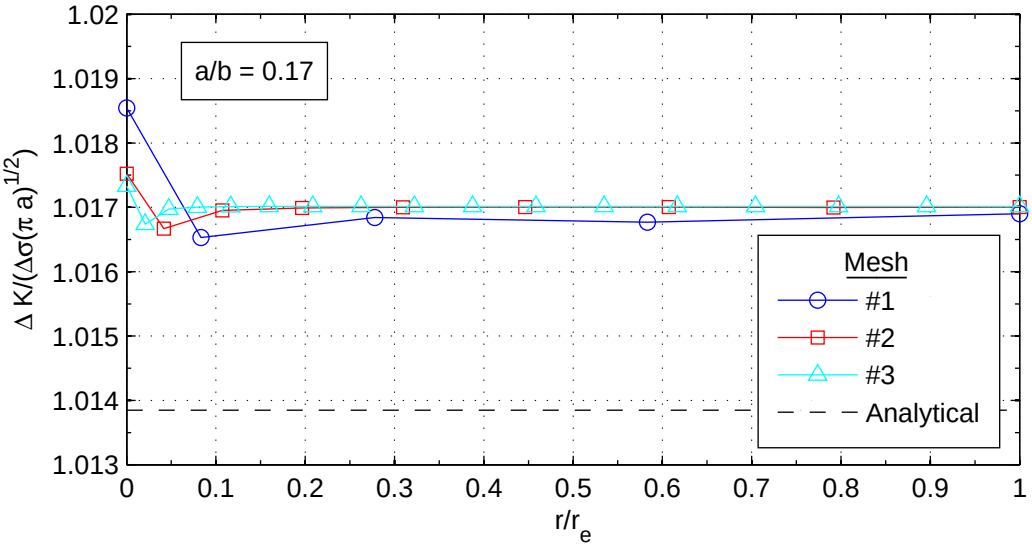

Fig. C.2: FEQP normalized ΔK_I variation with contour intervals: $a = 1.0$ in. (25.4 mm)

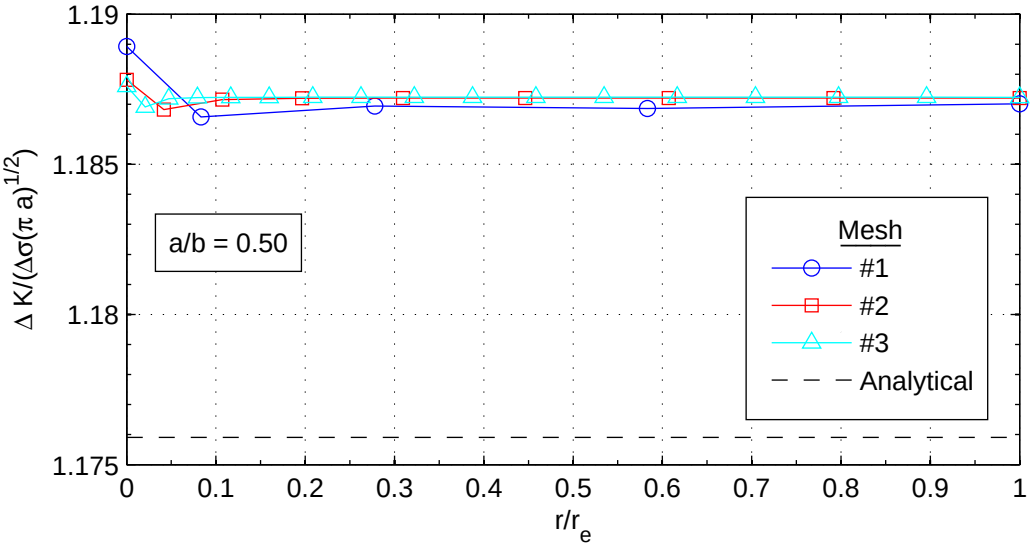

Fig. C.3: FEQP normalized ΔK_I variation with contour intervals: $a = 3.0\,\text{in.}$ (76.2 mm).

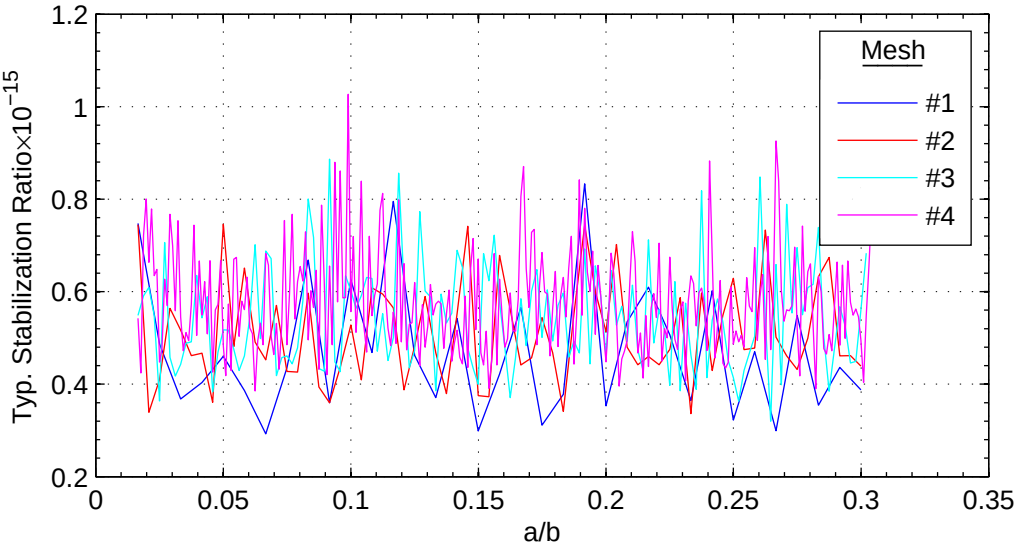

Fig. C.4: Typical stabilization ratio for VCCT and XFEM-PN direct cyclic analyses during fatigue crack propagation.

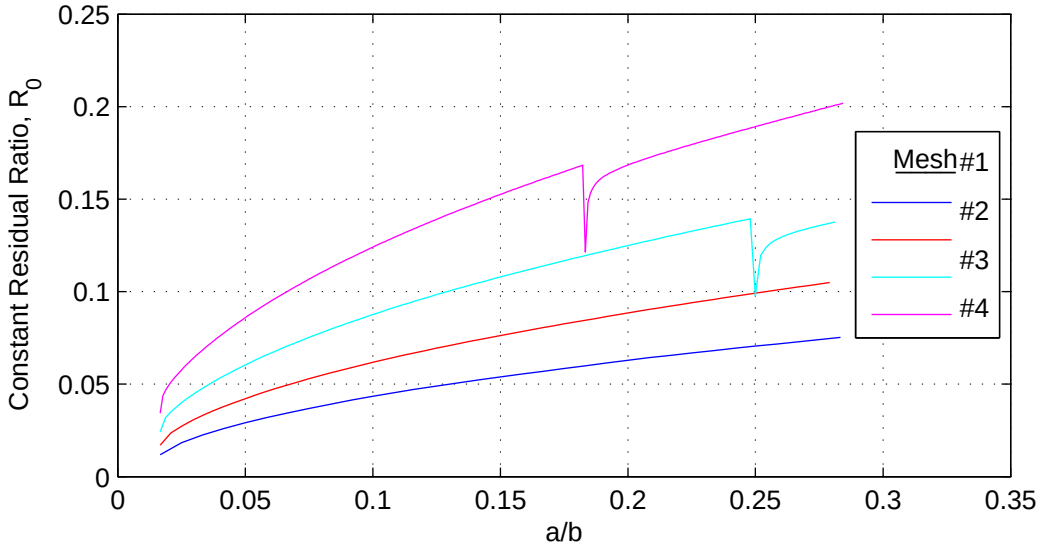

Fig. C.5: VCCT constant residual stabilization ratio, R_0, during fatigue crack propagation.

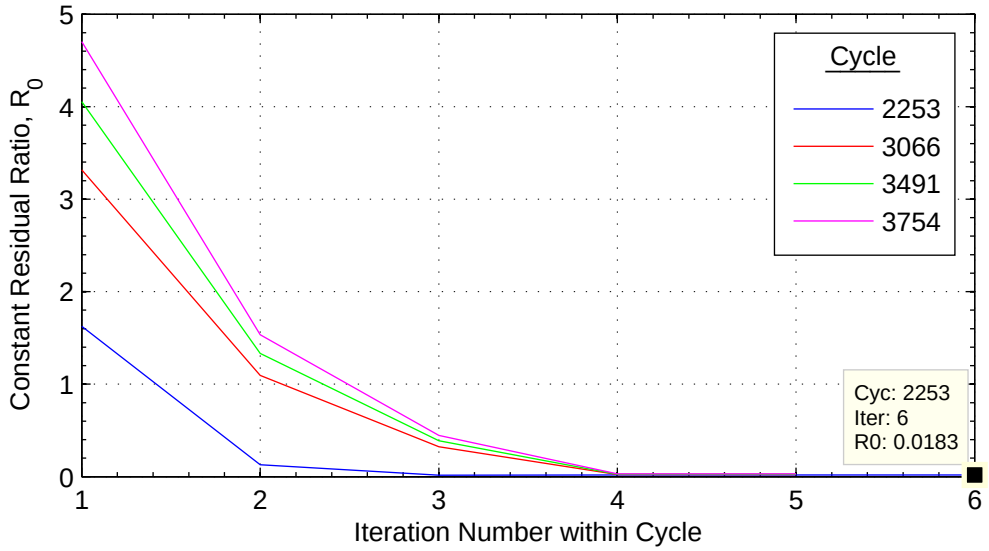

Fig. C.6: VCCT constant residual stabilization ratio, R_0, during four cycle iterations with R_0 labeled once cycle 2253 has been stabilized.

Appendix D

Monte Carlo Analysis Results

Distributions for the correlated Walker constants C_0, m, and γ of Eq. (2.11) are provided in Figs. D.1-D.3 for the 10,000 samples used in the analytical Monte Carlo analysis. Figs. D.4 and D.5 show the distributions of the 500 samples of c_3 and c_4 used in Abaqus' interpretation of the Paris law, Eq. (4.19). A normal Probability Density Function (PDF) is superimposed on the m, γ, and c_4 histograms, and a log-normal PDF is superimposed on the C_0 and c_3 histograms. Each normal PDF was generated based on the mean, μ, and standard deviation, σ, of the data. All PDFs fit well as expected since the correlated random numbers were generated from a multivariate normal distribution in log-log space.

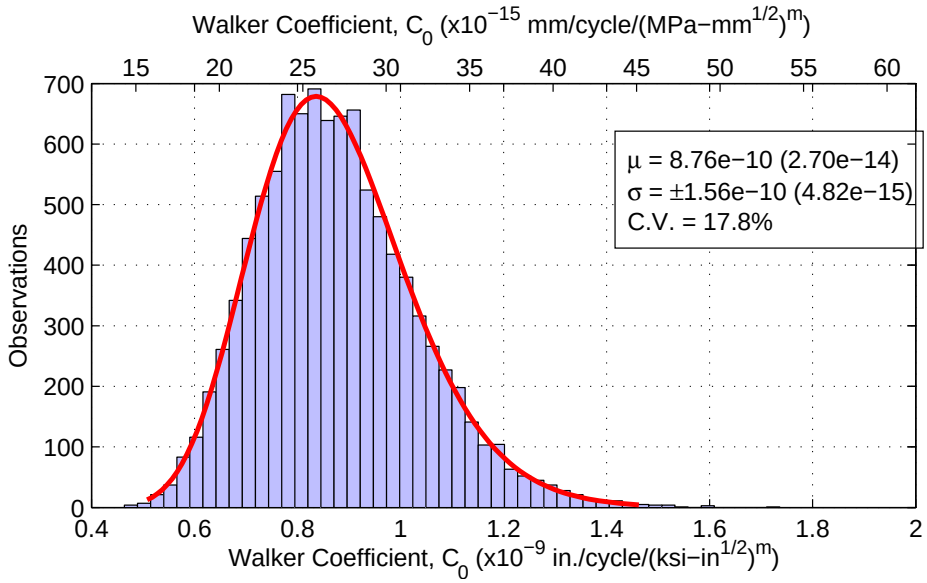

Fig. D.1: Distribution of Walker constant, C_0, with 10,000 samples and a log-normal PDF.

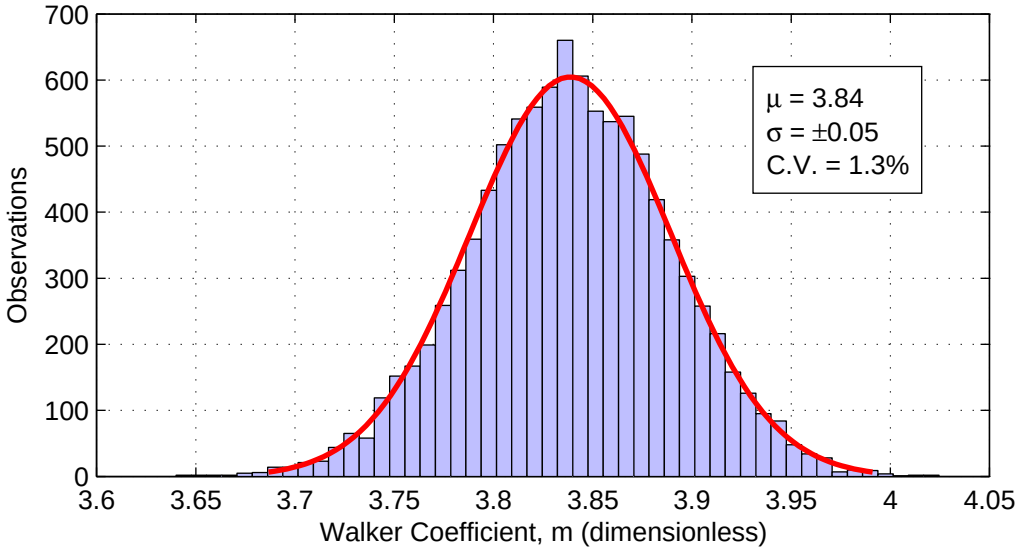

Fig. D.2: Distribution of Walker constant, m, with 10,000 samples and a normal PDF.

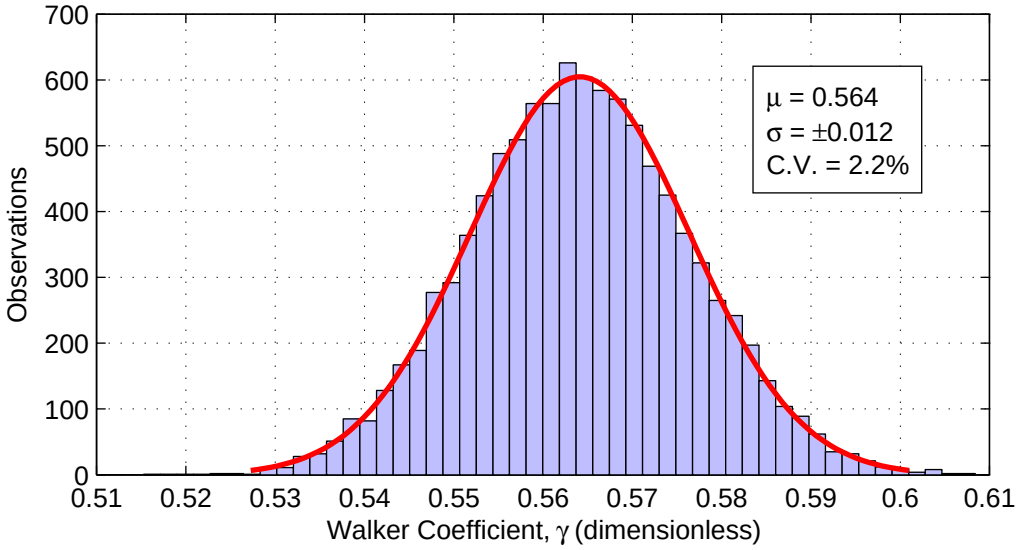

Fig. D.3: Distribution of Walker constant, γ, with 10,000 samples and a normal PDF.

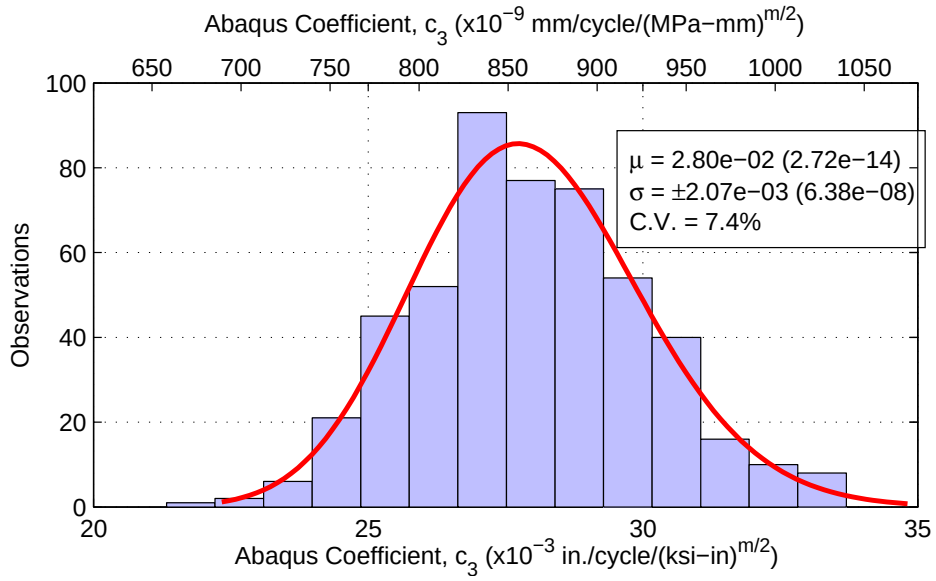

Fig. D.4: Distribution of Walker constant, c_3, with 500 samples and a log-normal PDF.

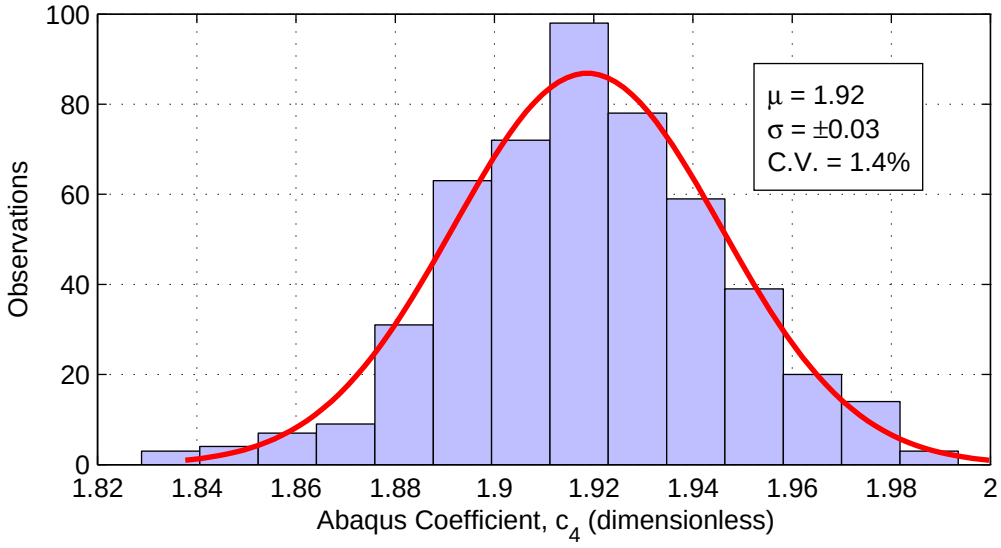

Fig. D.5: Distribution of Walker constant, c_4, with 500 samples and a normal PDF.

Appendix E

Compact Specimen Analysis Results

This appendix provides data used to model the three-dimensional crack growth in a compact specimen. Table E.1 lists material data taken from Orringer et al. [78], Feddersen and Broek [79], and supplemented by medium carbon steel properties found on www.matweb.com [74] for those properties not found in the literature. Jeong, Tang, Orringer, and Perlman [80] used a form of the Walker equation shown in Eq. (E.1). By comparing Jeong et al.'s equation with the Walker equation, the Walker constant γ was solved for in terms of the empirical constants m and Q. Loading and boundary conditions on the reference points representing the loading pins are listed in Table E.2. The loading applied to the top reference pin was reduced by half since only half of the compact specimen was modeled. Required dimensions are listed in Fig. E.1.

$$\frac{da}{dN} = C_0 \frac{(\Delta K)^m}{(1-R)^Q} \tag{E.1}$$

$$\gamma = 1 - \frac{Q}{m} \tag{E.2}$$

The resulting deformation modes, u_x, u_y, and u_z are presented in Figs. E.2, E.3, and E.4, respectively. The maximum and minimum principal strains, ϵ_{max} and ϵ_{min} are provided since principal values are independent of coordinate system. Figs. E.5 and E.7 show ϵ_{max} and ϵ_{min}, respectively, for the entire compact specimen. Figs. E.6 and E.8 show ϵ_{max} and ϵ_{min}, respectively, with the top portion of the compact specimen removed and zoomed in on the crack tip. These last two figures illustrate the through-thickness variation of strain at the crack tip. The jump in red contours in Fig. E.6 illustrate a non-uniform crack tip extension.

Table E.1: Rail Steel Material Properties

Rail Steel Material Properties		
E	29 600 ksi	[79]
ν	0.293	[8]
C_0	$1 \times 10^{-11} \frac{\text{in./cycle}}{\left(\text{ksi}\sqrt{\text{in.}}\right)^{m/2}}$	[80]
m	4	[80]
Q	1.63	[80]
γ	0.592	(1)
K_c	73.7 ksi	[74]
σ_{ult}	133 ksi	[78]
c_3	5.6×10^{-3}	(2)
c_4	2	(2)

(1) Calculated from Eq. (E.2)
(2) Calculated from Eq. (4.19)

Table E.2: Compact Specimen Loading and Boundary Conditions

Location	Boundary Conditions	Loading Conditions
Top Pin Reference Point	$u_x = 0$	$\Delta P = 2.5$ kips
Bottom Pin Reference Point	$u_x = u_y = 0$	-
Symmetry Plane	$u_z = 0$	-

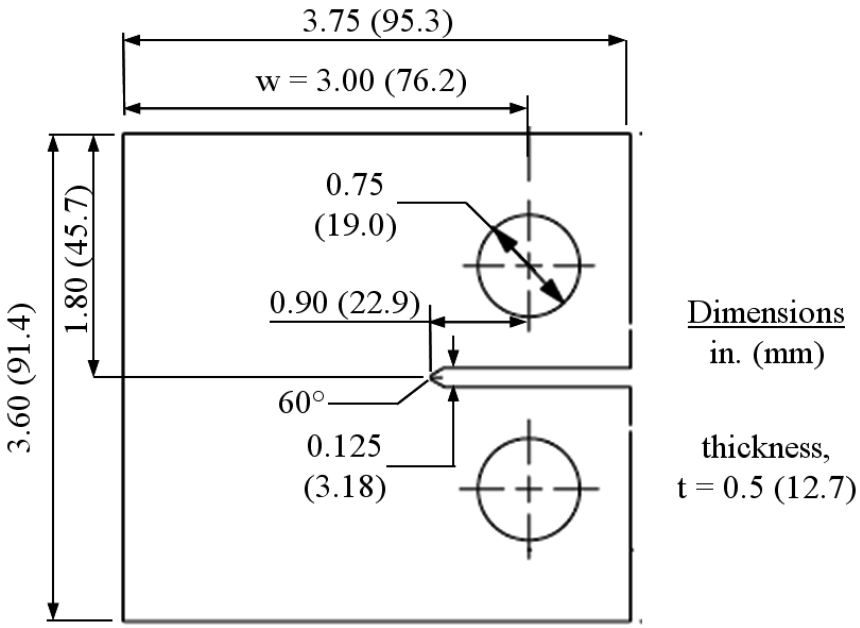

Fig. E.1: Compact specimen dimensions.

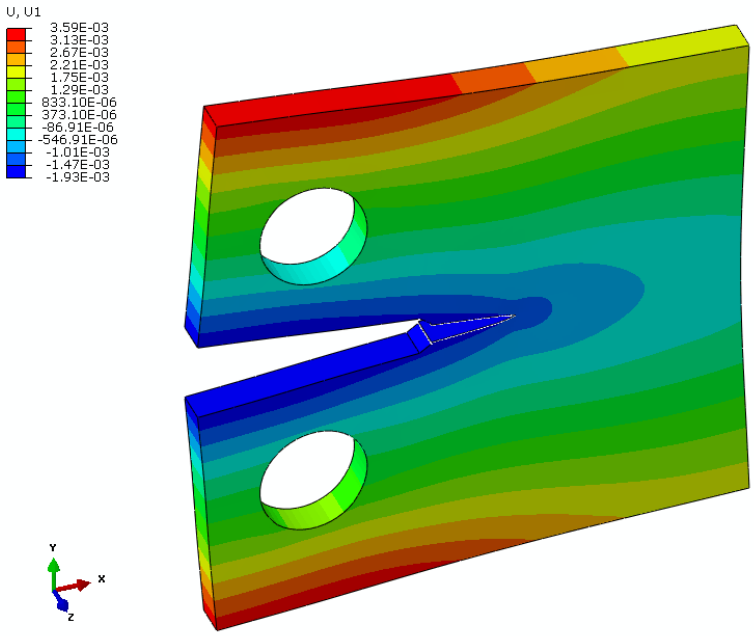

Fig. E.2: Compact specimen deformation mode u_x with $a = 1.46\,\text{in.}$ $(R = 0.2)$.

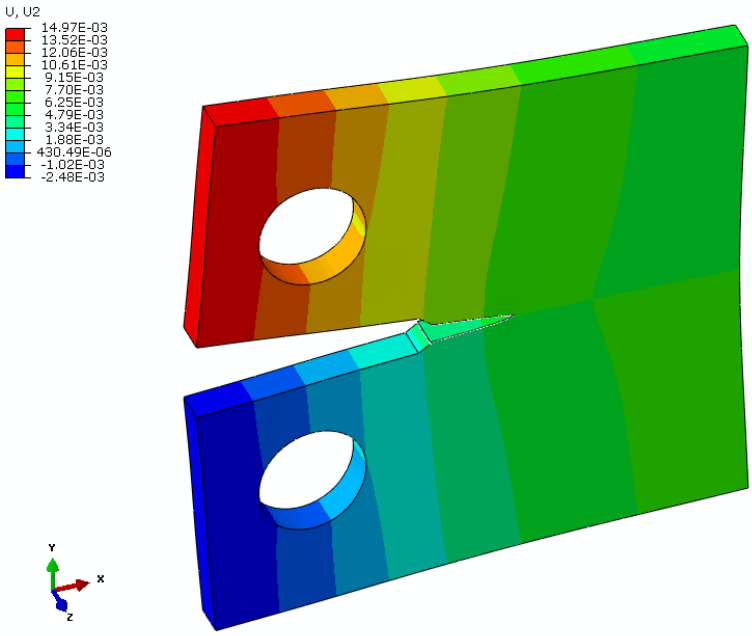

Fig. E.3: Compact specimen deformation mode u_y with $a = 1.46\,\text{in.}$ $(R = 0.2)$.

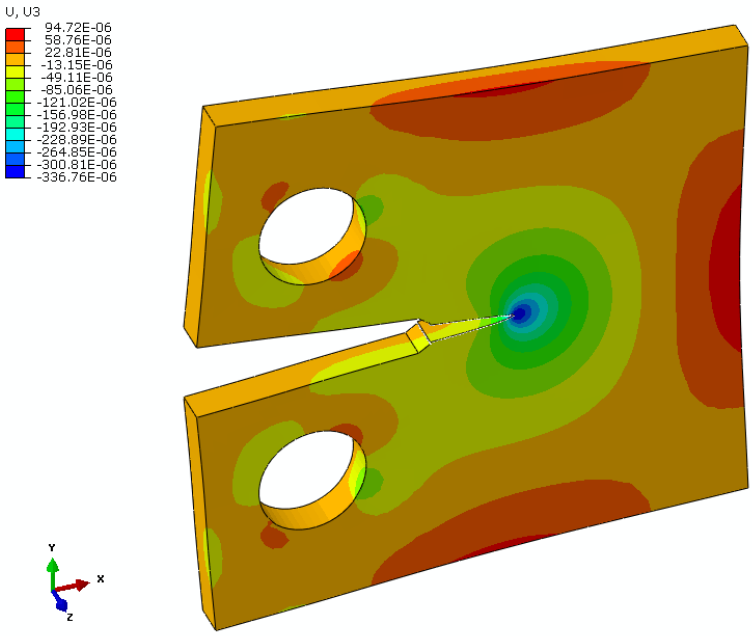

Fig. E.4: Compact specimen deformation mode u_z with $a = 1.46\,\text{in.}$ $(R = 0.2)$.

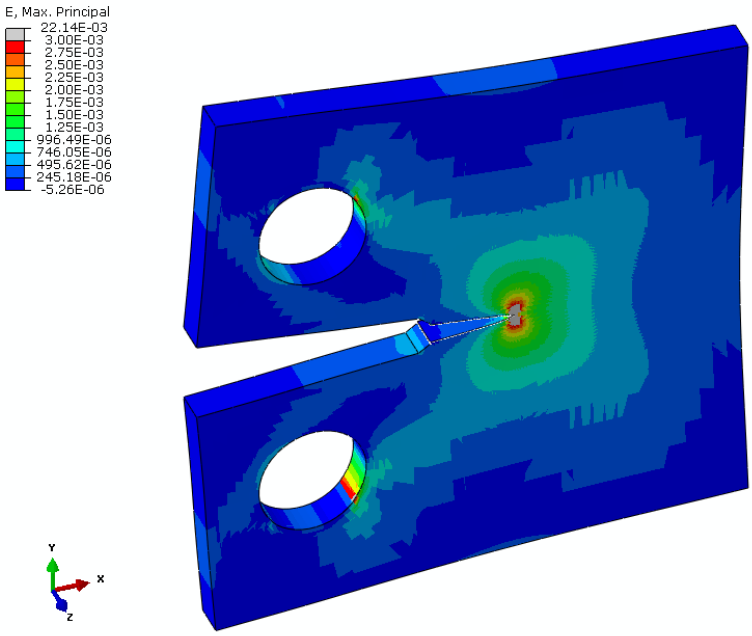

Fig. E.5: Compact specimen principal strain ϵ_{max} with $a = 1.46\,\text{in.}$ ($R = 0.2$).

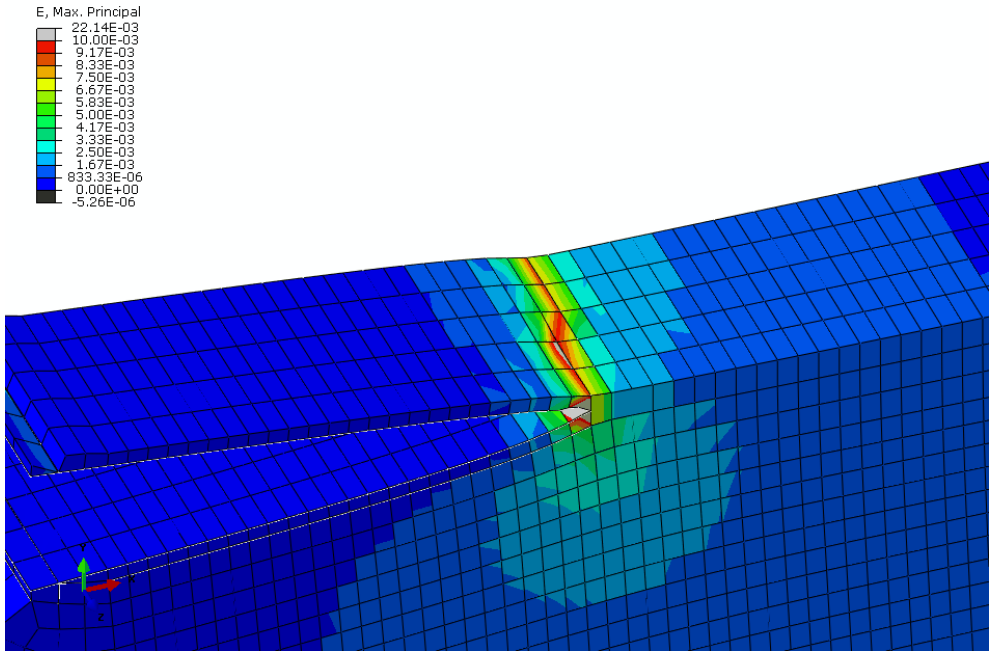

Fig. E.6: Crack tip of compact specimen principal strain ϵ_{max} with $a = 1.46\,\text{in.}$ ($R = 0.2$).

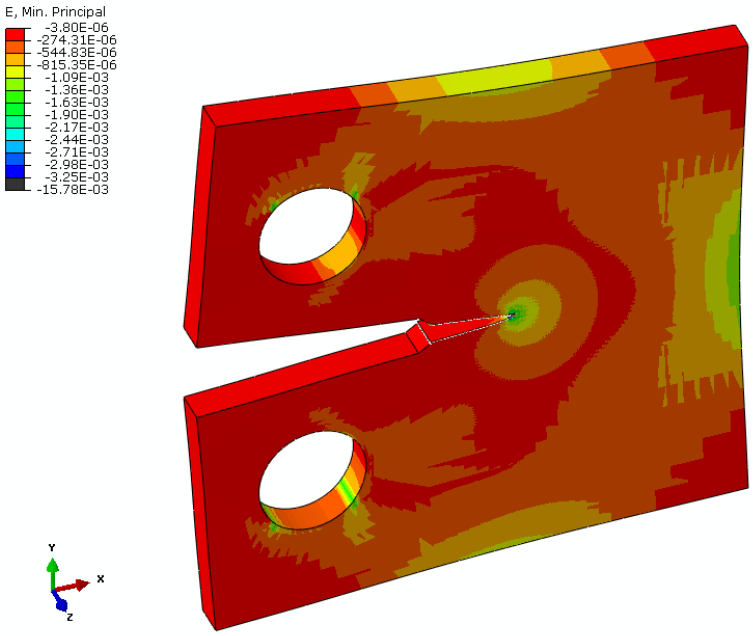

Fig. E.7: Compact specimen principal strain ϵ_{min} with $a = 1.46\,\text{in.}$ ($R = 0.2$).

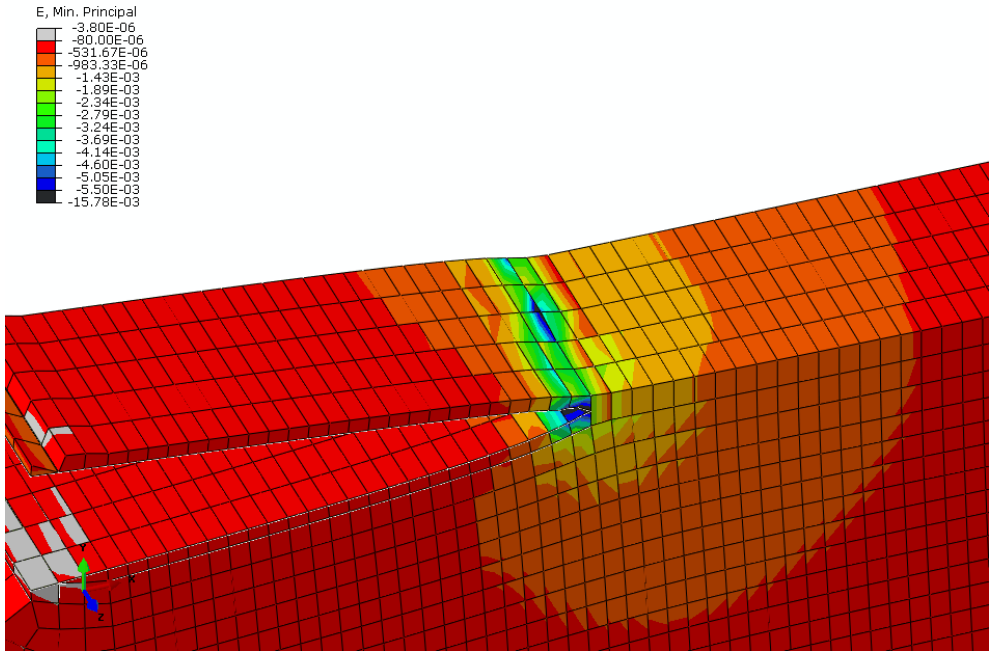

Fig. E.8: Crack tip of compact specimen principal strain ϵ_{min} with $a = 1.46\,\text{in.}$ ($R = 0.2$).

Appendix F

MATLAB Code

The following code was developed in MATLAB verions R2013a and R2013b.

Function: CenterCrackF.m

```
function [F] = CenterCrackF(a, b)
% Calculates dimensionless geometry factor, F, for a center
% craked plate based on equations from Figure 8.12 in Dowling pg
  327
%
% a: crack half length
% b: specimen half width
%
% Rev   Date        Name            Changes
% 00    05-29-12    Josh Melson     -Created
%
alpha = a/b;
F = (1 - 0.5*alpha + 0.326*alpha.^2)./sqrt(1 - alpha);
```

Function: CenterCrackFinalF.m

```
function [F, ac, alpha, fval, flag] = CenterCrackFinalF(Pmax, a, b,
   t, KIc)
% Created by Josh Melson  05-29-12, modified 08-01-12
% Function solves for the final crack length based on KIc of a
  center crack
% plate, equations from Figure 8.12, Dowling pg 327

% x(1): F       [dimless]
% x(2): ac      [length]
% x(3): alpha   [dimless]
%
% Rev   Date        Name            Changes
% 00    05-29-12    Josh Melson     -Created
% 01    08-01-12    Josh Melson     -Adjusted solving process
%
Smax = Pmax /(2*b*t);       % force/length^2

G = @(x) [  x(1) - (1 - 0.5*x(3) + 0.326*x(3)^2)/sqrt(1 - x(3)) ;
            KIc - x(1)*Smax*sqrt(pi*x(2)) ;
            x(3) - x(2)/b ];

x0 = [ 1.12; a; a/b ];

option = optimset('Display','off');

[x, fval, flag] = fsolve(G,x0,option);

F = x(1);
ac = x(2);       % length
alpha = x(3);
```

Function: dadN_Regression.m (Use with experimental data)

```
function [ai, dadN] = dadN_Regression(a, N, pt)
% Calculates da/dN using a quadratic fit to data points of a cycles
% vs crack length curve
%
% Input:    a: Crack Length
%           N: Cycles
%           pt: Total number of points to use
% Output:   ai: Crack length evaluated from regression curve at
  cycle Ni
%           dadN: Slope of regression curve at cycle Ni
%
% Rev   Date         Name            Changes
% 00    11-06-13     Josh Melson     -Created
%
%% Check for Errors
if nargin < 3;       error('Not enough inputs');
                       end
if nargin > 3;       error('Too many inputs');
                          end
if rem(pt,1) ~= 0;   error('Number of points must be an integer');
    end
if pt > 3;           warning('Suggested points less than 4');
         end
%% Initialize variables
ai   = zeros(length(a)-2*pt,1);
dadN = zeros(length(a)-2*pt,1);
if length(ai) > length(a); return; end
%% Adjust magnitude of N to be silimar to a for a better fit
Nord = order(N(2:end));
Ordr = ceil(mean(Nord));
Adj  = 10^Ordr;
Nadj = N/Adj;
%% Fit Points
for i = pt+1:length(N)-pt
    NT = Nadj(i-pt:i+pt);                                % Reduced set
      of N
    aT = a(i-pt:i+pt);                                   % Reduced set
      of a
    b  = polyfit(NT,aT,2);                               % Quad fit
    db = polyder(b);                                     % Derivative
      Quad fit
    ai(i-pt) = polyval(b, Nadj(i));                      % Crack length
      at Ni
    dadN(i-pt) = polyval(db, Nadj(i))/Adj;               % da/dN at Ni
end
```

Function: LifeIntegral_Walker.m (Use with Analytical model)

```
function [dNda] = LifeIntegral_Walker(a, b, deltaS, R, C0, m, gamma
  )
% Function used for integral of Walker fatigue crack growth model
% a: crack length, can be a vector
% b: specimen half width
% deltaS: stress range
% R: R-Ratio
% C0, m, gamma: Walker coefficients
% dNda: inverse of the da/dN for integration of N from ai to ac
%
% Rev   Date         Name            Changes
% 00    10-28-13     Josh Melson     -Created
%
F = CenterCrackF(a, b);
C1 = C0/(1 - R)^(m*(1 - gamma));

dNda = 1 ./ (C1*(F*deltaS.*sqrt(pi*a)).^m);
```

Function: MonteCarlo_Walker.m (Use with Analytical model)

```
function [N, C0, m, gam] = MonteCarlo_Walker(coeff, covar, num, ai,
   ac, b, dS, R)
% Monte Carlo simulation using Walker coefficients and covariance
  matrix
%
% coeff: log space fit coefficients vector
% covar: log space covariance matrix
% num:   number of observations
%
%  Fit based on Walker equation
%  da/dN = C0/(1-R)^(m*(1-gamma))*DeltaK^m
%
%  Linearize equation
%  log(da/dN) = log(C0) + m*log(DeltaK) - m*(1-gamma)*log(1-R)
%           Y = b + m1*X1 + m2*X2
%
%  <<log base 10>>
%
%  Input to fit:
%      X1 : log(delK)
%      X2 : log(1-R)
%      Y  : log(dadN)
%  Output from fit into Monte Carlo:
%      coeff(1): b, intercept
%      coeff(2): m1, slope1
%      coeff(3): m2, slope2
%  Conversion to untransformed space
%      C0 = 10^b
%      m  = m1
%      gamma = 1 + m2/m1
%
% Rev   Date         Name            Changes
% 00    03-04-14     Josh Melson     -Created
%

% covar = diag(diag(covar));        % Use for uncorrelated results
rndcoeff = mvnrnd(coeff,covar,num); % Correlated random numbers

C0  = 10.^rndcoeff(:,1);
m   = rndcoeff(:,2);
gam = 1 + rndcoeff(:,3)./rndcoeff(:,2);
N   = zeros(size(C0));

for j = 1:num
    Nfun = @(a) LifeIntegral_Walker(a, b, dS, R, C0(j), m(j), gam(j
      ));
```

```
    N(j) = integral(Nfun, ai, ac);
end
```

Script: Abaqus_CrackSizeAnalysis.m (Use with FEQP)

```
% Creates data file for Python to read and Abaqus to execute
% Modified for Center Crack Plate, Hudson 1996, 7075-T6
%
% Rev   Date        Name            Changes
% 00    01-15-14    Josh Melson     -Created
%
close all; clear all; clc;
format shortG

%% Load Data
dataPath = 'C:\Users\jmelson\Documents\Thesis\';
load(strcat(dataPath,'Al_7075T6_Hudson'))
load(strcat(dataPath,'Al_7075T6_Hudson_Dimensions'))

%% Define Models to Analyze
Rfld = {  'R0';   'R2';   'R5';   'R7'};
Sfld = {'Sm15'; 'Sm15'; 'Sm15'; 'Sm15'};
Meth = {'Walker'};

%% Set Number of Steps
j = 20;                 % Must be even integer

%% Define CPU Properties
Memory = '18000mb';
CPUs   = '4';

%% Analyze in Abaqus
cmd1Val = 'abaqus python CenterCrack_CFE_AdjustValues.py %s %s %s';
for m = 1:length(Rfld)
    S = Al.(Rfld{m}).(Sfld{m});                 % Reduce calling
      structure
    S.CFE = struct;

    r  = (S.ac/ai)^(1/j);
    jj = 1:j+1;
    S.CFE.a = ai*r.^(jj-1);
    S.CFE.a = S.CFE.a';

    S.CFE.K1  = zeros(length(S.CFE.a), 1);
    S.CFE.K2  = zeros(length(S.CFE.a), 1);
    S.CFE.JKs = zeros(length(S.CFE.a), 1);
    S.CFE.J   = zeros(length(S.CFE.a), 1);
    Rvalue    = Rfld{m};                        % Current R
    pLoad     = strcat('-',num2str(S.dS));      % Current Load

    for n = 1:length(S.CFE.a)
```

```
        aLen    = num2str(S.CFE.a(n));              % Current Length

        if length(aLen) < 5;     q = length(aLen);
        else q = 5; end

        jobName = strcat('CFE_',Rvalue,'_S',num2str(round(S.dS)),'
          _C',aLen(1),aLen(3:q));
        cmd1    = sprintf(cmd1Val, Rvalue, pLoad, aLen);
        cmd2    = sprintf('abaqus cae noGUI=
          CenterCrack_CFE_AdjustCrack.py');

        s1 = system(cmd1);                          % Adjust Python
          file for new crack
        s2 = system(cmd2);                          % Adjust Abaqus
          file for new crack
        s3 = RunAbaqus(jobName, CPUs, Memory);      % Run Abaqus
        if s3 == 0
            matName = grabDataFunc(jobName);        % If executed
              correctly, grab data
        end

        res = load(matName);                        % Load Results
        S.CFE.K1(n)  = mean(res.K1(3:end));         % Mean excluding
          contours 1 and 2
        S.CFE.K2(n)  = mean(res.K2(3:end));
        S.CFE.JKs(n) = mean(res.JKs(3:end));
        S.CFE.J(n)   = mean(res.J(3:end));

        clear res

    end

    Al.(Rfld{m}).(Sfld{m}) = S;                     % Replace calling
      structure
end
AbaqusFileDelete;
save('Al_7075T6_Hudson_CFE.mat','Al')
```

Script: Abaqus_XFEM_terminate.m (Use with XFEM-PN)

```
% Terminates Abaqus when cyclic analysis increases by a single
  cycle
% Must be run in a separate instance of Matlab while Abaqus is
  running.
% Modified for Center Crack Plate, Hudson 1996, 7075-T6
%
% Rev    Date         Name             Changes
% 00     02-17-14     Josh Melson      -Created
%
close all; clear all; clc;
format shortG

%% Define Variables and Strings
psTime = 10*60;                         % Pause time in seconds
ckTime = 3*60;
cmdSta = 'abaqus python CenterCrack_XFEM_StatusCheck.py %s';
cmdTer = 'abaqus job=%s terminate';

%% Look for .lck file
endFile = [];
while isempty(endFile) == 1
    lckFile = [];
    while isempty(lckFile) == 1
        lckFile = dir('*.lck');
    end

    jobName = lckFile.name(:,1:end-4);
    fprintf('Checking job = %s\n', jobName)

    staFile = [];
    while isempty(staFile) == 1
        staFile = dir(strcat(jobName, '.sta'));
    end

    hltFile = [];
    while isempty(hltFile) == 1
        pause(psTime)
        s1 = system(sprintf(cmdSta,jobName));
        hltFile = dir(strcat(jobName,'.hlt'));
    end

    fprintf('Terminating job = %s\n', jobName)
    s2 = system(sprintf(cmdTer,jobName));

    pause(ckTime)
    endFile = dir('JobComplete.end');
```

```
    if isempty(endFile) == 1
        fprintf('Continuing Abaqus Termination Process\n\n')
    end
end
fprintf('Program complete\n')
delete('JobComplete.end');
```

Script: Abaqus_XFEM_MonteCarloAnalysis.m (Use with XFEM-PN)

```
% Creats new Input file, executes in Abaqus, and terminates at
  single
% cycle steps. Modified for Center Crack Plate, Hudson 1996, 7075-
  T6
%
% Rev    Date         Name             Changes
% 00     03-11-14     Josh Melson      -Created
%
close all; clear all; clc;
format shortG

%% Load Data
dataPath = 'C:\Users\jmelson\Documents\Thesis\';
M  = load(strcat(dataPath,'Al_7075T6_Hudson_MatProp'));
     load(strcat(dataPath,'Al_7075T6_Hudson'));
MC = load('Al_7075T6_Hudson_MonteCarlo05base');

%% Define Models to Analyze
Rfld = 'R2';
Sfld = 'Sm15';
Meth = 'Walker';
S = Al.(Rfld).(Sfld);
W = M.Walker;
dGc =  (M.Kc^2/M.E)*(1 - S.R^2);

%% Define CPU Properties
Memory = '18000mb';
CPUs   = '4';

%% Analyze in Abaqus
fprintf('****** RUN TERMINATION CODE IN PARALLEL ******\n')
fprintf(' Starting analysis of %d samples\n', length(MC.C0))
cmd1Str = 'abaqus python CenterCrack_XFEM_MC_AdjustValues.py %.6e
  %.5f %s';
A    = struct;
for m = 1:length(MC.C0)
    [A.c3, A.c4] = AbaqusGrowthConstants( ...
        Meth, S.R, MC.C0(m), MC.m(m), M.E, MC.gam(m));

    MC.c3(m,1) = A.c3;
    MC.c4(m,1) = A.c4;

    if m < 10
        runName = sprintf('00%d',m);    % Define file name
    else
        runName = sprintf('0%d',m);
```

```
    end

    cmd1 = sprintf(cmd1Str, A.c3, A.c4, runName);
    s1   = system(cmd1);                % Adjust Inp file for new
     load

    jobName = (sprintf('XFEM_R2_MC%s',runName));
    fprintf('\n\n***** Starting Run %d (%s) *****\n', m, jobName)
    s2 = RunAbaqus(jobName, CPUs, Memory);  % Run Abaqus

    matFile = grabODBdata_XFEM(jobName);

    res = load(matFile);
    res.da = res.C(:,1);
    res.dG = res.G1(:,1);
    res.c3 = A.c3;
    res.c4 = A.c4;
    res = rmfield(res, {'C';'G1'} );

    [res.N, res.a] = LifeEstimate_XFEM_Walker(res.da, res.dG, A.c3,
       A.c4, 'Poly');

    indx1  = find(res.dG <= dGc, 1, 'last');    % Reduce data to
     dGc
    if isempty(indx1) == 0
        MC.frm(m,1) = res.frm(indx1);
        indx2 = find(res.a == res.da(indx1));
        if isempty(indx2) == 1
            indx2 = find(res.a == res.da(indx1+1));
        end
        MC.N(m,1)  = res.N(indx2);
    else
        MC.frm(m,1) = NaN;
        MC.N(m,1)   = NaN;
    end

    MC.mdl.(sprintf('MC%s',runName)) = res;
    clear res
    jobKeep = dir('*.odb');
    AbaqusFileDelete(jobKeep,{'inp'});
end
save('Al_7075T6_Hudson_MonteCarlo.mat', '-struct', 'MC')
save('JobComplete.end')
```

Function: LifeEstimate_XFEM_Walker.m (Use with XFEM-PN, Integrate Polynomial Method)

```
function [N, a, dN] = LifeEstimate_XFEM_Walker(ai, dG, c3, c4,
  Method)
% Estimate life based on integrated polynomial or trapezoidal
  method
%
% Rev    Date         Name             Changes
% 00     02-22-14     Josh Melson      -Created
% 01     03-14-14     Josh Melson      -Add last increment w even
  numbers
%

if nargin < 5;  Method = 'Poly';    end

if max(strcmp(Method,{'Poly';'Trapz'})) == 0
    error('Methods available are: Poly or Trapz')
end

if rem(length(ai),2) == 0
    warning('Vector length of a and dK must be even integers')
    aLast = 3;
else
    aLast = 2;
end

if length(ai) ~= length(dG)
    error('Vectors a and dK must be the same length')
end

dadN_Abaq = @(delK) c3*delK.^c4;          % da/dN function
N = 0;                                    % Cycles
a = ai(1);                                % at crack length
j = 2;
for n = 1:2:length(ai)-aLast
    y = 1./dadN_Abaq(dG(n:n+2));          % y vector
    x = ai(n:n+2);                        % x vector

    switch Method
        case 'Trapz'
            dN(j-1) = trapz(x,y);         % trapezoidal integration
        case 'Poly'
            pf = polyfit(x,y,2);          % polynomial fit
            pfint = polyint(pf);          % polynomial integration
            dN(j-1) = polyval(pfint,x(end)) - polyval(pfint,x(1));
    end
```

```
    N(j) = N(j-1) + dN(j-1);              % Sum cycles
    a(j) = x(end);
    j = j + 1;
end

% Solve for last increment if even number comes in
if aLast == 3
    y = 1./dadN_Abaq(dG(end-1:end));
    x = ai(end-1:end);
    dN = trapz(x,y);
    N(end+1) = N(end) + dN;
    a(end+1) = x(end);
end
```

Appendix G

Python Code

The following code was developed in Python version 2.7.

CenterCrack_CFE_AdjustValues.py (Use with FEQP)

```
## Opens CenterCrack_CFE_AdjustCrack.py and replaces values for
  crack analysis
#
# Rev    Date         Name             Changes
# 00     01-15-14     Josh Melson      -Created
#
# Modules
import os, sys
#-----------------------------
# Read in Variable names
Rvalue = sys.argv[1]
pLoad  = sys.argv[2]
aLen   = sys.argv[3]
#-----------------------------
#Check for 3 input arguments
checkarg = len(sys.argv) - 1
numarg   = 3
if checkarg < numarg:
    print "Fail"
    sys.exit("CenterCrack_CFE_AdjustCrack: Not enough input
      arguments")

elif checkarg > numarg:
    print "Fail"
    sys.exit("CenterCrack_CFE_AdjustCrack: Too many input arguments
      ")

#-----------------------------
```

```
# Names to search for
rName = 'Rvalue = '
pName = 'pLoad  = '
aName = 'aLen   = '
#-----------------------------
# Replace Names
rNameReplace = rName+'\''+Rvalue+'\''
pNameReplace = pName+pLoad
aNameReplace = aName+aLen
#-----------------------------
# Read File and Replace
pyFile = open('CenterCrack_CFE_AdjustCrack.py').read()
newLine = '\n'.join(rNameReplace if line.startswith(rName) else
 line for line in pyFile.splitlines())
open('CenterCrack_CFE_AdjustCrack.py', 'w').write(newLine)

pyFile = open('CenterCrack_CFE_AdjustCrack.py').read()
newLine = '\n'.join(pNameReplace if line.startswith(pName) else
 line for line in pyFile.splitlines())
open('CenterCrack_CFE_AdjustCrack.py', 'w').write(newLine)

pyFile = open('CenterCrack_CFE_AdjustCrack.py').read()
newLine = '\n'.join(aNameReplace if line.startswith(aName) else
 line for line in pyFile.splitlines())
open('CenterCrack_CFE_AdjustCrack.py', 'w').write(newLine)
```

CenterCrack_CFE_AdjustCrack.py (Use with FEQP)

```
# Adjust crack in CAE file
#
# Rev    Date         Name             Changes
# 00     01-15-14     Josh Melson      -Created
#
# Modules
from abaqus import *
from abaqusConstants import *
from caeModules import *
import mesh, sys

# Crack Lengths and Loading, Copied from Matlab
aLen   = 3.3386
pLoad  = -6
Rvalue = 'R7'

# Variable Names
fileName = 'CenterCrack_CFE'
mdlShort = 'CFE_short'
mdlLong  = 'CFE_long'
partName = 'Plate'
instName = 'Plate-1'
partionName = 'PartitionCrack'
loadName  = 'StressLoad'
dimCrack  = 'CrackLength'
dimRadius = 'EnrichRadius'
dimBuffer = 'BufferTop'
jobList   = []

# Update Abaqus Model
mdb = openMdb(fileName)

if aLen <= 0.8:
        mdlName = mdlShort
else:
        mdlName = mdlLong

p = mdb.models[mdlName].parts[partName]
s = p.features[partionName].sketch
mdb.models[mdlName].ConstrainedSketch(name='__edit__', objectToCopy
  =s)
s1 = mdb.models[mdlName].sketches['__edit__']
g, v, d, n = s1.geometry, s1.vertices, s1.dimensions, s1.
  constraints
s1.setPrimaryObject(option=SUPERIMPOSE)
```

```
p.projectReferencesOntoSketch(sketch=s1, upToFeature=p.features[
  partionName], filter=COPLANAR_EDGES)
s = mdb.models[mdlName].sketches['__edit__']

cBuffer = []
cRadius = []

if aLen <= 0.18:
        cBuffer = 0.18
        cRadius = 0.265625*aLen + 0.0534375
elif aLen > 0.18 and aLen <= 0.8:
        cBuffer = aLen
        cRadius = cBuffer*0.5625
else:
        cBuffer = 0.8
        cRadius = 0.45

s.parameters[dimCrack].setValues(expression=str(aLen))
s.parameters[dimBuffer].setValues(expression=str(cBuffer))
s.parameters[dimRadius].setValues(expression=str(cRadius))

s1.unsetPrimaryObject()
p = mdb.models[mdlName].parts[partName]
p.features[partionName].setValues(sketch=s1)
del mdb.models[mdlName].sketches['__edit__']
p = mdb.models[mdlName].parts[partName]
p.regenerate()

a = mdb.models[mdlName].rootAssembly
a.regenerate()
partInstances = (a.instances[instName], )
a.deleteMesh(regions=partInstances)
e1 = a.instances[instName].edges

seedRight = []
seedLeft  = []

if aLen < 0.6:
        seedRight = int( (6 - aLen - cBuffer)/5.72*13 )
        seedCircm = 2
        pickedEdges1 = e1.getSequenceFromMask(mask=('[#2800800 ]',
          ), )
        pickedEdges2 = e1.getSequenceFromMask(mask=('[#5 ]', ), )
        pickedEdges3 = e1.getSequenceFromMask(mask=('[#2c2a3550
          #55995 ]', ), )
```

```
        a.seedEdgeByBias(biasMethod=SINGLE, end1Edges=pickedEdges1,
           end2Edges=pickedEdges2, ratio=4.0, number=seedRight,
          constraint=FINER)
        a.seedEdgeByNumber(edges=pickedEdges3, number=seedCircm,
          constraint=FINER)

elif aLen >= 0.6 and aLen <= 0.8:
        seedRight = int( (6 - aLen - cBuffer)/5.72*13 )
        seedCircm = 3
        pickedEdges1 = e1.getSequenceFromMask(mask=('[#2800800 ]',
          ), )
        pickedEdges2 = e1.getSequenceFromMask(mask=('[#5 ]', ), )
        pickedEdges3 = e1.getSequenceFromMask(mask=('[#2c2a3550
          #55995 ]', ), )
        a.seedEdgeByBias(biasMethod=SINGLE, end1Edges=pickedEdges1,
           end2Edges=pickedEdges2, ratio=4.0, number=seedRight,
          constraint=FINER)
        a.seedEdgeByNumber(edges=pickedEdges3, number=seedCircm,
          constraint=FINER)

elif aLen > 0.8 and aLen <= 1.1:
        seedLeft  = 1
        seedRight = int( round((6 - aLen - cBuffer)/4.2*8, 0 ) )
        pickedEdges1 = e1.getSequenceFromMask(mask=('[#2000 ]', ),
          )
        pickedEdges2 = e1.getSequenceFromMask(mask=('[#905 ]', ), )
        a.seedEdgeByBias(biasMethod=SINGLE, end1Edges=pickedEdges1,
           end2Edges=pickedEdges2, ratio=4.0, number=seedLeft,
          constraint=FINER)
        pickedEdges3 = e1.getSequenceFromMask(mask=('[#0 #2000a0 ]'
          , ), )
        pickedEdges4 = e1.getSequenceFromMask(mask=('[#0 #60000 ]',
           ), )
        a.seedEdgeByBias(biasMethod=SINGLE, end1Edges=pickedEdges3,
           end2Edges=pickedEdges4, ratio=4.0, number=seedRight,
          constraint=FINER)

else:
        seedRight = int( round((6 - aLen - cBuffer)/4.2*8, 0 ) )
        if seedRight < 1:
                seedRight = int( 1 )

        seedLeft  = 10 - seedRight
        pickedEdges1 = e1.getSequenceFromMask(mask=('[#2000 ]', ),
          )
        pickedEdges2 = e1.getSequenceFromMask(mask=('[#905 ]', ), )
```

```
            a.seedEdgeByBias(biasMethod=SINGLE, end1Edges=pickedEdges1,
               end2Edges=pickedEdges2, ratio=4.0, number=seedLeft,
              constraint=FINER)
            pickedEdges3 = e1.getSequenceFromMask(mask=('[#0 #2000a0 ]'
              , ), )
            pickedEdges4 = e1.getSequenceFromMask(mask=('[#0 #60000 ]',
               ), )
            a.seedEdgeByBias(biasMethod=SINGLE, end1Edges=pickedEdges3,
               end2Edges=pickedEdges4, ratio=4.0, number=seedRight,
              constraint=FINER)

a = mdb.models[mdlName].rootAssembly
partInstances = (a.instances[instName], )
a.generateMesh(regions=partInstances)

mdb.models[mdlName].loads[loadName].setValues(magnitude=pLoad)

jobName = 'CFE_'+Rvalue+'_S'+str(int(round(pLoad)))[1:]+'_C'+str(
  aLen)[0]+str(aLen)[2:5]
mdb.Job(name=jobName, model=mdlName)
mdb.jobs[jobName].writeInput(consistencyChecking=OFF)
del mdb.jobs[jobName]

# jobList.append('%s' % jobName)
```

CenterCrack_XFEM_MC_AdjustValues.py (Use with XFEM-PN Monte Carlo)

```
## Opens CenterCrack_CFE_AdjustCrack.py and replaces values for
  crack analysis
#
# Rev    Date         Name             Changes
# 00     03-14-14     Josh Melson      -Created
#
# Modules
import os, sys, fileinput
#----------------------------------------
# Read in Variable names
c3  = sys.argv[1]
c4  = sys.argv[2]
num = sys.argv[3]
#----------------------------------------
#Check for 3 input arguments
checkarg = len(sys.argv) - 1
numarg   = 3
if checkarg < numarg:
    print "Fail"
    sys.exit("CenterCrack_XFEM_MC_AdjustLoad: Not enough input
      arguments")

elif checkarg > numarg:
    print "Fail"
    sys.exit("CenterCrack_XFEM_MC_AdjustLoad: Too many input
      arguments")

#----------------------------------------
# File Name
fileName = 'XFEM_R2_MC'
#----------------------------------------
# Names to search for
c3Name = 'c3'
c4Name = 'c4'
#----------------------------------------
# Replace Names
c3NameRepl = 'c3 = '+c3
c4NameRepl = 'c4 = '+c4
#----------------------------------------
# File Names
newFile = fileName+num
#----------------------------------------
# Read File, Replace, and Write
fIn = open(fileName+'.inp').read()
fOut  = open(newFile+'.inp', 'w')
for line in fIn:
```

```
        fOut.write(line)
fOut.close()

fLine  = open(newFile+'.inp').read()
fLine = '\n'.join(c3NameRepl if line.startswith(c3Name) else line
 for line in fLine.splitlines())
fLine = '\n'.join(c4NameRepl if line.startswith(c4Name) else line
 for line in fLine.splitlines())
open(newFile+'.inp', 'w').write(fLine)
```

CenterCrack_XFEM_StatusCheck.py (Use with all direct cyclic analyses

```
## Opens and examines status file for cycle increments
#
# Rev    Date         Name             Changes
# 00     02-17-14     Josh Melson      -Created
# 01     03-11-14     Josh Melson      -Add maxinc check
import numpy, fileinput, os, sys, fnmatch
## Read in values
fName = sys.argv[1]
## Define status file name
staName = fName+'.sta'
## Read data
staFile = fileinput.input([staName])
## Matching String for Step 2, Any Increment Size
step2Str = '   2 *'
cycLast  = 1
incAtOne = 0
maxInc   = 20
for line in staFile:
        if fnmatch.fnmatch(line, step2Str):
                lineNum = [int(r) for r in line.split() if r.
                  isdigit()]
                cycCurrent = lineNum[1]

                if cycCurrent - cycLast == 1:
                        incAtOne = incAtOne + 1
                        cycLast = cycCurrent
                        if incAtOne >= maxInc:
                                open(fName+'.hlt','w').close();

                else:
                        cycLast = cycCurrent


fileinput.close()
```

msgextract_ConvergCriteria_2D.py (Use with all direct cyclic analyses)

```
## Opens and examines stabilization criteria for direct cyclic
#
# Rev    Date         Name             Changes
# 00     02-10-14     Josh Melson      -Created
#
import numpy, fileinput, os, sys, fnmatch, re
## Read in values
fName = sys.argv[1]
## Define msg file names
msgName = fName+'.msg'
## Read data
msgFile = fileinput.input([msgName])
cycleStr  = '*CYCLE*STARTS*'
iternStr  = '*ITERATION*SUMMARY*'
avgFrcStr  = '*AVERAGE FORCE*TIME AVG. FORCE*'
maxDispStr = '*MAX. COEFFICIENT OF DISP.*'
coefR0uStr = '*COEFF. OF RESI. FORCE  ON CONST. TERM*'
coefRnuStr = '*COEFF. OF RESI. FORCE  ON PERIODIC TERMS*'
corrU0uStr = '*CORR. TO COEFF. OF DISP.    ON CONST. TERM*'
corrUnuStr = '*CORR. TO COEFF. OF DISP.    ON PERIODIC TERMS*'
fourierStr = '*NUMBER OF FOURIER TERMS*'
allZeroStr = '*ALL*ZERO*'
directCycl = '*D I R E C T    C Y C L I C*'
staticStep = '*S T A T I C   A N A L Y S I S*'
cNumStr = []
iNumStr = []
avgFStr = []
avgMStr = []
maxUStr = []
maxRStr = []
R0uStr  = []
RnuStr  = []
R0rStr  = []
RnrStr  = []
U0uStr  = []
UnuStr  = []
U0rStr  = []
UnrStr  = []
frrStr  = []
dirCycl = 0
for line in msgFile:
    if fnmatch.fnmatch(line, directCycl):
        dirCycl = 1

    if fnmatch.fnmatch(line, staticStep):
        dirCycl = 0
```

```
if dirCycl == 1:

    # Cycle
    if fnmatch.fnmatch(line, cycleStr):
        cNum = [int(s) for s in line.split() if s.isdigit()]

    # Iteration
    if fnmatch.fnmatch(line, iternStr):
        iNum = [int(r) for r in line.split() if r.isdigit()]
        iNumStr.append(str(iNum[0]))
        cNumStr.append(str(cNum[0]))

    # Fourier
    if fnmatch.fnmatch(line, fourierStr):
        fNum = [float(x) for x in re.findall("[0-9]+", line)]
        frrStr.append(str(fNum[0]))

    # Force
    if fnmatch.fnmatch(line, avgFrcStr):
        avgF = [float(x) for x in re.findall("-?\d+.?\d*(?:[Ee
          ]-?\+?\d+)?", line)]
        avgFStr.append(str(avgF[1]))

    if fnmatch.fnmatch(line, maxDispStr):
        maxU = [float(x) for x in re.findall("-?\d+.?\d*(?:[Ee
          ]-?\+?\d+)?", line)]
        maxUStr.append(str(maxU[0]))

    if fnmatch.fnmatch(line, coefR0uStr):
        if fnmatch.fnmatch(line, allZeroStr):
            R0u = [0]
        else:
            R0u = [float(x) for x in re.findall("-?\d+.?\d*(?:[
              Ee]-?\+?\d+)?", line)]
        R0uStr.append(str(R0u[0]))

    if fnmatch.fnmatch(line, coefRnuStr):
        if fnmatch.fnmatch(line, allZeroStr):
            Rnu = [0]
        else:
            Rnu = [float(x) for x in re.findall("-?\d+.?\d*(?:[
              Ee]-?\+?\d+)?", line)]
        RnuStr.append(str(Rnu[0]))

    if fnmatch.fnmatch(line, corrU0uStr):
        if fnmatch.fnmatch(line, allZeroStr):
```

```
                U0u = [0]
            else:
                U0u = [float(x) for x in re.findall("-?\d+.?\d*(?:[
                  Ee]-?\+?\d+)?", line)]
            U0uStr.append(str(U0u[0]))

        if fnmatch.fnmatch(line, corrUnuStr):
            if fnmatch.fnmatch(line, allZeroStr):
                Unu = [0]
            else:
                Unu = [float(x) for x in re.findall("-?\d+.?\d*(?:[
                  Ee]-?\+?\d+)?", line)]
            UnuStr.append(str(Unu[0]))

fileinput.close()
g = open(fName+'_Converge.txt','w')
for d in range(len(cNumStr)):
    g.write('%s, %s, %s, %s, %s, %s, %s, %s, %s\n' \
    %(cNumStr[d], iNumStr[d], frrStr[d], \
    avgFStr[d], maxUStr[d], R0uStr[d], RnuStr[d], U0uStr[d], UnuStr
      [d]))
g.close()
```

msgextract_ConvergLimits_2D.py (Use with all direct cyclic analyses)

```
## Opens and examines stabilization limits for direct cyclic
#
# Rev    Date         Name             Changes
# 00     02-10-14     Josh Melson      -Created
#
import numpy, fileinput, os, sys, fnmatch, re

## Read in values
fName = sys.argv[1]
## Define msg file names
msgName = fName+'.msg'
## Read data
msgFile = fileinput.input([msgName])
directCycl = '*D I R E C T    C Y C L I C*'
staticStep = '*S T A T I C   A N A L Y S I S*'
CR0 = '*CRITERION*RESI*CONST*'
CRn = '*CRITERION*RESI*ANY*'
CU0 = '*CRITERION*CORR*CONST*'
CUn = '*CRITERION*CORR*ANY*'
dirCycl = 0

for line in msgFile:
    if fnmatch.fnmatch(line, directCycl):
        dirCycl = 1

    if fnmatch.fnmatch(line, staticStep):
        dirCycl = 0

    if dirCycl == 1:

        # Limits
        if fnmatch.fnmatch(line, CR0):
            CR0val = [float(x) for x in re.findall("-?\d+.?\d*(?:[
              Ee]-\d+)?", line)]
            CR0Str = str(CR0val[0])

        if fnmatch.fnmatch(line, CRn):
            CRnval = [float(x) for x in re.findall("-?\d+.?\d*(?:[
              Ee]-\d+)?", line)]
            CRnStr = str(CRnval[0])

        if fnmatch.fnmatch(line, CU0):
            CU0val = [float(x) for x in re.findall("-?\d+.?\d*(?:[
              Ee]-\d+)?", line)]
            CU0Str = str(CU0val[0])
```

```
        if fnmatch.fnmatch(line, CUn):
            CUnval = [float(x) for x in re.findall("-?\d+.?\d*(?:[
              Ee]-\d+)?", line)]
            CUnStr = str(CUnval[0])

        if all(('CR0val' in locals(), 'CRnval' in locals(), 'CU0val
          ' in locals(), 'CUnval' in locals())):
            break

fileinput.close()
g = open(fName+'_Limits.txt','w')
g.write('%s, %s, %s, %s\n' %(CR0Str, CRnStr, CU0Str, CUnStr))
g.close()
```