2003

Plastic Dissipation Energy in Mixed-Mode Fatigue Crack Growth on Ductile Bimaterial Interfaces

Jeremy S. Daily
Wright State University

Plastic Dissipation Energy in Mixed-Mode Fatigue Crack Growth on Ductile Bimaterial Interfaces

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

By

JEREMY S. DAILY
B.S., Wright State University, 2001

2003
Wright State University

WRIGHT STATE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

June 9, 2003

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Jeremy S. Daily ENTITLED Plastic Dissipation Energy in Mixed-Mode Fatigue Crack Growth on Ductile Bimaterial Interfaces BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

Nathan W. Klingbeil, Ph.D. (Signed)
Thesis Director

Richard J. Bethke, Ph.D. (Signed)
Department Chair

Committee on
Final Examination

Nathan W. Klingbeil, Ph.D. (Signed)

Joseph C. Slater, Ph.D., P.E. (Signed)

Eric J. Tuegel, Ph.D. (Signed)

Joseph F. Thomas, Jr., Ph.D. (Approved)
Dean, School of Graduate Studies

Abstract

Daily, Jeremy S., M.S. Egr., Department of Mechanical and Materials Engineering, Wright State University, 2003. *Plastic Dissipation Energy in Mixed-Mode Fatigue Crack Growth on Ductile Bimaterial Interfaces.*

A new theory of fatigue crack growth in ductile solids has recently been proposed based on the total plastic energy dissipation per cycle ahead of the crack. This and previous energy-based approaches in the literature suggest that the total plastic dissipation per cycle can be closely correlated with fatigue crack growth rates under Mode I loading. The goal of the current study is to extend the dissipated energy approach to steady-state crack growth under mixed-mode loading conditions, with application to cyclic delamination of ductile interfaces in layered materials. The total plastic dissipation per cycle is obtained by 2-D elastic-plastic finite element analysis of a stationary crack in a general mixed-mode specimen geometry under constant amplitude loading. Both elastic-perfectly plastic and bi-linear kinematic hardening constitutive behaviors are considered, and numerical results for a dimensionless plastic dissipation per cycle are presented over the full range of relevant mechanical properties and mixed-mode loading conditions. In addition, numerical results are presented for the case of fatigue crack growth along a bonded interface between materials with identical elastic, yet dissimilar plastic properties, including mismatches in both kinematic hardening modulus and yield strength. Finally, the approach is generalized

to include mismatches in both elastic and plastic properties, and results for the dimensionless plastic dissipation per cycle are reported over the complete design space of bimaterial interfaces. The results of this thesis are of interest in soldering, welding, coating, electronic packaging, and a variety of layered manufacturing applications, where mismatches in both elastic and plastic properties can exist between the deposited material and the substrate.

Contents

List of Figures

List of Tables

List of Symbols

Chapter 1

$\frac{da}{dN}$	Fatigue crack growth rate
$\mathscr{G}$	Strain energy release rate
$\frac{dW}{dN}$	Plastic dissipation energy per cycle
σ_{ij}	Stress tensor
ε_{ij}	Strain tensor
K	Stress intensity factor
R	Load ratio

Chapter 2

P	Normal force
V	Shear Force
ψ	Mode-mix ratio
Π	Total potential energy
E	Young's modulus
U	Strain Energy
ν	Poisson's ratio
a	Crack Length

Chapter 3

η	Ratio of the layer thickness
σ_y	Yield Strength
J	Value of the path independent J-Integral
Π	Total potential energy
E_t	Strain hardening or tangent modulus

Chapter 4

$\hat{\sigma}$	Yield strength mismatch parameter

Chapter 5

α	Dundurs' mismatch parameter
β	Dundurs' mismatch parameter
μ	Shear modulus
Σ	Elastic Mismatch parameter
δ	Location of the neutral axis
ε	Oscillation index
i	The imaginary number, $\sqrt{-1}$
f	Load ratio

Acknowledgments

This research was funded by a masters fellowship from the Ohio Space Grant Consortium (OSGC), a seed grant funded through the National Aeronautics and Space Administration (NASA). I would like to thank my advisor, Dr. Klingbeil, for his guiding hand and verbose discussions during this research. Without him, none of this research would be possible. I hope to enjoy a prolific relationship in the future. I would also like to express my appreciation to Dr. Joseph Slater and Dr. Eric Tuegel for serving as members of my thesis committee.

Dedications

I would like to thank God for the many wonderful blessings and gifts He has granted me. All things I do through Christ who strengthens me. I would like to take this opportunity to thank my wonderful wife, Lisa, for her patience, love, and kindness during those long hours and late nights during this research. Her quiet pride and undying love inspired me to always do my best. I'd like to thank my father and mother, John and Carolyn, for their love and encouragement. It was my dad who challenged me and made me want to learn and think, for which I am grateful. I would like to recognize my Uncle Scott for our shared interest in "geeking." Talking about the wonderful world of computers gave me a lot of insight and desire to learn and understand computers, which was essential for this research. I really appreciate all my friends and family who have encouraged and supported me through the years.

Chapter 1

Introduction

1.1 Motivation

Fatigue cracks are a major concern throughout industry, with particular application to the aerospace community. With older aircraft dominating the service fleet for both military and civilian populations, the prospect of catastrophic failure due to fatigue cracking becomes more likely with time. Most work regarding fatigue crack growth has been empirical with countless tests being conducted through the past century. These tests have been used to construct fatigue crack growth rate curves and process maps for design purposes, but they shed little light on the physical mechanism of fatigue crack growth. All the empirical data from the years of testing can only be used for existing materials.

One proposed mechanism of fatigue crack growth is the total cyclic plastic dissipation energy[1]. In a recent paper by Klingbeil [1], the fatigue crack growth rate was shown to be directly proportional to the cyclic plastic dissipation energy under mode I loading con-

[1]The term "plastic dissipation energy" and "plastic work" are used interchangeably through out this thesis.

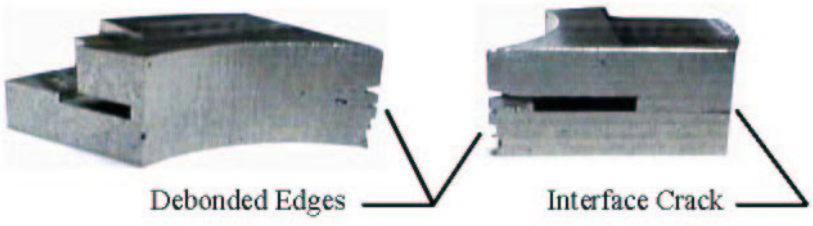


Figure 1.1: Interface cracking in a layered material system.

ditions.[2] Most fatigue cracks orient themselves into a pure mode I condition where the load opens the crack without any sliding along the crack face. As such, the literature and data recorded for fatigue crack growth is dominated by mode I conditions. There are, however, instances where cracks will not grow in a pure mode I direction. The case of fatigue cracking in a bimaterial or a layered manufactured system typically constrains the crack to the interface of the two layers– regardless of the mode of cracking. This occurs because the interface fracture toughness is lower than the homogeneous fracture toughness, which gives the fatigue crack a path of least resistance. Since fatigue crack growth can occur in mixed mode and the plastic dissipation energy is known to be a driving parameter in mode I, the mixed mode cyclic plastic dissipation energy is a quantity worth calculating. With the advent of new materials and material systems, an analytical fatigue crack growth predictor would prove useful in accelerating the introduction of new material systems. The plastic dissipation energy is directly related to fatigue crack growth in ductile metals, thus making it a primary quantity in developing any sort of fatigue crack growth rate predictor equations.

The numerical results for mixed mode plastic dissipation energy are of interest in welding, soldering and layered manufacturing applications, where high temperature material

[2]This study by Klingbeil [1] was conducted for C(T) specimen geometry which is Mode I because the shearing along the crack plane is zero.

deposition can result in mismatches in mechanical properties between the deposited material and the substrate. Figure 1.1 shows examples of these systems that arise in layered manufacturing, where a material is deposited on a similar substrate. Processes controlling the deposition procedure can also influence the strength, resulting an an elastically matched yet plastically mismatched material. Finally, by considering both elastic and plastic mismatches, this thesis provides a general survey of the cyclic plastic dissipation energy for all possible ductile metal interface systems.

1.2 Literature Review

As previously mentioned, Klingbeil [1] proposed a new theory of fatigue crack growth in ductile solids based on the total plastic energy dissipation per cycle ahead of the crack. The results of this and previous energy-based approaches in the literature suggest that the total plastic dissipation per cycle is a driving force for fatigue crack growth in ductile solids, and can be closely correlated with fatigue crack growth rates under mode I loading. The goal of the current paper is to extend the dissipated energy approach to steady-state crack growth under mixed-mode loading, with application to fatigue delamination of ductile interfaces in layered materials.

A critical plastic dissipation criterion for fatigue crack extension in ductile solids was first suggested by Rice [2]. Dissipated energy approaches to fatigue crack growth prediction have since been the subject of numerous analytical [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and experimental [14,15,16,17,18,19,20,21,22] investigations. The current approach considers the total plastic dissipation per cycle occurring throughout the reversed plastic zone ahead of the crack, which is a quantity of both theoretical and practical interest. As shown herein, the total plastic dissipation per cycle is directly related to the range of applied energy release

rate, which is typically used to correlate fatigue crack growth rates under mixed-mode loading [23]. Moreover, as opposed to the crack tip stresses and strains, the total plastic dissipation per cycle is a bounded quantity, which allows for straightforward interpretation of numerical results. Finally, numerical results for the total plastic dissipation per cycle can be directly compared to measurements of dissipated energy during fatigue crack growth, which have been reported in the literature by a number of researchers [14, 15, 16, 17, 18, 19, 20, 21, 22]. While the above cited studies have been restricted to mode I loading, the results of this work can be compared with subsequent studies of dissipated energy during sustained mixed-mode crack growth.

As discussed in the review paper by Qian and Fatemi [23], surface flaws and short cracks in homogeneous materials are typically subject to mixed-mode loading conditions, yet ultimately orient themselves such that Paris-regime crack growth occurs primarily in mode I. As such, the majority of the fatigue crack growth literature has focused on mode I loading. Recent studies of fatigue crack growth under mixed-mode loading have typically been concerned with the growth of short cracks [24, 25], fatigue crack threshold behavior [25,26,27], and the effect of mode-mix on crack growth direction [28,29,30]. A noteworthy investigation of fatigue crack growth in a homogeneous material under sustained mixed-mode loading has been conducted by Magill and Zwerneman [31].

This thesis considers the plastic energy dissipation associated with steady-state fatigue crack growth under mixed-mode loading, with particular application to cyclic delamination of ductile interfaces in layered materials. Fatigue delamination is a potential mode of failure in a variety of applications involving bonded layers of material, where mixed-mode crack growth along the bonded interface can be energetically favorable to mode I crack growth within either bonded layer. Layered material systems are the basis for numer-

ous solid freeform fabrication and layered manufacturing applications, and they occur in welding, soldering, coating, and electronic packaging applications [32, 33, 34, 35]. While little comprehensive experimental data is available, researchers have begun to investigate fatigue crack growth along solder joints and other bonded interfaces where mixed-mode delamination can be a predominant mode of failure [36, 37, 38, 39, 40, 41].

1.3 Modeling Approach

1.3.1 Dissipated Energy Theory

Following the work of Bodner *et al* [6], Klingbeil [1] has recently proposed a crack growth law of the form

$$\frac{da}{dN} = \frac{1}{\mathscr{G}_c}\frac{dW}{dN}, \tag{1.1}$$

where $\frac{da}{dN}$ is the fatigue crack growth rate, $\mathscr{G}_c$ is the critical strain energy release rate under monotonic loading (i.e., the fracture toughness), and $\frac{dW}{dN}$ is the total plastic dissipation per cycle occurring throughout the reversed plastic zone ahead of the crack tip.[3] The proposed crack growth law assumes that the total energy required to propagate a crack a unit distance in a given material is independent of the manner in which the energy is dissipated, be it monotonic or fatigue loading conditions. As outlined in [1], the proposed crack growth law results in a $(\Delta K)^4$ dependence of the fatigue crack growth rate[4], and has been shown to collapse the measured Paris-regime crack growth data for several ductile metals under constant amplitude, mode I loading conditions. Moreover, numerical results for the plastic dissipation per cycle were shown to be consistent with a variety of dissipated energy

[3] The plastic dissipation W is per unit width, as required by the units of equation (1.1).

[4] The work done by Klingbeil [1] shows the plastic dissipation is proportional to the fourth power of the loading, $(\Delta K)^4$which coincides with the actual measured fatigue crack growth rates.

measurements reported in the literature.

In theory, the crack growth law of equation (1.1) is applicable to fatigue crack growth under general mixed-mode loading conditions, where both $\mathscr{G}_c$ and $\frac{dW}{dN}$ depend on the mode-mix ratio. Hence, application of the crack growth law requires numerical calculation of the quantity $\frac{dW}{dN}$, which is the total plastic dissipation per cycle integrated over the reversed plastic zone ahead of the crack:

$$\frac{dW}{dN} = \int \int \left\{ \oint \sigma_{ij} d\varepsilon_{ij}^{p} \right\} dA. \tag{1.2}$$

1.3.2 Stationary Crack Modeling

In the current study, the total plastic dissipation per cycle of equation (1.2) is obtained by 2-D elastic-plastic finite element analysis of a stationary crack in a general mixed-mode layered specimen geometry. As discussed in [1], a stationary (as opposed to growing) crack modeling approach neglects the contribution of the actual crack extension to the total plastic dissipation occurring during any given load cycle. However, for Paris-regime crack growth in ductile solids, both the plastic work and surface energy contributions associated with the actual crack extension in any given cycle are negligible compared to the total plastic dissipation occurring throughout the reversed plastic zone ahead of the crack. As such, modeling the actual crack extension is unnecessary.

That said, it is important to note that stationary crack modeling is unable to capture the transient evolution of the cyclic constitutive behavior as the fatigue crack extends through previously yielded material [42], and neglects the possibility of plasticity-induced crack closure. In the current study, only elastic-perfectly plastic and bi-linear kinematic hardening constitutive behaviors are considered, each of which predicts plastic shakedown after

only a single cycle. As such, the results of this work should be viewed as a first approximation to the stabilized cyclic response under constant amplitude loading, and do not attempt to account for load ratio effects typically associated with fatigue crack closure.[5]

It should finally be noted that numerical results presented herein can be interpreted from a number of standpoints. First, in the context of the fatigue crack growth law of equation (1.1), the results are applicable to stabilized, self-similar crack extension under mixed-mode loading conditions. As previously outlined, such results are most applicable to layered material systems, where sustained mixed-mode crack growth is a potential mode of failure. However, the results may also be taken at face value, i.e., as simply the plastic dissipation associated with a single load cycle applied to a stationary crack tip under mixed-mode loading. In this context, the results of this work may be useful in the development of energy-based approaches for predicting crack growth direction or mixed-mode fatigue crack threshold behavior. Finally, the trends in plastic dissipation with mode-mix ratio presented herein may provide insight into discrepancies between mode I model results and dissipated energy measurements reported in the literature, which have been attributed in part to a mix of crack extension modes at the crack tip [22, 1].

1.4 Overview and Contributions

1.4.1 Overview

In this thesis, the total plastic dissipation per cycle is determined by 2-D elastic-plastic finite element modeling of a stationary crack in a general mixed-mode layered specimen geometry. The merits and limitations of stationary crack modeling are discussed in [1],

[5] In the absence of crack closure, the applied load ratio $R = K_{min}/K_{max}$ was shown in [1] to have only a negligible effect on the total plastic dissipation per cycle, and is not considered further herein.

and were reviewed in section 1.3.2. Comprehensive numerical results for a dimensionless plastic dissipation per cycle are presented over the full range of mixed-mode loading for both elastic-perfectly plastic and bi-linear kinematic hardening materials. The numerical results reported in Chapter 3 provide significant insight into the role of crack tip constraint, material hardening behavior, and mode-mix ratio on the dissipated plastic energy during fatigue crack growth. Results are presented in Chapter 4 for the case of fatigue crack growth along an interface between bonded layers with identical elastic yet dissimilar plastic properties, including a mismatch in both yield strength and hardening modulus. Chapter 5 considers the case of elastic mismatch across an interface crack with matching plastic properties. The results of the numerical analysis presented in this thesis are applied in Chapter 6 and the pertinent results are summarized in Chapter 7. The Appendix holds some illustrations of crack tip plastic zones, raw data from the finite element runs, and the script used to generate the data.

The goal of this thesis is to take the reader through the required background in fracture mechanics and fatigue crack growth to understand the context of the current research. A sufficient background is necessary to understand the terms used to describe the mechanics of interface crack problems. The initial analysis was to determine the dependence of the plastic dissipation energy on the magnitude of the loading which showed a power law relationship between the plastic work and the strain energy release rate. The plastic dissipation energy was made dimensionless to show the effects of mode, specimen geometry and plastic constraint for the case where both layers are matched. The plastic mismatch analysis shows the effect of a yield strength mismatch and hardening modulus mismatch. The effect of an elastic modulus mismatch on the cyclic plastic dissipation energy is also shown. Finally, a couple of cases are shown for exemplar bimaterial systems. From these

results some general design guidelines can be articulated to make interfaces more debond resistant.

1.4.2 Contributions of This Thesis

- This thesis gives previously unreported data for the plastic dissipation energy under mixed mode loading for i) homogeneous materials, ii) plastically mismatched materials, and iii) elastically mismatched materials.
- Insight is provided concerning the contributions of plastic constraint, specimen geometry, and material hardening behavior to plastic dissipation energy for both layered and homogeneous systems.
- Illustrations of the plastic zone shape are revealed for different modes and elastic mismatches. These pictures provide insight into the physical mechanisms of crack tip mechanics.
- New methods in extracting numerical results for the ω term found in interfacial fracture problems [43, 44, 34, 35].
- A complete listing of a Python script used to iterate the finite element analysis.
- Design guidelines to minimize the plastic work per cycle (and reduce the fatigue crack growth rate).
- Comparison with literature that shows a promising application of the plastic dissipation energy to predicting the fatigue crack growth rate in bimaterial systems.

Chapter 2

Background

2.1 Linear Elastic Fracture Mechanics

2.1.1 Stress Intensity Factors

The stress analysis of a crack tip can be found in most texts on fracture mechanics (e.g. [45]), and reveals the stress fields near a crack tip to be singular. Although non-physical, the crack tip stress singularity is important in understanding the mechanics of crack extension

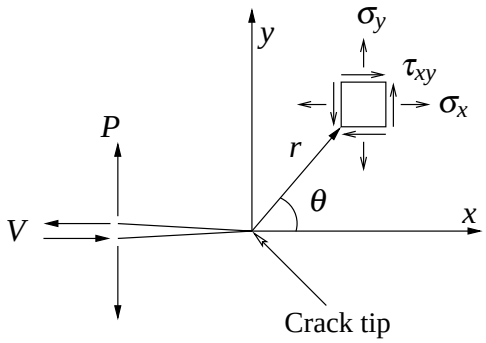


Figure 2.1: Definition of a coordinate axis at the crack tip.

by fatigue. Figure 2.1 shows the polar coordinates used to derive the stress fields at the crack tip where, V is the shearing force and P is the normal force. An asymptotic analysis of this traction prescribed problem show that the stress fields have a series solution where the first term goes as $1/\sqrt{r}$. As $r \to 0$ the first term of the series solution becomes dominant. The stress fields can then be written as a function of θ and a proportionality constant known as the *stress intensity factor.* The analysis of stress results in the following formulas (in tensor notation):

$$\lim_{r\to 0} \sigma_{ij}^{I} = \frac{K_I}{\sqrt{2\pi r}} f_{ij}(\theta)^I \tag{2.1}$$

$$\lim_{r\to 0} \sigma_{ij}^{II} = \frac{K_{II}}{\sqrt{2\pi r}} f_{ij}(\theta)^{II} \tag{2.2}$$

$$\lim_{r\to 0} \sigma_{ij}^{III} = \frac{K_{III}}{\sqrt{2\pi r}} f_{ij}(\theta)^{III} \tag{2.3}$$

As $r \to 0$ in equations (2.1-2.3) the values of the stresses become large thus exhibiting the classic square root singularity. The proportionality constants K have subscripts denoting the mode of loading: mode I, mode II, and mode III. In a linear elastic material, these stresses can be linearly superposed resulting in a total stress tensor according to:

$$\sigma_{ij}^{(total)} = \sigma_{ij}^{(I)} + \sigma_{ij}^{(II)} + \sigma_{ij}^{(III)}. \tag{2.4}$$

The modes correspond to the different ways a crack tip can be loaded as shown in Figure 2.2. For this study only a planar analysis is considered, so there is no out of plane shearing (mode III). As such there are two modes under consideration, which leads to a straight

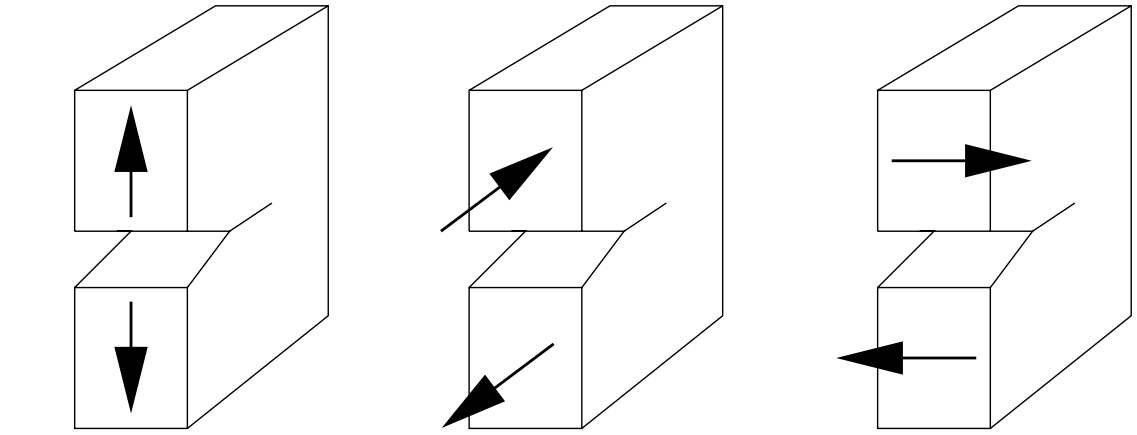


Figure 2.2: Schematic of modes for a crack.

forward definition of the mode-mix as:

$$\psi = -\tan^{-1}\left(\frac{K_{II}}{K_I}\right). \tag{2.5}$$

The above definition gives an angular representation of the ratio of the mode II and mode I stress intensity factors. This is beneficial when reporting results for both mode I and mode II on the same plot because the axis defining mode-mix is bound between -90° and 90°. However, due to symmetry arguments, the positive values and negative values of ψ yield the same results for elastically matched layers so only half the range is needed to report results for all possible mode-mix ratios. A negative value of ψ corresponds to a negative value of the K_{II} component, which is a negative value of the shear. For homogenoeus materials, a negative shear has the same physical effect of a positive shear so it is required to only report positive values of the mode-mix ratio.

2.1.2 Strain Energy Release Rate

The strain energy release rate determines how much energy is available for crack extension. The rate is not a derivative with respect to time, rather with respect to new crack area:

$$\mathscr{G} = -\frac{d\Pi}{dA} \tag{2.6}$$

where Π is the total potential energy. $\mathscr{G}$ can be determined experimentally through either a load controlled test or a displacement controlled test where the details can be found in [45, 46, 47].

The stress intensity factors and the strain energy release rate are related with the following expression:

$$\mathscr{G} = \frac{K_I^2 + K_{II}^2}{\bar{E}}, \tag{2.7}$$

where $\bar{E} = E$ for plane stress and $\bar{E} = \frac{E}{1-\nu^2}$ for plane strain (ν is Poisson's ratio). This relationship provides much needed convenience when analyzing mixed mode problems, because the loading magnitude can be reported as a single quantity $\mathscr{G}$, as opposed to two different stress intensity factors K_I and K_{II}. It can also be noted that only two of the four parameters ($K_I, K_2, \mathscr{G}$, and ψ) are independent, and that equations (2.5) and (2.7) provide the relationships to recover all four values. That being said, reporting mixed mode data requires either K_I and K_{II} or $\mathscr{G}$ and ψ.

2.1.3 J-integral

The J-integral is a path independent contour integral about the crack tip defined as:

$$J = \oint_\Gamma \left(U dy - T_i \frac{\partial u_i}{\partial x} ds \right), \tag{2.8}$$

where U is the strain energy density, $T_i = \sigma_{ij} n_j$ are the traction vector components, u_i are the displacement components and ds is the incremental contour length around any closed contour Γ. Among the details found in [45] is the important conclusion that for a linear elastic problem, the J-integral is equal to the strain energy release rate $\mathscr{G}$. This relationship allows for a verification procedure of the numerical models as presented in section 3.3.3.

2.1.4 Small Scale Yielding

Since the stress fields exhibit singular behavior near the crack tip, plastic deformation is inevitable. However, assuming the extent of plasticity is small, the plastic zones near the crack tip are solely controlled by the stress intensity fields. When the plastic zone becomes too large, the linear elastic fracture quantities K_I and K_{II} are no longer the sole driving forces of the crack tip plasticity. The assumption of small scale yielding says linear elastic fracture mechanics applies to the crack problem, and this can be verified by checking the analytical solution of $\mathscr{G}$ against the numerical result for the J-integral.

2.2 Fatigue Crack Growth

It is not uncommon that machine components in service are known to have existing cracks. This raises the question of how fast the cracks will grow and when the part will fail. These questions motivate fatigue crack growth rate studies. Figure 2.3 is an illustration of a crack growing with respect to the number of cycles. The actual fatigue crack growth rate is the first derivative of the a vs. N curve. Measurements of fatigue cracks will record crack length as a function of the number of cycles. If a load controlled experiment is conducted, then the stress intensity factor will increase as the crack length increases according to the

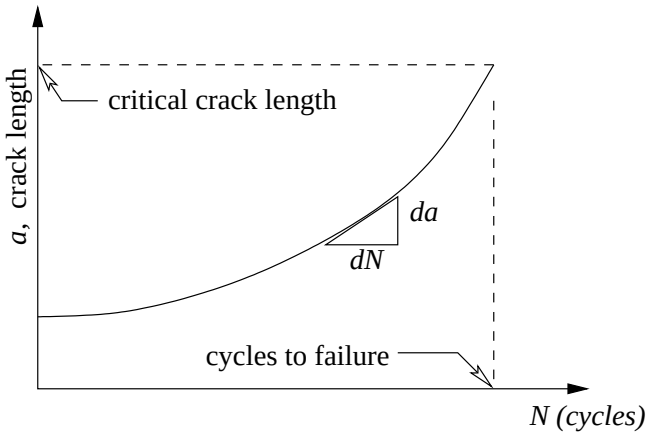


Figure 2.3: Growth of a fatigue crack from detection until failure.

definition $\Delta K = F\Delta\sigma\sqrt{\pi a}$ where F is a geometry factor. It is also important to note $\Delta K = K_{max} - K_{min}$ and the load ratio $R = \frac{K_{max}}{K_{min}}$. These equations can, of course, be written in terms of $\mathscr{G}$ by using equation (2.7). However, if the minimum value is not zero,[1] the interpretation of the Δ value changes when switching from stress intensity factors (K) to strain energy release rate ($\mathscr{G}$). In other words, care must be taken since $\Delta\mathscr{G} = \mathscr{G}_{max} - \mathscr{G}_{min} = \frac{K_{max}^2 - K_{min}^2}{E} \neq \frac{(K_{max} - K_{min})^2}{E}$. A more detailed discussion of performing fatigue crack growth experiments can be found in [47].

Fatigue crack growth modeling is dominated by experimental data and empirical curve fits. The most famous regression equation is from P.C. Paris [48] who observed a power law dependence for the fatigue crack growth rate of the form

$$\frac{da}{dN} = C(\Delta K)^m. \tag{2.9}$$

Since K and $\mathscr{G}$ are related by equation (2.7), the power law can be rewritten in terms of

[1] The analyses performed herein were conducted with a load ratio $R = 0$ following reports from [1] that load ratio had a minimal effect on the overall plastic dissipation energy.

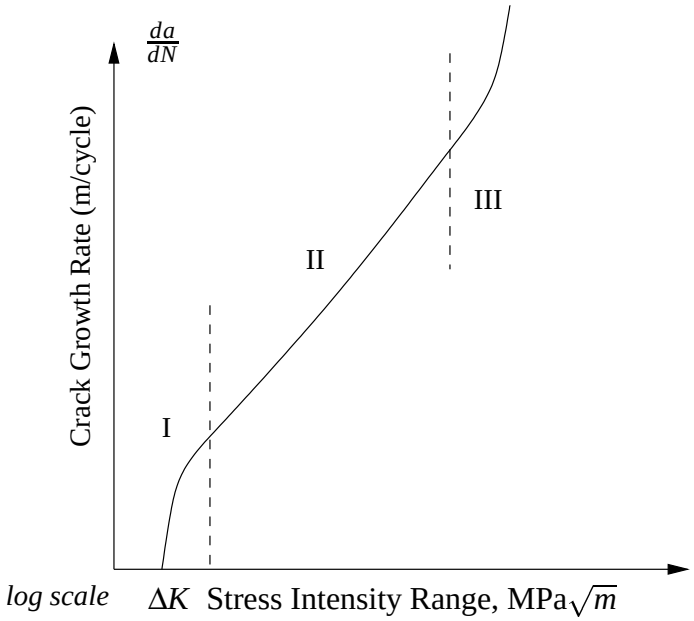


Figure 2.4: Typical fatigue crack growth rate curve with three distinct regions: I– Threshold region, II– Power law region $\left(\frac{da}{dN} = C(\Delta K)^m\right)$, and III– unstable rapid crack extension.

$\mathscr{G}$ as

$$\frac{da}{dN} = C(\Delta\mathscr{G})^m, \tag{2.10}$$

which is the preferred method of reporting mixed mode fatigue crack growth data [23]. The constants C and m are typically determined experimentally, and reports of these values often include a 10% error. These values are still useful as engineering constants for a given material. However, the actual mechanics of fatigue crack growth require analytical modeling because they are not revealed with these empirical curve fits.

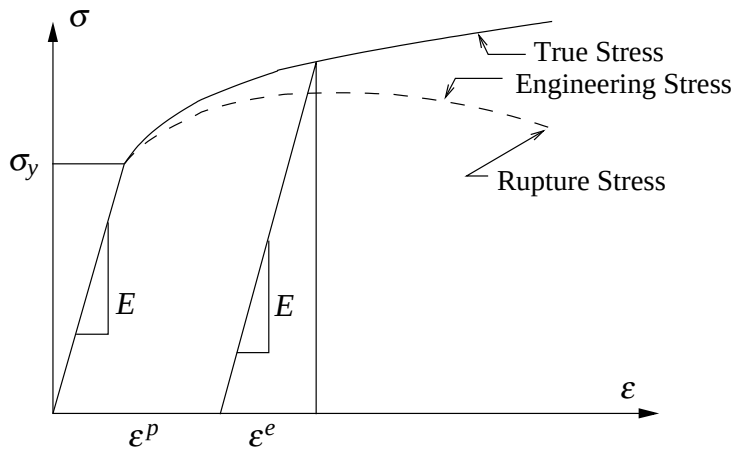


Figure 2.5: Stress strain curves showing engineering stress and true stress.

2.3 Elastic-Plastic Behavior

Every material has a finite strength so that, when exceeded, it will either rupture or yield. Because the stresses will increase as $r \to 0$, a plastic zone will appear as a result of the higher stress. This plasticity will occur even if the global stresses are well under the yield strength. Knowing that the material at the crack tip will yield, a discussion of some of the properties and models of plasticity is in order.

Figure 2.5 shows a typical stress-strain diagram for a ductile metal revealing the linear elastic region and the plastic region. The most significant difference between elastic and plastic behavior is that the material does not return to its original state after undergoing plastic deformation. When a material is loaded elastically, it will deform the same each time with a given amount of stress. The deformation of an elastic material is only dependent on the application of the stress that created the deformation. Plastic behavior, however, depends on the load history in addition to the applied loads.

The mechanism for plastic deformation is dislocation motion. A detailed discussion of

the physical metallurgy behind plastic deformation is beyond the scope of this thesis and is only briefly summarized here. Dislocation motion in a polycrystalline material (most ductile metals) involves planes of atoms sliding or gliding over one another. These glissile motions are characterized by the slip plane and slip direction. During plastic deformation, these dislocations can accumulate and impede each other resulting in strain hardening [46].

Another important note is that a plastically deforming material has no volume change. Hydrostatic stresses will not plastically deform a material even when each component is higher than the uniaxial tensile strength. Yield is governed by the deviatoric stress components from which the Von Mises yield criterion is established. Once a material has exceeded the Von Mises criterion for yield, it will not undergo any more volume change. This incompressible behavior is equivalent to a Poisson's ratio of $\nu = \frac{1}{2}$ for a plastically deforming material.

The onset of yielding is determined by the Von Mises criterion[2] which is recognized as

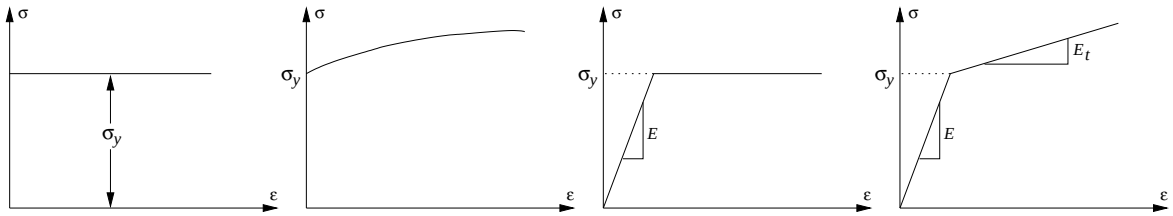
$$\bar{\sigma} = \frac{1}{\sqrt{2}}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_1 - \sigma_3)^2\right]^{1/2} = \sigma_y \tag{2.11}$$

where σ_i are the principal stresses, $\bar{\sigma}$ is the effective stress. Plastic deformation occurs when the effective stress value exceeds that of the uniaxial yield strength.

2.3.1 Constitutive models

The microscopic mechanism of plastic deformation is inconsequential to the macroscopic quantification of energy, which can be obtained from a load-displacement or stress-strain analysis. The classic stress-strain diagram of engineering materials shows the response of

[2] Synonyms for the Von Mises yield criterion include: i) J_2 flow criterion, ii) equivalent distortion energy criterion, and iii) maximum octahedral shear stress criterion.



(a) Rigid-Perfectly Plastic (b) Rigid plastic (c) Elastic-perfectly plastic (d) bilinear hardening

Figure 2.6: Idealized stress-strain diagrams.

materials from which work can be calculated. The engineering stress curve is shown to distinguish between the two methods of reporting stress-strain data. While the engineering stress is determined by dividing the force by the initial area, true stress is determined by dividing the stress by the actual area. The difference in the curves at low values of strain ($\leq 5\%$) is small compared to that of larger strains. For all subsequent analysis in this thesis only true stress-strain relationships are considered.

Figure 2.5 also illustrates the concept of elastic strain ε^e and plastic strain ε^p. The total strain is the addition of the elastic and plastic strain. The elastic and plastic components are determined by drawing a line parallel to the initial linear section of the curve from the point corresponding to the total strain. Where the line intersects the abscissa gives the plastic strain. Also, the elastic strain will increase as the material strain hardens more. Understanding these concepts is important when implementing the material properties in a finite element modeling code as described in section 2.3.3.

For modeling purposes there are a few common idealized stress strain diagrams illustrated in Figure 2.6. The rigid-perfectly plastic (Figure 2.6a) model is usually employed when dealing with large-scale plastic deformation, where the elastic deformation makes up

a very small contribution to the overall deformation. Since the elastic and plastic strains near a crack tip under small scale yielding are comparable in magnitude, this model is not useful herein. Likewise, the rigid-plastic (Figure 2.6b) model assumes large scale yielding with strain hardening and is also not valid for the analyses herein. However, the elastic-perfectly plastic (Figure 2.6c) and bi-linear hardening (Figure 2.6d) models are valid when considering problems in the elastic-plastic regime. The elastic-perfectly plastic case is also a special case of the bi-linear hardening model when the tangent modulus $E_t = 0$. It is recognized that other constitutive models exist and may be more realistic, but the simplicity of the bi-linear hardening model is preferred for this thesis.[3] The bi-linear hardening model can account for different strain hardening rates in terms of a single ratio E_t/E. So far, only uniaxial tensile response has been illustrated in Figures 2.5 and 2.6 and the actual application of these models requires a discussion of cyclic response.

2.3.2 Isotropic Cyclic Hardening

Figure 2.7 shows the stress strain diagram for a material that exhibits isotropic cyclic hardening. The cyclic response involves loading followed by unloading, or loading in the opposite direction. As a material is loaded past yield, it will harden and make the material stronger. Using an isotropic material model says that when the material is unloaded and reloaded it will yield in compression after the same magnitude of stress is applied (i.e. $\Delta\sigma = 2\sigma'$) as shown in Figure 2.7. The value σ' is the highest value of stress experienced during the previous loading cycle. Subsequent hardening takes place and further increases the yield strength. This pattern leads to an always increasing yield strength which, of

[3] Also, the quantity desired, plastic dissipation energy, is an integrated quantity and the error associated with using a simplified material model is smoothed and averaged over the whole plastic zone. As a result, capturing the exact behavior of the material as it transitions from the linear elastic region to the plastic regime is of limited consequence.

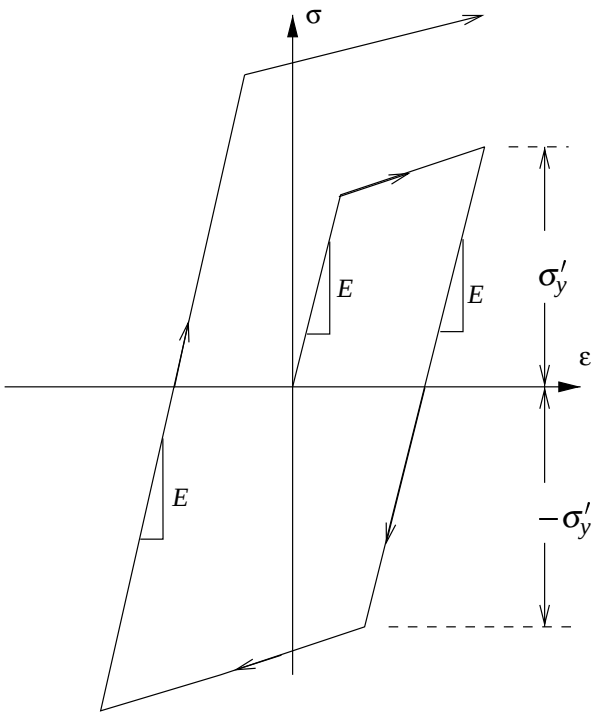


Figure 2.7: Cyclic isotropic material hardening behavior when loaded to equivalent positive and negative strain values.

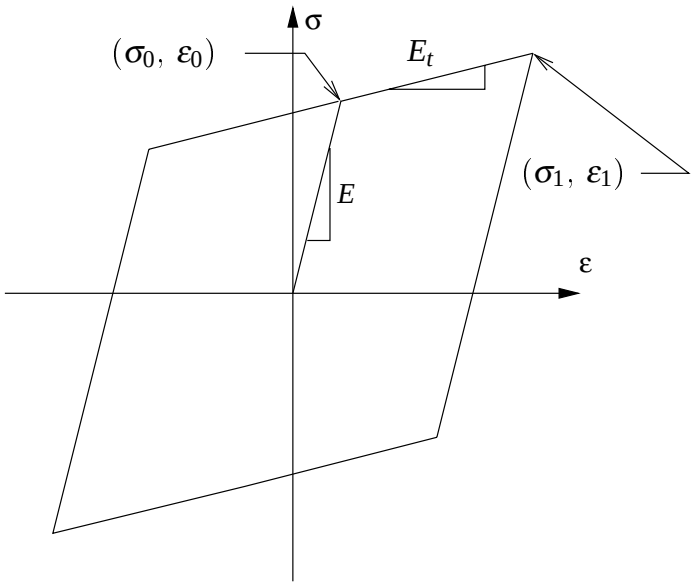


Figure 2.8: Bi-linear kinematic hardening model when loaded to similar positive and negative strain values. Plastic shakedown occurs after one cycle.

course, is non-physical. When the hardening modulus is low, the material will require more iterations before moving the yield a similar amount than if the tangent modulus is higher. Obviously when the material is elastic-perfectly plastic, no hardening is exhibited and there is no gain in yield strength.

2.3.3 Kinematic Cyclic Hardening

Kinematic hardening predicts that the material will yield in the reverse direction after a change in stress $\Delta\sigma = 2\sigma_y$ where σ_y is the initial yield strength. The bi-linear kinematic hardening model shown in Figure 2.8 is the model of choice for this analysis, and includes the limiting case of elastic-perfectly plastic (Figure 2.6c). In the context of classical small-strain elastoplasticity, the bi-linear kinematic hardening model can be used to approximate the stabilized cyclic response during constant amplitude loading. Real materials exhibit

isotropic behavior for the first few cycles and evolve to kinematic behavior. This trend toward kinematic behavior is known as plastic shakedown. Figure 2.8 shows that the bi-linear kinematic hardening model provides for a reduced yield strength upon reversal (the Bauschinger effect), and predicts plastic shakedown after only a single cycle.

The elastic modulus E, tangent modulus E_t, and yield strength σ_y are the three independent parameters completely defining the material response for a bi-linear kinematic hardening model. The tangent modulus can range from a slope of zero to that of Young's modulus ($0 \leq E_t \leq E$). Obviously a tangent modulus equal to the elastic modulus is far fetched, so the results of such analysis are included only for academic completeness. Without loss of meaning, the tangent modulus is more conveniently expressed as a ratio E_t/E, whose values have the range $0 \leq E_t/E \leq 1$.

The material models previously discussed need to be implemented into a finite element program. The software used for this for this thesis is ABAQUS, produced by HKS Software. The implementation of the kinematic hardening model in the finite element code was not trivial and is detailed in the next paragraphs.

ABAQUS uses the plastic strains to define the material behavior instead of total strains. The first data point in the plastic properties table has to be the yield point when the plastic strain is zero. If no other data are given, ABAQUS assumes elastic-perfectly plastic behavior. To give a non zero value to the tangent modulus, another point must be added to the plastic behavior definition table. By fixing a value of stress slightly higher than the yield point, a range of plastic strain components can be computed corresponding to different moduli ratios E_t/E. Given the bi-linear kinematic hardening model as shown in Figure

2.8, the slope of the tangent modulus is defined as

$$E_t = \frac{\sigma_1 - \sigma_0}{\varepsilon_1 - \varepsilon_0}. \tag{2.12}$$

The strain at yield is solely elastic strain, whereas the strain at point 1 has both elastic and plastic parts. The elastic strain is defined from a simplified Hooke's law as $\frac{\sigma_1}{E}$. Given those simple definitions and dividing equation (2.12) through by Young's modulus yields

$$\frac{E_t}{E} = \frac{\sigma_1 - \sigma_0}{E\varepsilon_{1p} + \sigma_1 - \sigma_0}. \tag{2.13}$$

If the yield point and elastic modulus are known, solving equation (2.13) for the plastic strain is determined by

$$\varepsilon_{1p} = \frac{1}{E}\left(\frac{(\sigma_1 - \sigma_0)}{E_t/E} - \sigma_1 + \sigma_0\right). \tag{2.14}$$

The values generated using equation (2.14) are input in the property definition tables of the finite element software.

The primary goal of these modeling procedures is to account for some type of material hardening. It is true that a real material does not exhibit bi-linear behavior, but it can be represented with an equivalent bi-linear model with reasonable accuracy. The procedure for extracting the plastic dissipation energy will be the same as those presented herein even if the constitutive model were to be changed.

Chapter 3

Matching Layers

3.1 Global Problem Geometry

Some of the desired qualities of the specimen geometry include the presence of an analytical solution to verify the numerical results, the ease of implementation of finite element codes, and the applicability of real world specimens. Examination of the literature lead to a generalized specimen found in a paper by Suo and Hutchinson [43] and is shown in Figure 3.1. This specimen has a special case equivalent to the 4-point bend test specimen proposed by Charalabides *et al.* in [49].

The mixed-mode layered specimen geometry of Figure 3.1 is composed of two bonded layers of isotropic materials #1 and #2, which can have different thicknesses (h_1 and h_2), different elastic properties (E_1, E_2, ν_1, and ν_2) and different plastic properties (σ_{y1}, σ_{y2}, E_{t1}, and E_{t2}). In this chapter, both the elastic and plastic properties of materials 1 and 2 are the same, while mismatches are considered in plastic and elastic properties in Chapters 4 and 5. The relative thickness ratio of the layers $\eta = \frac{h_1}{h_2}$ only changes the mode of the problem,[1]

[1]The relative thicknesses also affect $\mathscr{G}$, but the end results are normalized by $\mathscr{G}$ so this is irrelevant.

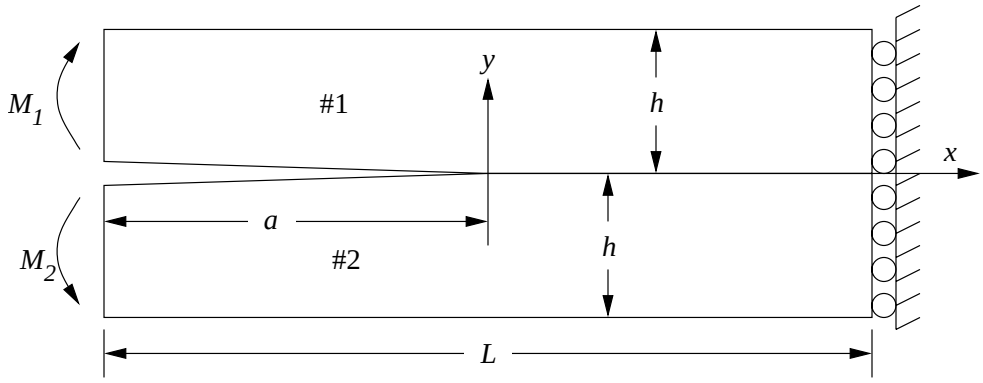


Figure 3.1: Specimen geometry for mixed-mode cracking with matching layer thicknesses.

which can also be changed by changing the loads. As a result, the ratio $\eta = \frac{h_1}{h_2} = 1$ is used in this analysis for ease of implementation in the computer code.

The loading consists of pure bending moments M_1 and M_2 applied to the top and bottom layers, which are equilibrated by a symmetry condition on the right hand side. The variation in the bending moments M_1 and M_2 allows consideration of the full range of mode-mix values, from pure mode I to pure mode II.

In light of the symmetry condition, the modeled length L actually represents half the total specimen length. Both the length L and the crack length a are sufficiently long to allow for steady-state conditions at the crack tip, so that the energy release rate is independent of crack length (see [43, 44] for more details). Also, the slenderness of the layers allows the specimen to be analyzed using beam theory, which provides a check on the computer solution. The dimensions used in all numerical analyses discussed in the next sections were $L = 50$ mm, $h_1 = 5$ mm, $h_2 = 5$mm, and $a = 25$ mm.

3.2 Analytical Models

As mentioned in section 3.1, a semi-analytical solution for steady-state cracking along the interface of a general bimaterial specimen configuration, having mismatches in both layer thickness and elastic properties, has been provided by Suo and Hutchinson [43]. As detailed in the next few sections, the results of [43] can be reduced to provide an analytical solution for the specimen configuration considered herein, in which there is no mismatch in either elastic properties or layer thickness.

3.2.1 General solution

For the case of no mismatch in layer thickness or elastic properties the mode I and mode II stress intensity factors for the problem of Figure 3.1 are

$$K_I = \frac{\sqrt{3}(M_1 + M_2)}{h^{3/2}} \tag{3.1}$$

$$K_{II} = -\frac{3(M_1 - M_2)}{2h^{3/2}}. \tag{3.2}$$

It is important to note that in the presence of small scale yielding, the elastic stress intensity factor solutions of equations (3.1) and (3.2) are valid even in the presence of a mismatch in plastic properties across the interface, so long as the elastic properties E and ν are identical.

Inspection of equations (3.1) and (3.2) reveals that when $M_1 = M_2$, the K_{II} component vanishes leaving pure mode I loading ($\psi = 0^\circ$). Also, when $M_1 = -M_2$, the K_I component vanishes leaving a pure mode II condition ($\psi = 90^\circ$). Another simplification of (3.1) and (3.2) occurs when $M_1 = 0$ (or $\psi = 41^\circ$), which is a special case of the four-point bend test

specimen geometry commonly used for inter-facial fracture testing of layered materials [49, 32].

Substitution of equations (3.1) and (3.2) into equation (2.7) gives the steady-state energy release rate for the problem of Figure (3.1a) as

$$\mathscr{G} = \frac{3(7M_1^2 + 2M_1M_2 + 7M_2^2)}{4\bar{E}h^3}. \tag{3.3}$$

The above result can also be directly determined from the difference in strain energy per unit crack area far behind and ahead of the crack tip, which is the hallmark of steady-state delamination problems [43]. The difference in strain energy for the problem of Figure 3.1 can be determined from elementary beam theory, as discussed in the next section.

3.2.2 Beam Theory Solution

The beam theory solution is helpful in providing insight into the mechanics, as well as a crosscheck for the previous results. As discussed in [32, 44], advancing a steady state crack is the same as taking the amount of crack advance da from ahead of the crack and placing it behind the crack, as shown in Figure 3.2. The strain energy release rate is then quantified

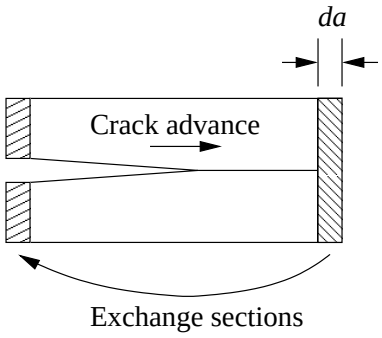


Figure 3.2: Illustration of an equivalent steady state crack advancement.

by the difference in the strain energy in the piece that was ahead of the crack (whole) and the same piece behind the crack (split).

Consider a section of the specimen far ahead of the crack in Figure (3.1a). The strain energy per unit length is given as:

$$U = \int \sigma_{ij}\varepsilon_{ij} dV. \tag{3.4}$$

Since the layer thicknesses are equal, the materials are elastically matched and the strains are linearly distributed across the cross section in the x-direction, the strain energy per unit length in a section becomes:

$$\frac{dU}{da} = \frac{M^2}{2EI}. \tag{3.5}$$

Defining $\mathscr{G}$ with equation (2.6) and subbing in equation (3.5) gives

$$\mathscr{G} = -\left(\frac{6(M_1 - M_2)^2}{E(h+H)^3} - \left(\frac{6M_1^2}{Eh^3} + \frac{6M_2^2}{EH^3}\right)\right) \tag{3.6}$$

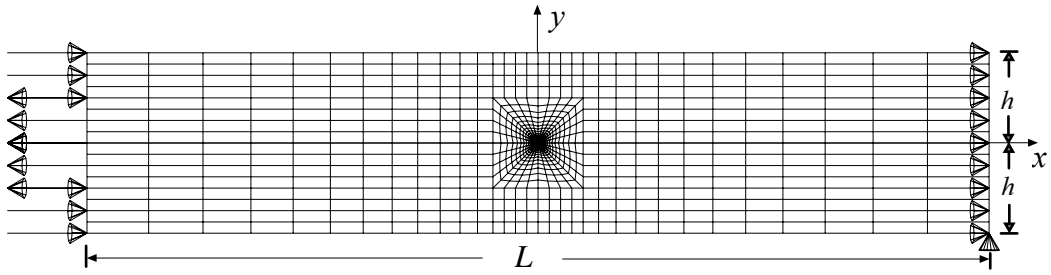
when $dA = B\,da$ and $d\Pi = dU_{ahead} - dU_{behind}$. Throughout this thesis B refers to the depth in the z-direction which is considered unity. Equation (3.6) further reduces to equation (3.3)

$$\mathscr{G} = \frac{3(7M_1^2 + 2M_1M_2 + 7M_2^2)}{4Eh^3}, \tag{3.7}$$

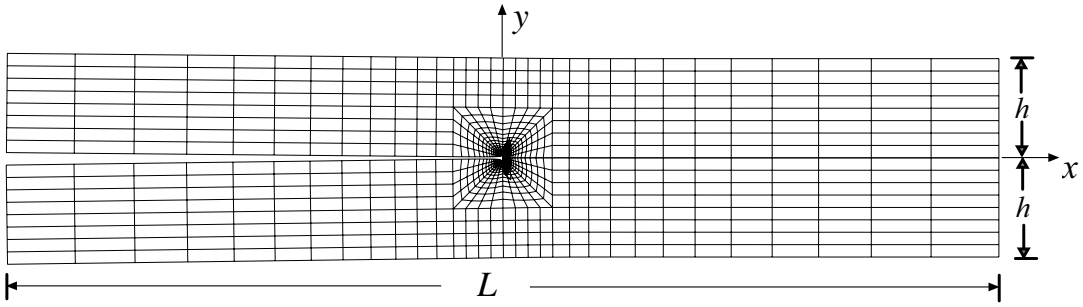
thus showing a simple beam theory solution and verification of the strain energy release rate found in Suo and Hutchinson [44, 43].

3.3 Numerical Modeling

3.3.1 Finite Element Analysis



(a) Undeformed finite element model



(b) Finite element model with a defomation factor of 1000

Figure 3.3: Finite element mesh, loading, and boundary conditions

The total plastic dissipation per cycle is obtained herein from a 2-D finite element model of the geometry of Figure 3.1 under constant amplitude, mixed-mode loading. The finite element mesh, applied loads and boundary conditions are illustrated in Figure 3.3. For

ease of implementation, the moments M_1 and M_2 are applied in the form of equal and opposite uniform stress distributions. The loading illustrated in Figure 3.1 results in equal and opposite bending moments, which corresponds to the case of pure mode I. A pure mode II loading would have the applied moments in the same direction. For matching materials, the full range of mixed-mode loading has been considered by holding M_1 constant and varying M_2 in the range $-M_1 \leq M_2 \leq M_1$.

The finite element model uses 8-node bi-quadratic reduced integration elements contained in the commercial software package ABAQUS. The analysis employs classical small-strain incremental elastoplasticity with Von Mises yield criterion, which is generally appropriate for metals and other ductile solids. Reduced integration elements are chosen for their accuracy during nearly incompressible material response, which results from the pressure-independent yielding assumed in the elastoplasticity formulation. The elements are highly biased toward the crack tip, with the smallest element measuring only $0.5\,\mu m$. As discussed in [1], such fine mesh resolution is needed to accurately resolve the reversed plastic zone upon load reversal, and to ensure convergence of the continuum theory solution.[2] As discussed in [1], the total plastic dissipation per cycle is insensitive to the choice of crack-tip elements, so standard (as opposed to quarter-point) elements are used at the crack tip.

One very powerful tool in ABAQUS is the ability to write scripts to automate the finite runs. Version 6.1 and later of ABAQUS is built on the Python interpreted language and scripts can be written and recorded to reproduce the exact procedure used to generate a numerical result. Those steps can then be iterated with a script to set up a parametric study as shown in Appendix C.

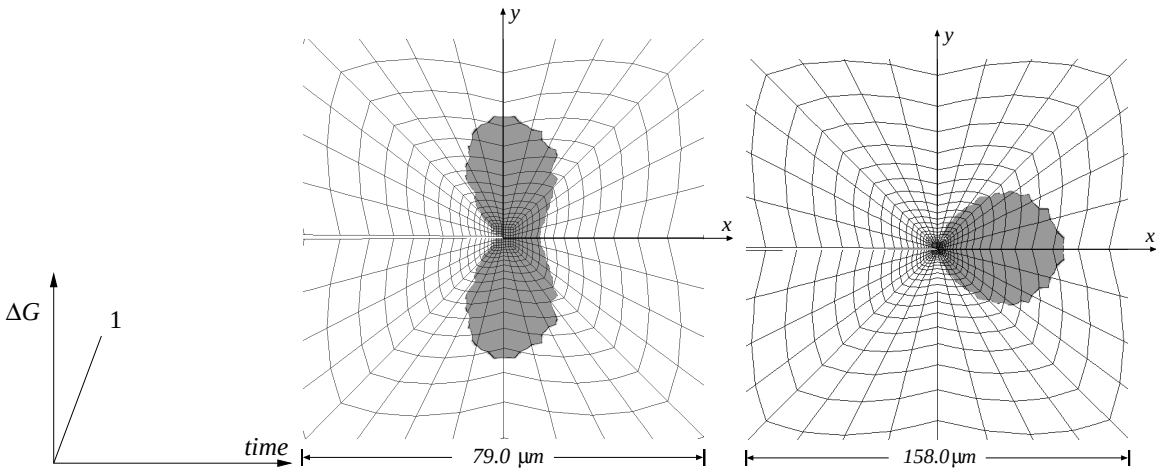
[2] It should be noted that convergence of the continuum solution does not police its applicability. As such, care should be taken in applying the results of this work for cases in which the reversed plastic zone is on the order of the grain size of the material.

3.3.2 Crack Tip Plasticity

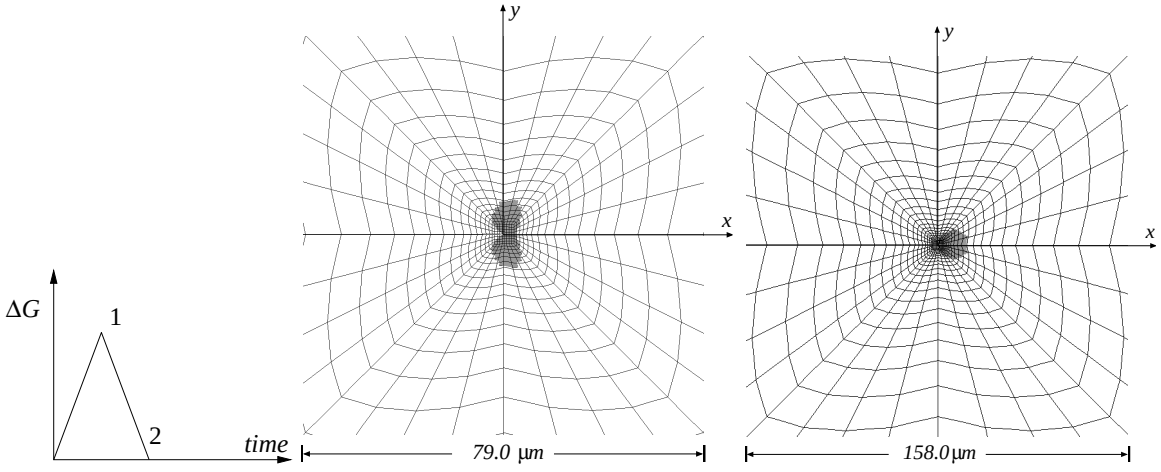
The effect of mode-mix ratio on the evolution of forward and reversed plastic zones during a complete load cycle ($R = \mathscr{G}_{min}/\mathscr{G}_{max} = 0$) is illustrated for both plane stress and plain strain in Figures 3.4 to 3.6. The material considered is elastic-perfectly plastic ($E_t/E = 0$) with elastic modulus $E = 73.1$ GPa, yield strength $\sigma_y = 300$ MPa, and Poisson's ratio $\nu = 0.3$. For ease of comparison, the applied range of energy release rate is held constant at $\Delta\mathscr{G} = 200\ J/m^2$.

As shown in Figures 3.4a and 3.4b, both the shape and size of the forward plastic zones under pure mode I loading are in keeping with expectations from classical fracture mechanics analyses, as well as with previous results in the literature [45]. In particular, unconstrained yielding results in a much larger plastic zone in plane stress (Figure 3.4b) than in plane strain (Figure 3.4a). While the forward plastic zones scale with $(\Delta K/\sigma_y)^2$, the reversed plastic zones scale with $(\Delta K/2\sigma_y)^2$, which is in keeping with the plastic superposition argument first put forth by Rice [2]. As such, the greatest extent of the reversed plastic zones of Figures 3.4c and 3.4d is roughly 1/4 that of the forward plastic zones of Figures 3.4a and 3.4b.

The asymmetry of crack tip plasticity during mixed-mode loading is evident from Figure 3.5, where the phase angle of $\psi = 41^\circ$ represents a nearly equal mix of mode I and II loading. More importantly, a comparison of the scale factors in Figures 3.4 and 3.5 reveals that an increase in mode II component significantly increases the extent of crack tip plasticity in both plane stress and plane strain. The difference in the plane stress and plane strain plastic zones reduces with an increasing shearing component. The reason stems from the definitions of plane stress and plane strain. For plane stress the out of plane principal stress

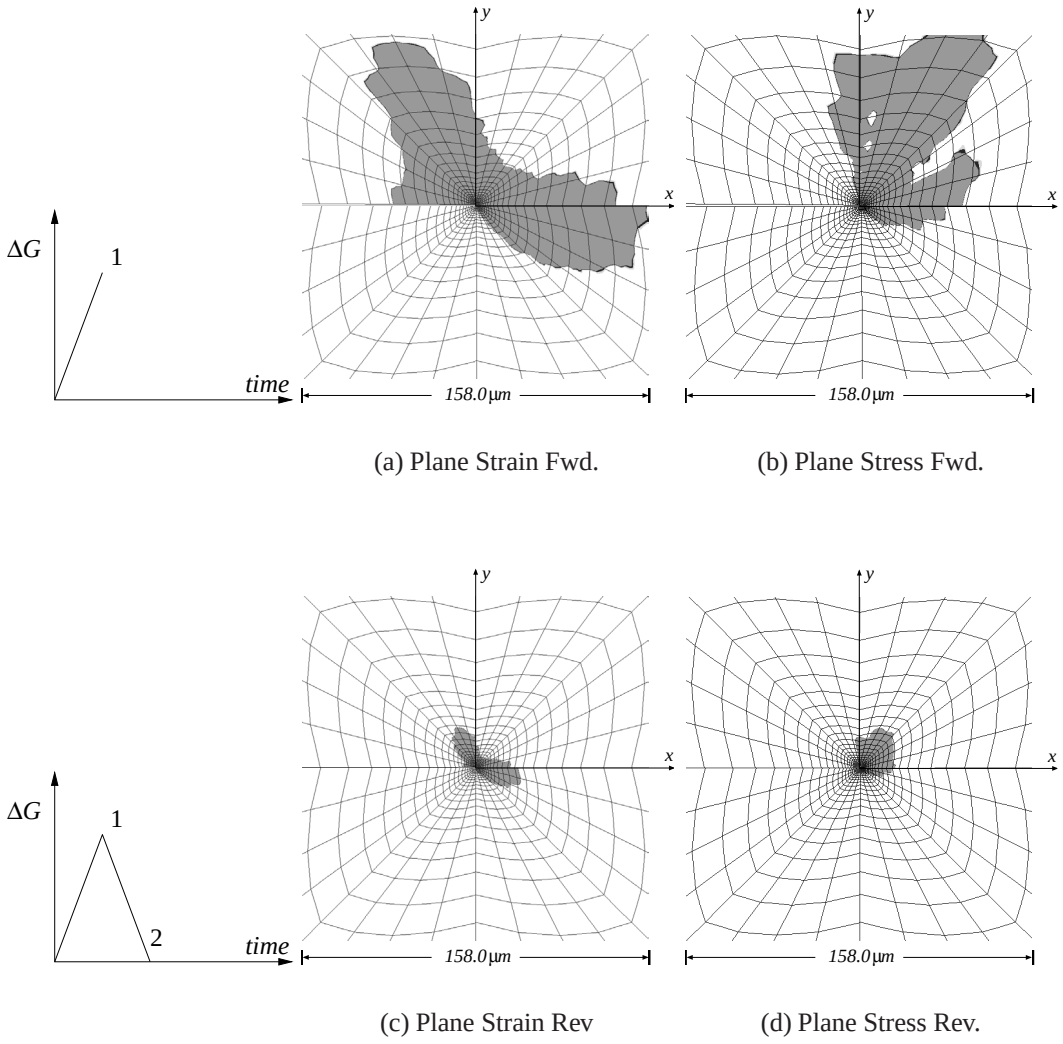


(a) Plane Strain Fwd. (b) Plane Stress Fwd.



(c) Plane Strain Rev. (d) Plane Stress Rev.

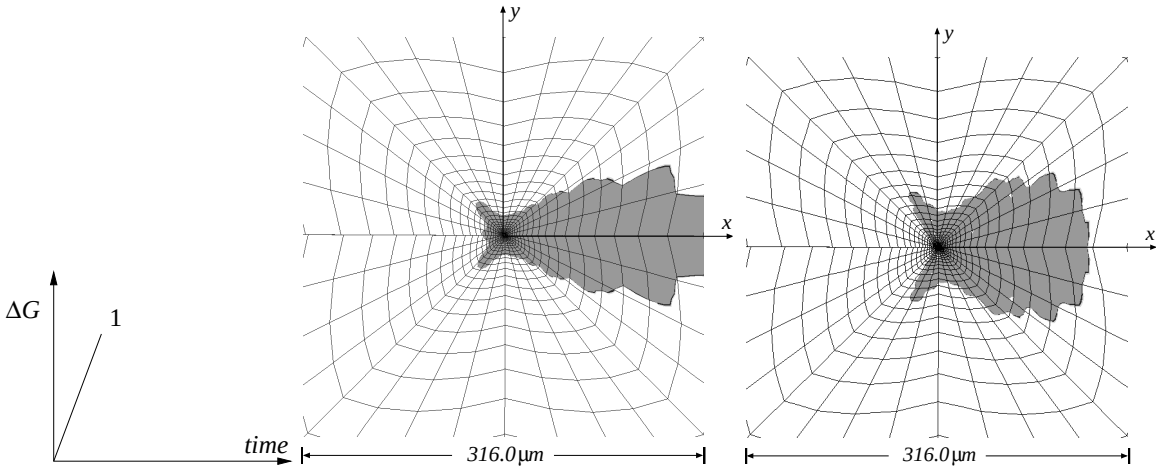
Figure 3.4: Forward and reversed plastic zones in pure mode I
with $E = 73.1$ GPa, $\nu = 0.3$, $\sigma_y = 300$ MPa, and $\Delta G = 200\,J/m^2$.



(a) Plane Strain Fwd. (b) Plane Stress Fwd.

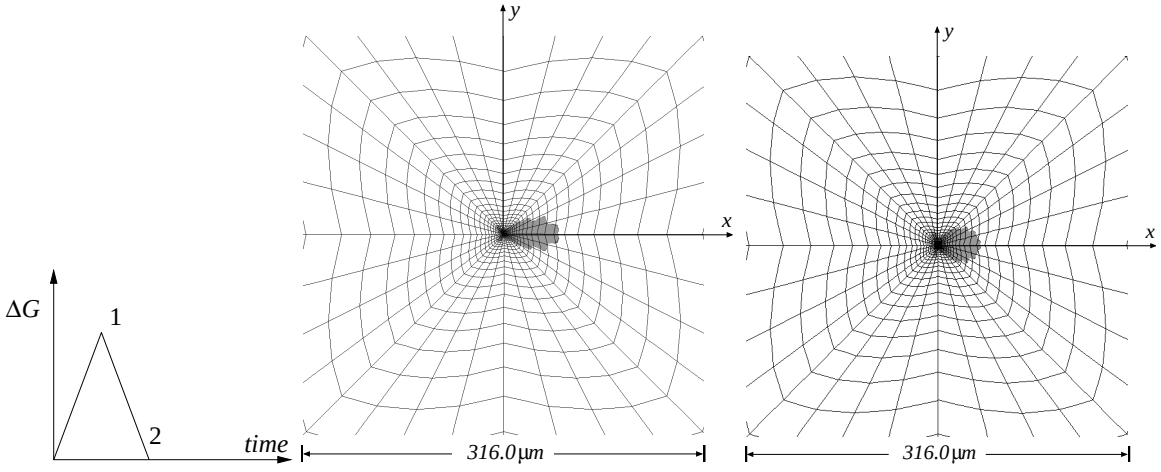
(c) Plane Strain Rev (d) Plane Stress Rev.

Figure 3.5: Forward and reversed plastic zone when $\psi = 41^\circ$
with $E = 73.1$ GPa, $\nu = 0.3$, $\sigma_y = 300$ MPa, and $\Delta G = 200\,J/m^2$.



(a) Plane Strain Fwd.

(b) Plane Stress Fwd.



(c) Plane Strain Rev.

(d) Plane Stress Rev.

Figure 3.6: Forward and reversed plastic zones in pure mode II with $E = 73.1$ GPa, $\nu = 0.3$, $\sigma_y = 300$ MPa, and $\Delta G = 200\,J/m^2$.

is

$$\sigma_3 = 0 \tag{3.8}$$

and for plane strain the out of plane stress is

$$\sigma_3 = \nu(\sigma_1 + \sigma_2) \tag{3.9}$$

If a material is under pure in-plane shear (pure mode II), the principal stresses are equal and opposite, leaving no distinction between equations (3.8) and (3.9).

3.3.3 Verifying Small Scale Yielding

It should be noted that the plastic zone sizes of Figures 3.4-3.6 are well within the range of small-scale yielding, which has been independently verified for all cases considered herein. First, J-integral estimates available in ABAQUS have been calculated at maximum load and directly compared to equation (3.3) ($J = \mathscr{G}$ for linear elastic fracture). While crack tip plasticity invalidates J-integral estimates within the plastic zone, those taken from contours outside the plastic zone have been found to agree with equation (3.3) to five significant digits. Such agreement can only be obtained in the presence of small-scale yielding. In addition, interaction integral estimates for the stress intensity factors have been obtained from elastic finite element runs of the specimen geometry. The results have been in excellent agreement with the the closed-form solutions of equations (3.1) and (3.2), as well as with the J-integral estimates obtained from the elastic-plastic analysis.

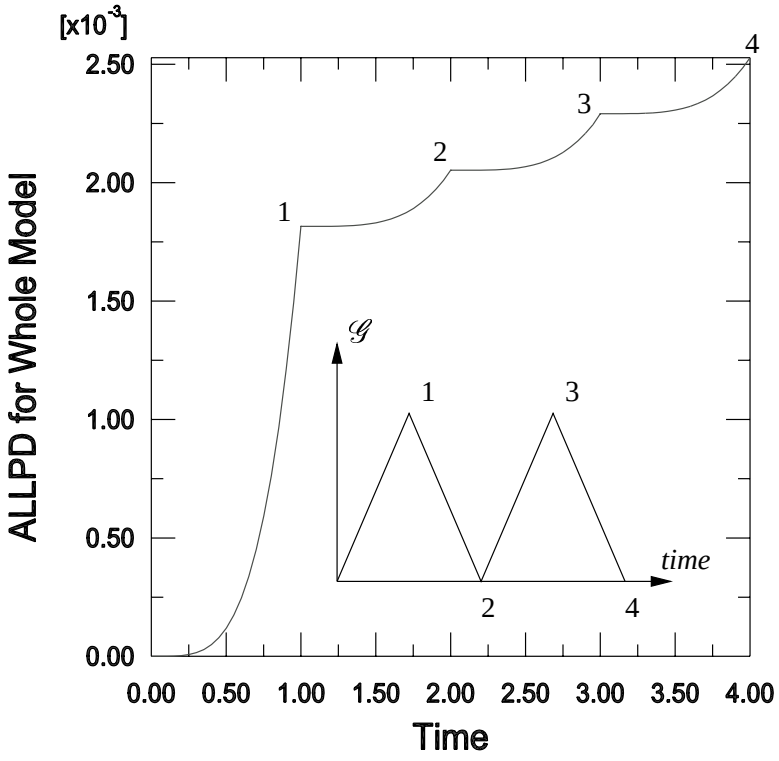


Figure 3.7: Representative time history graph of the cumulative plastic dissipation energy (keyword: ALLPD in ABAQUS).

3.4 Numerical Results and Discussion

3.4.1 Preliminary Finite Element Results

Figure 3.7 shows a representative trace of the time history of the plastic dissipation energy automatically tracked in ABAQUS. As Figure 3.7 shows, the plastic dissipation energy in the first loading cycle is much larger than in subsequent cycles. Two cycles are necessary because the plastic deformation during the first load cycle occurs throughout the forward plastic zone, while plastic deformation in subsequent cycles is restricted to the reversed plastic zone. Moreover, for both elastically-perfectly plastic and bi-linear kinematic hard-

ening materials, the plastic dissipation remains constant after the second cycle. As such, the quantity $\frac{dW}{dN} = \Delta W_{24}$ represents a steady-state value of $\frac{dW}{dN}$ in all subsequent cycles. The cyclic dissipated energy is the value of ALLPD at time step 4 less the value of ALLPD at time step 2[3]. This further illustrates the concept of plastic shakedown where every cycle dissipates the same amount of energy.

Representative plane strain finite element results for the total plastic dissipation per cycle $\frac{dW}{dN}$ as a function of applied range of energy release rate $\Delta\mathscr{G}$ are plotted over the full range of mode-mix values in Figure 3.8. As shown in Figure 3.8, a least-square curve fit of the numerical data results in a power-law relation of the form

$$\frac{dW}{dN} = C\left(\Delta\mathscr{G}\right)^m. \tag{3.10}$$

The results of the regression analysis showed that the exponent of the power law relation for all cases considered was in the range $1.99 \leq m \leq 2.04$ with $R^2 \geq 0.99$. Thus, to within numerical error, the exponent of the power law relation is $m = 2$, and is unaffected by the mode-mix ratio. In light of equation (1.1), the predicted fatigue crack growth rate is proportional to $\Delta\mathscr{G}^2$, which is within the range of observations of mixed-mode fatigue crack growth on ductile interfaces [36, 37, 38, 39, 40, 41]. It should also be noted that for an applied load ratio $R = 0$, the quantity $\Delta\mathscr{G}^2$ corresponds directly to $\Delta\left|K\right|^4$, or for the case of mode I loading, ΔK^4. This is in keeping with previous energy-based theories of fatigue crack growth under mode I loading, and is consistent with fatigue crack growth data for a variety of ductile metals [1].

[3]The actual process of extracting the results is automated in the `extractWork` function of the script in Appendix C.

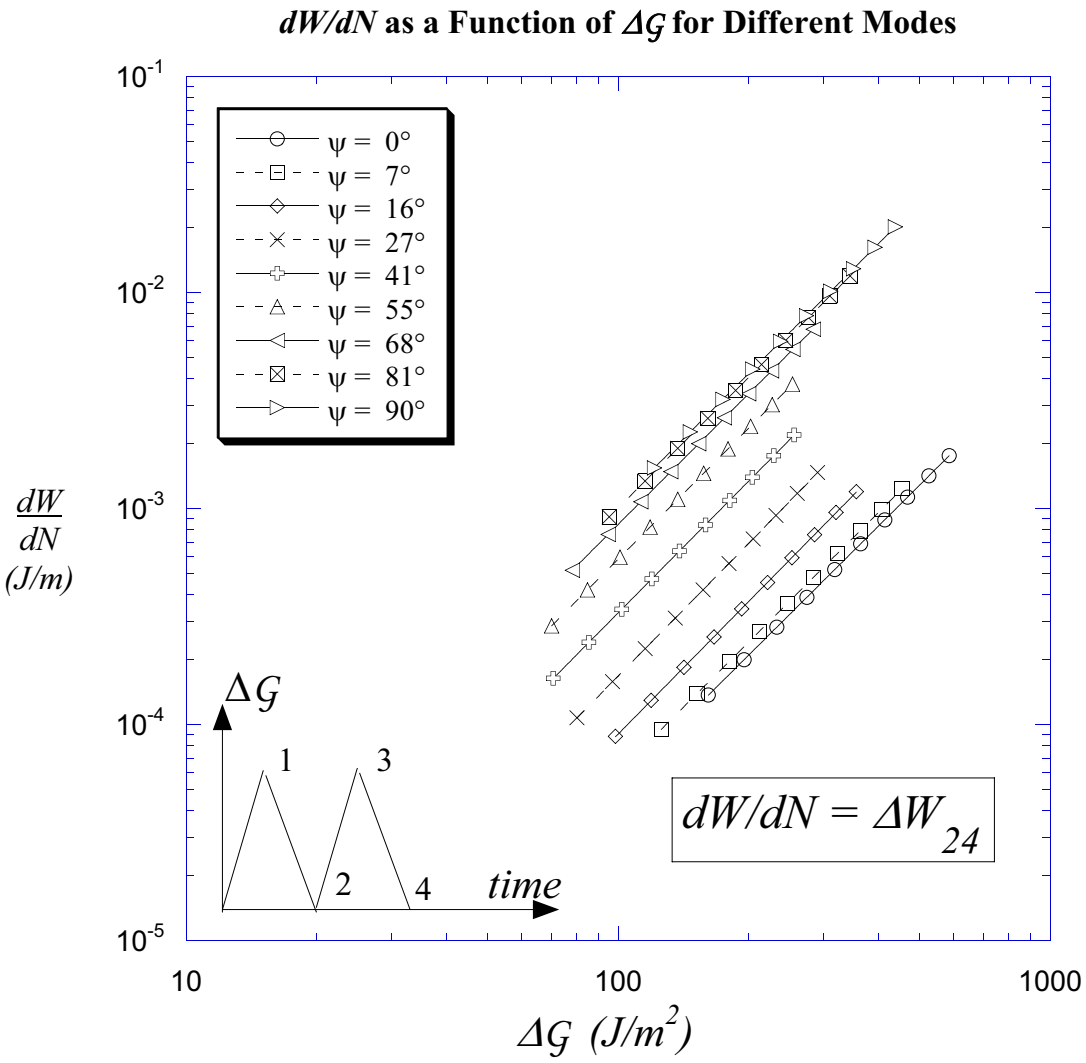


Figure 3.8: Effect of mode-mix on $\frac{dW}{dN}$ in plane strain when $E_t/E = 0$, $E = 73.1$ GPa, $\nu = 0.3$, and $\sigma_y = 300$ MPa.

Convergence of $\frac{dW}{dN}$	Mesh Resolution		
Time Step	Coarse	Medium	Fine
.1	1.1216E-3	1.1138E-3	1.1138E-3
.05	1.1258E-3	1.1184E-3	1.1188E-3
.025	1.1281E-3	1.1189E-3	1.1191E-3

Table 3.1: Convergence study of $\frac{dW}{dN}$ for pure mode I with matching layers when $E_t/E = 0$, $\nu = 0.3$, $E = 73.1$ GPa, $\sigma = 300$ MPa and $\Delta\mathscr{G} = 200 J/m^2$.

Convergence of $\frac{dW}{dN}$	Mesh Resolution		
Time Step	Coarse	Medium	Fine
.1	1.3399E-3	1.3422E-3	1.3427E-3
.05	1.3491E-3	1.3509E-3	1.3510E-3
.025	1.3514E-3	1.3532E-3	1.3531E-3

Table 3.2: Convergence study of $\frac{dW}{dN}$ for $\psi = 41^\circ$ with matching layers when $E_t/E = 0$, $\nu = 0.3$, $E = 73.1$ GPa, $\sigma = 300$ MPa and $\Delta\mathscr{G} = 200 J/m^2$.

3.4.2 Convergence Studies

It should finally be noted that a rigorous convergence study was performed in both time and space by successively halving both the element edge length and the maximum time step used in ABAQUS' automatic time-stepping algorithm. In so doing, the value of $\frac{dW}{dN}$ from the production mesh of Figure 3.3 varied less than 1 percent from the value of $\frac{dW}{dN}$ obtained from the finest mesh. A summary of these studies are presented in Tables 3.1-3.3 with the production runs being in the middle of the table (medium mesh resolution and time step = 0.05)

3.4.3 Non-Dimensionalization

In order to facilitate a general presentation of results, the plastic dissipation per cycle can be non-dimensionalized in terms of the yield strength, applied energy release rate and elastic

Convergence of $\frac{dW}{dN}$	Mesh Resolution		
Time Step	Coarse	Medium	Fine
.1	0.012585	0.012511	0.012507
.05	0.012667	0.012586	0.012579
.025	0.012688	0.012608	0.012601

Table 3.3: Convergence study of $\frac{dW}{dN}$ for pure mode II with matching layers when $E_t/E = 0$, $\nu = 0.3$, $E = 73.1$ GPa, $\sigma = 300$ MPa and $\Delta\mathscr{G} = 200\,J/m^2$.

modulus as

$$\frac{dW^*}{dN} = \frac{\sigma_y^2}{E\Delta\mathscr{G}^2}\frac{dW}{dN}. \tag{3.11}$$

In light of equation (1.1), the fatigue crack growth rate can be written in terms of the dimensionless plastic dissipation per cycle as

$$\frac{da}{dN} = \frac{\bar{E}\Delta\mathscr{G}^2}{\sigma_y^2\mathscr{G}_c}\frac{dW^*}{dN}. \tag{3.12}$$

For a material with matching elastic layers, the dimensionless plastic dissipation $\frac{dW}{dN}^*$ depends on the applied mode-mix ratio ψ, Poisson's ratio ν, and the hardening ratio E_t/E.

3.4.4 Effect of Mode-Mix

Figure 3.9 shows a plot of $\frac{dW}{dN}^*$ vs. ψ after applying equation 3.11 to the data of Figure 3.8. All ninety points from Figure 3.8 are collapsed to the "S" shaped curve of Figure 3.9 which validates the normalization of the data with equation (3.11). This collapse of the data shows a definite influence of the mode-mix on the plastic dissipation energy. Clearly, the plastic dissipation increases significantly with mode-mix, and is between one and two orders of magnitude greater in mode II than in mode I. This result might be expected in light of the increase in plastic zone size with mode-mix illustrated in Figures 3.4-3.6.

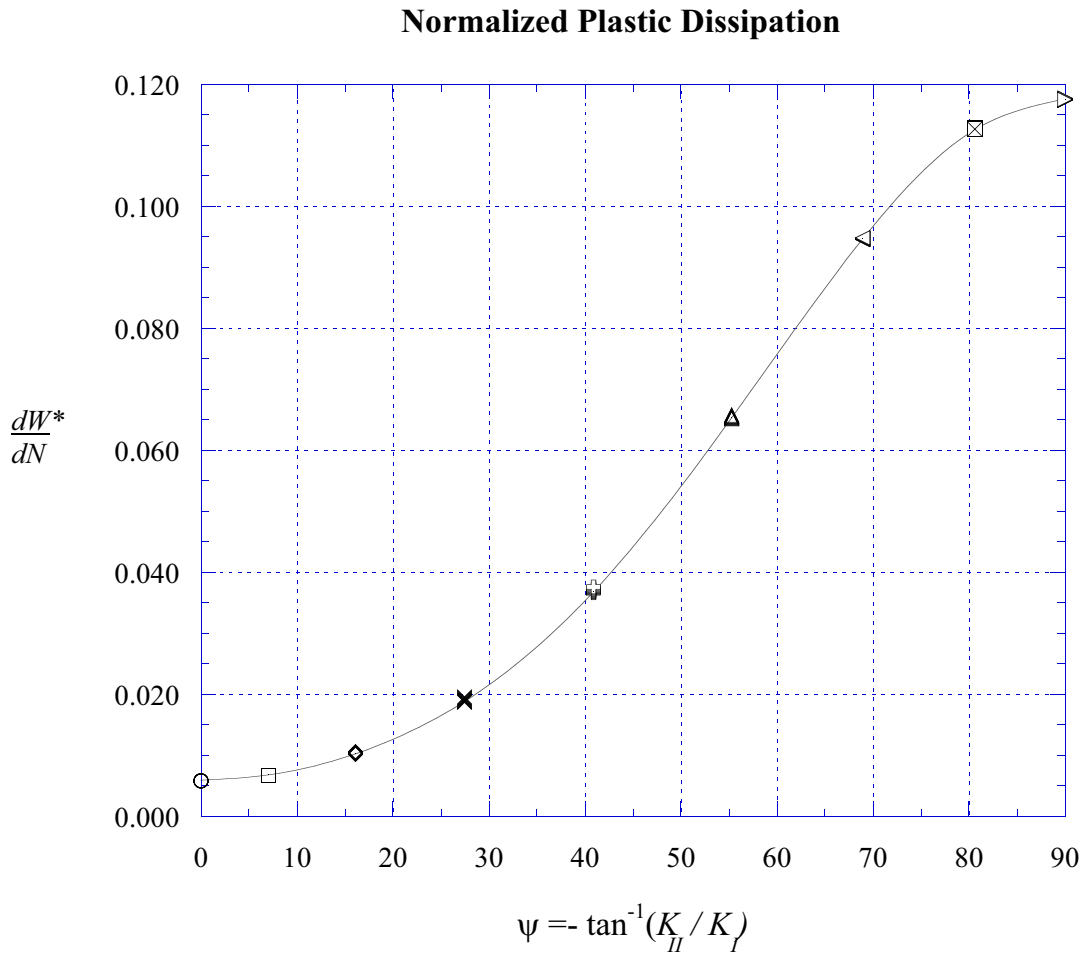


Figure 3.9: Dimensionless plastic dissipation $\frac{dW}{dN}^*$ vs. mode-mix ratio ψ for $E_t/E = 0$ and $\nu = 0.3$ in plane strain.

3.4.5 Effect of Plastic Constraint

A family of curves showing the dimensionless plastic dissipation as a function of mode-mix over the full range of Poisson's ratio and for $E_t/E = 0$ is shown in Figure 3.10. Results are also shown for plane stress, although these are independent of Poisson's ratio.[4] As shown in Figure 3.10, the plastic dissipation is greatest in plane stress, and decreases with increasing plastic constraint (i.e, increasing Poisson's ratio in plane strain). This result might be expected based on the relative plastic zone sizes in plane stress and plane strain, and is in keeping with the mode I results of [1]. The plot of Figure 3.10 also contains the master curve of Figure 3.9, which corresponds to $\nu = 0.3$ in plane strain. Evidently, changes in Poisson's ratio (i.e., plastic constraint) result in roughly uniform shifts of the master curve, although the magnitude of such shifts decreases with increasing plastic constraint. An important result is that for $\nu \geq 0.3$, values of $\frac{dW}{dN}^*$ vary by less than 0.5%. Thus, for all values of the mode-mix ratio, the effect of Poisson's ratio on $\frac{dW}{dN}^*$ is negligible for typical ductile metals where $\nu \geq 0.3$.

3.4.6 Effect of Hardening Modulus

Numerical results for $\frac{dW}{dN}^*$ vs. ψ are plotted in Figure 3.11 over the full range of E_t/E and for $\nu = 0.3$. The case of $E_t/E = 0$ (elastic-perfectly plastic response) coincides with the master curve of Figure 3.9, and represents an upper bound on the plastic work per cycle in plane strain. As should be expected, $\frac{dW}{dN}^*$ decreases with increasing hardening modulus, and approaches zero for the case of $E_t/E = 1$ (purely elastic response). Thus, for all values of mode-mix, the effect of increasing material hardening is to decrease the plastic work. In

[4] Note that for the case of $\nu = 0$, plane stress and plane strain are equivalent only in the elastic regime; for the case of plane strain, the incompressible response assumed and the classical plasticity formulation results in constrained yielding in the elastic-plastic regime.

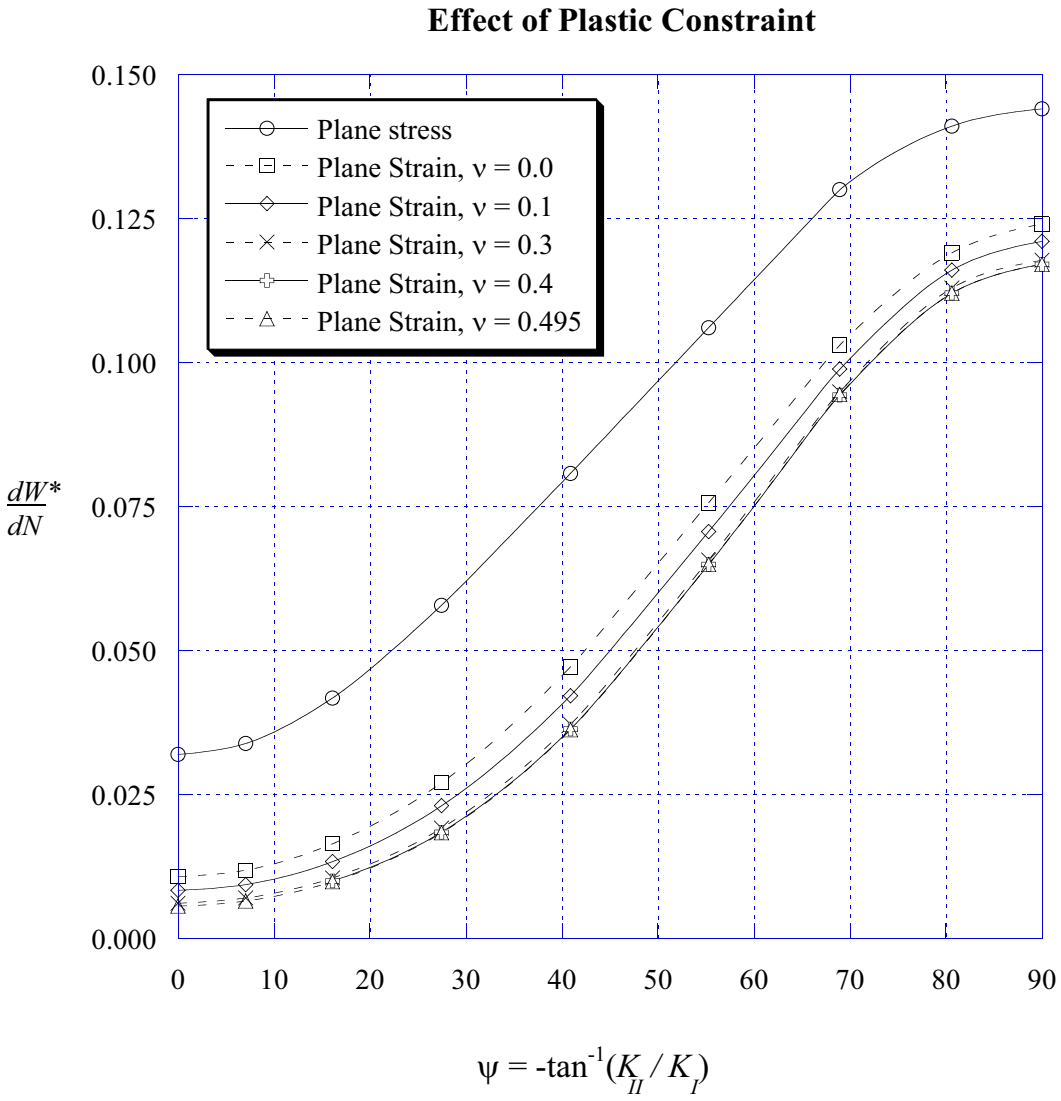


Figure 3.10: Dimensionless plastic dissipation energy $\frac{dW}{dN}^*$ vs. mode-mix ψ in plane stress and plane strain.

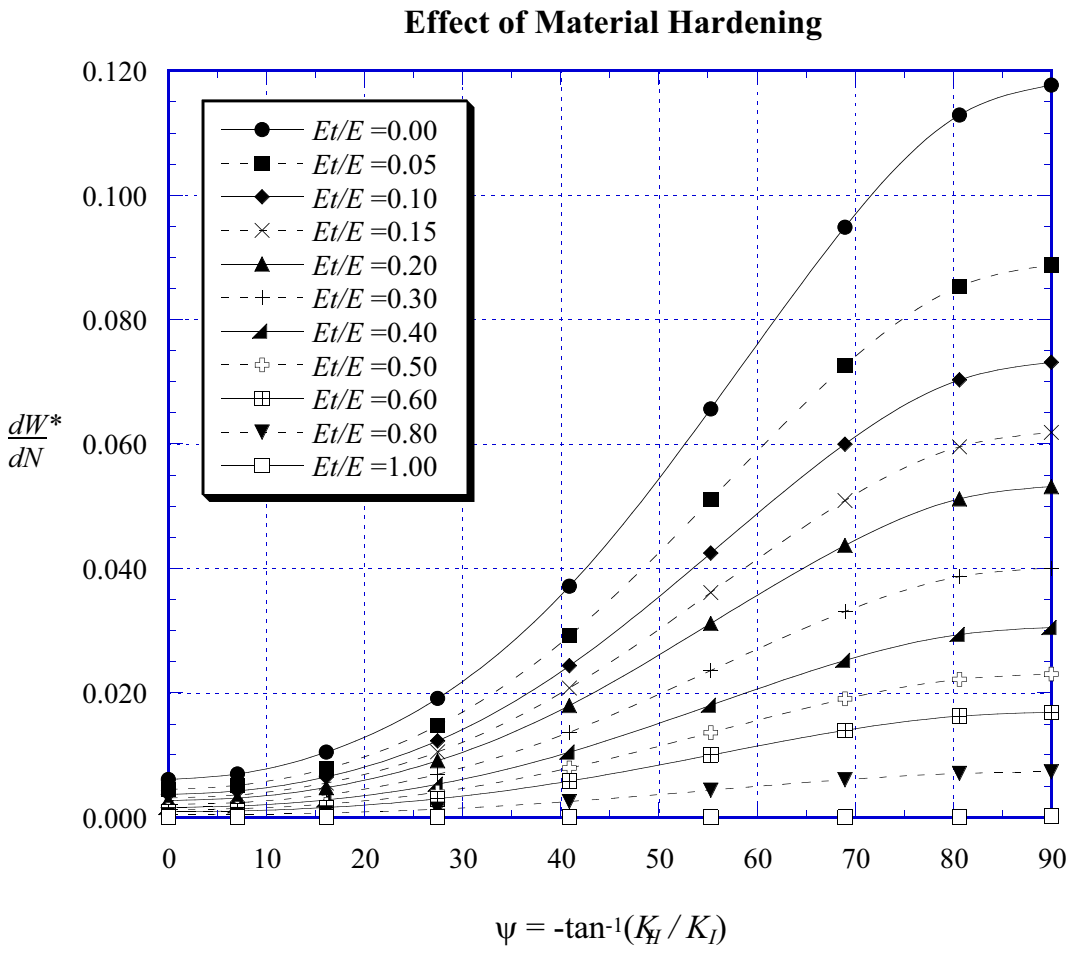


Figure 3.11: Effect of the tangent modulus ratio on $\frac{dW}{dN}^*$ vs. ψ for matching layers when $\nu = 0.3$.

an absolute sense, the results of Figure 3.11 indicate that $\frac{dW}{dN}^*$ is more sensitive to hardening ratio for high values of ψ. On the other hand, the effect of mode-mix is substantially more pronounced for low values of E_t/E, which is typical of ductile metals.

3.4.7 Effect of Specimen Geometry

The effects of hardening modulus on the dimensionless plastic dissipation have been considered for the case of mode I loading (C(T) specimen geometry) in [1]. Different specimen geometries and loading typically exhibit slight differences in both the shape and size of the crack tip plastic zones, which is typically attributed to differences in "T-stress" at the crack tip [45]. In order to investigate the sensitivity of $\frac{dW}{dN}^*$ to specimen geometry, both the current results for mode I loading and those of [1] are plotted verses E_t/E for both plane stress and plain strain ($\nu = 0.3$) in Figure 3.12. The most measurable difference is for the case of $E_t/E = 0$ in plane stress; however, this difference decreases with increasing hardening modulus, and appears to be negligible in plane strain. Hence, results suggest that specimen geometry has only a limited effect on the total plastic dissipation during plane strain fatigue crack growth in ductile solids.

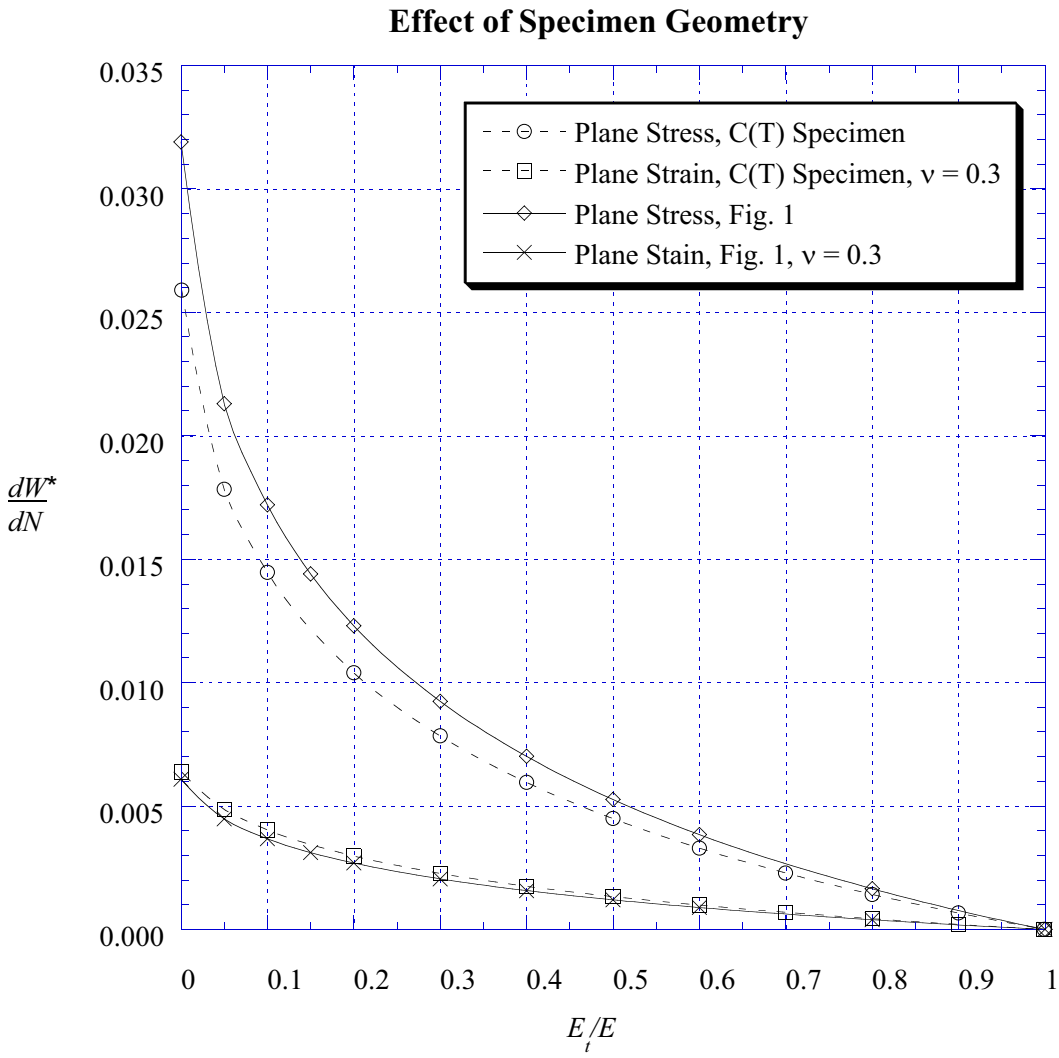


Figure 3.12: Comparison of current specimen results in mode I and C(T) results from [1].

Chapter 4

Plastic Mismatches

4.1 Modeling Procedure

A plastic mismatch is a difference in the hardening modulus and/or yield strength. These systems occur in layered or functionally graded materials where the elastic properties are the same and the plastic properties are altered through material processing. Because the analytical solutions presented in Chapter 3 are only dependent on the elastic properties and loading, the same equations can be used for interpreting the case for a plastic mismatch. A schematic of the different property mismatches is shown in Figure 4.1. Figure 4.1a shows the same hardening modulus with different yield strength and Figure 4.1b shows the same yield strength with a mismatch in hardening modulus. Both cases are considered in this chapter.

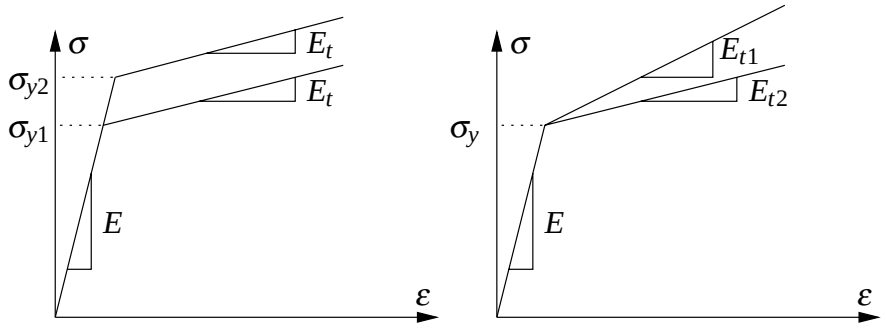


Figure 4.1: A schematic of different plastic property mismatches.

4.1.1 Definition of Yield Strength Mismatch

In this thesis, a yield strength mismatch is defined in terms of the dimensionless parameter

$$\hat{\sigma} = \frac{\sigma_{y1} - \sigma_{y2}}{\sigma_{y1} + \sigma_{y2}}, \tag{4.1}$$

where all possible values of strength mismatch are bounded between $-1 < \hat{\sigma} < 1$. If the top layer of the specimen is largely stronger than the bottom layer then $\hat{\sigma} \to 1$. Likewise, if the top layer is much weaker than the bottom layer then $\hat{\sigma} \to -1$. The case when $\hat{\sigma} = 0$ means there is no strength mismatch in the material. The cases considered herein use an elastic-perfectly plastic model when considering the strength mismatch. Also, all cases in this chapter use $\nu = 0.3$ in plane strain.

4.1.2 Definition of Hardening Modulus Mismatch

The ratio E_t/E is already dimensionless; as such, two parameters are needed to define the design space for a mismatch in the tangent modulus: E_{t1}/E and E_{t2}/E. To this end, full consideration of the design space requires varying each ratio independently. The higher values of E_t/E are more for academic completion than real application, because most duc-

tile metals have fairly low hardening moduli ($E_t/E \leq 0.1$). Since there are three independent variables (ψ, E_{t1}/E and E_{t2}/E) and one dependent variable ($\frac{dW}{dN}^*$), a single plot will no longer suffice to map out the plastic dissipation energy. Instead, a family of curves is needed to reflect the contributions of each variable.

4.1.3 Numerical Models

The numerical modeling is the same as that outlined in Chapter 3 with the only difference being the plasticity options for each layer. The plastic properties of each layer have to be defined independently. Each layer was changed for the FEA using a script similar to the one found in Appendix C.

4.2 Numerical Results and Discussion

4.2.1 Non-Dimensionalization

A modification of equation (3.11) is necessary for a more meaningful presentation of results. If normalized by the stronger material, the quantity $\frac{dW}{dN}^*$ will become very large as $\hat{\sigma} \to \pm 1$. As the yield strength used to normalize the data becomes large, then the quantity $\frac{dW}{dN}^*$ also becomes large according to equation (3.11). To alleviate this, $\frac{dW}{dN}$ can be normalized with respect to the weaker material as

$$\frac{dW}{dN}^* = \frac{\bar{\sigma_y}^2}{E\Delta\mathscr{G}^2}\frac{dW}{dN}, \tag{4.2}$$

where $\bar{\sigma_y} = \min(\sigma_{y1}, \sigma_{y2})$. Using the minimum yield strength makes for an easier graphical representation of the extreme values of $\hat{\sigma}$. Also, the weaker material controls the small

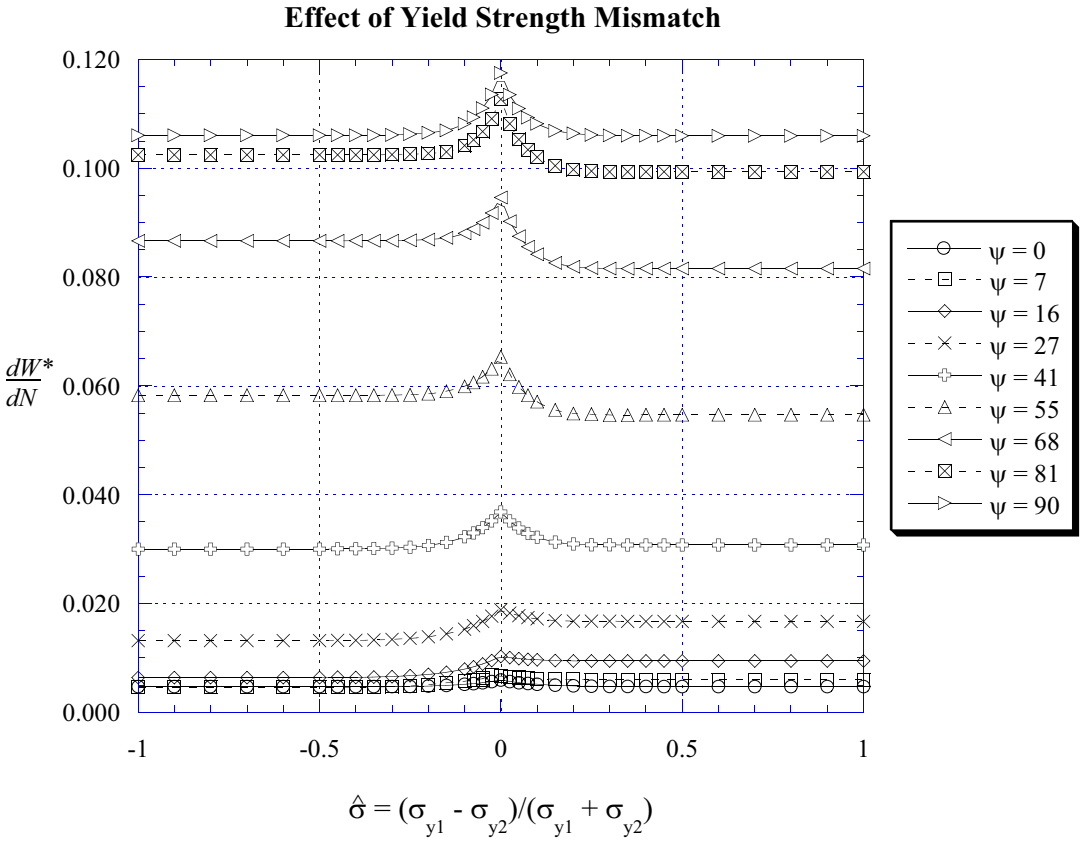


Figure 4.2: Illustration of the asymptotic effect of high strength mismatches.

scale yielding assumption. To generate results, the top layer would be assigned a fixed yield strength and the bottom would be assigned successively stronger materials to generate results for the values of $\hat{\sigma} < 0$. Likewise, to generate values of $\hat{\sigma} > 0$, the bottom layer is given a fixed yield strength and the top layer is increased in strength. This ensures small scale yielding because the plasticity will always be decreasing.

4.2.2 Yield Strength Mismatches

Using the new definition of the normalized plastic dissipation defined in equation (4.2), a plot of the plastic dissipation vs. yield strength mismatch is shown in Figure 4.2. A number of physical insights are evident in Figure 4.2. First, the effect of a strength mismatch is small compared to the effect of mode-mix. Second, there is an asymptotic effect of the yield strength mismatch. In other words, the value of $\frac{dW}{dN}^*$ for extreme values of the yield strength mismatch are the same as the values for $|\hat{\sigma}| \geq 0.25$ which is equivalent to a yield strength ratio of 5:3 (≈ 1.67).

For all values of ψ, the effect of $\hat{\sigma}$ is limited to the region of $-0.25 < \hat{\sigma} < 0.25$, after which, $\frac{dW}{dN}^*$ exhibits asymptotic behavior. As the strength mismatch continues to increase, there is a negligible contribution of plastic work from the stronger material. Thus, whenever a material is at least 1.67 times as strong as the other, the plastic dissipation energy is dominated by the weaker material. Figure 4.2 also shows the plastic work per cycle is not symmetric about $\hat{\sigma} = 0$ except for when $\psi = 0^\circ$ and $\psi = 90^\circ$.

The normalized plastic dissipation energy $\frac{dW}{dN}^*$ is plotted vs. ψ and several values of E_t/E and $\hat{\sigma}$ in Figure 4.3.The case of pure mode I or pure mode II is expected to be indifferent to whether the top is stronger or the bottom is stronger. This can easily be shown by symmetry arguments of the problem. However, as the mode becomes mixed from mode I, the values of $\frac{dW}{dN}^*$ are *higher* when $\hat{\sigma} > 0$. Conversely, as the mode becomes mixed from pure mode II, the values of $\frac{dW}{dN}^*$ are *lower* when $\hat{\sigma} > 0$. This means there is another value of ψ where $\frac{dW}{dN}^*$ is symmetric about the point $\hat{\sigma} = 0$. To illustrate this point, $\frac{dW}{dN}^*$ is plotted vs. ψ for $\hat{\sigma} = 0$ and $\hat{\sigma} = \pm 1$ in Figure 4.4. This shows the two extreme values of $\hat{\sigma}$ crossing near $\psi = 41^\circ$, which, coincidentally, is very close to the mode for the four point bend specimen [49]. As the absolute values of $\hat{\sigma}$ become closer to zero, the point

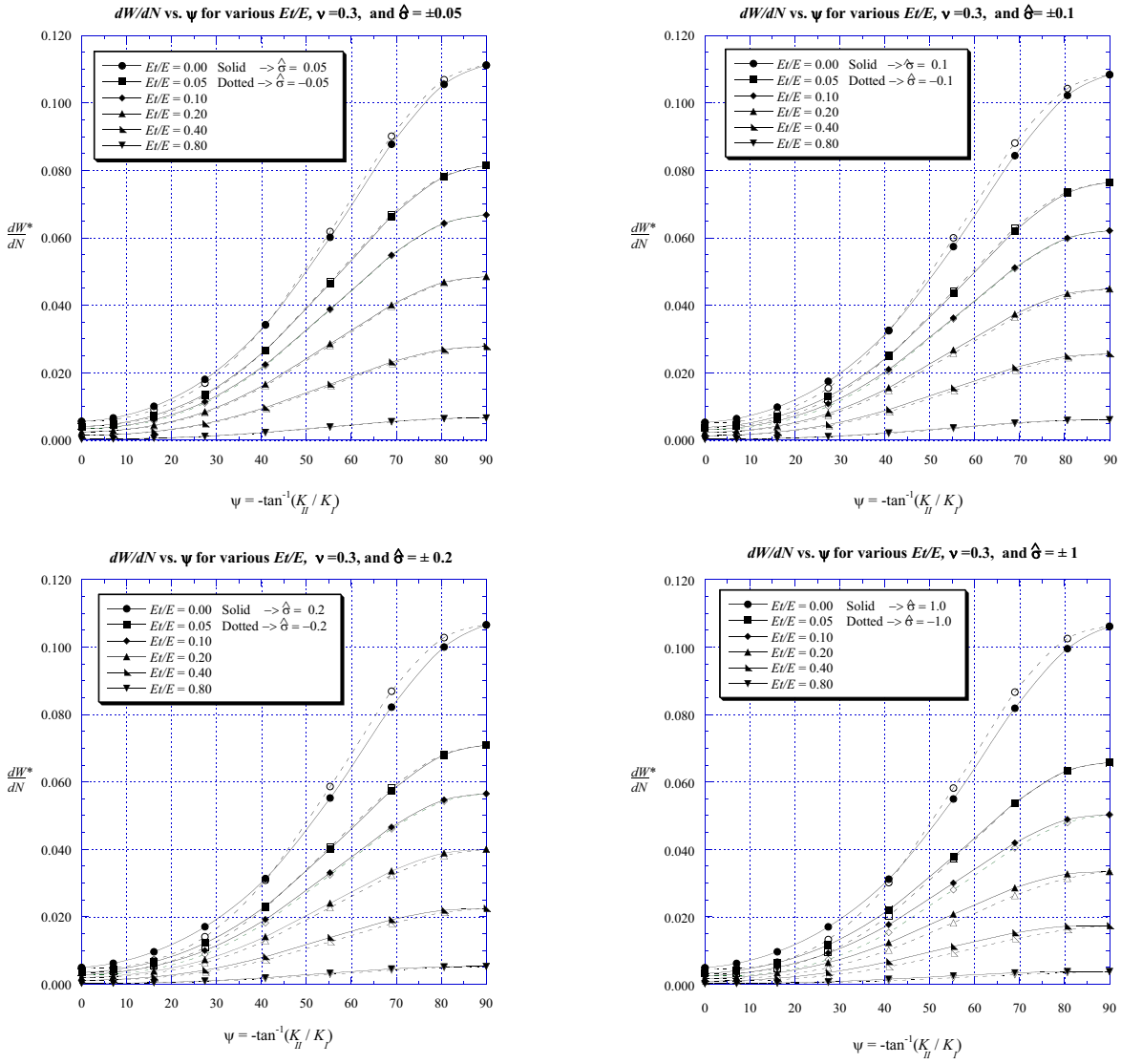


Figure 4.3: Plastic work shown for a combination of yield strength mismatches.

of intersection moves to a different location (see Figure 4.5 where $\hat{\sigma} = \pm 0.05$).

In addition to the crossing points, the plots of Figure 4.3 illustrate that the effect of yield strength mismatch is negligible compared to mode-mix. Even the extreme yield strength mismatches still follow the same shape and roughly the same magnitude of the homogeneous plot of Figure 3.11. The existence of these crossing points has very little consequence since the overall effect of the yield strength mismatch on the plastic dissipation energy is dominated by the mode. The crossing point plots do, however, illustrate the asymmetry of the mixed mode problem.

4.2.3 Hardening Modulus Mismatch

A second case of plastic mismatch, illustrated in Figure 4.1b, has the same yield strength between layers and different hardening ratios, E_t/E. Figure 4.6 shows a family of plots, each with a family of curves, with $\frac{dW}{dN}^*$ plotted vs. ψ for different combinations of E_{t1}/E, and E_{t2}/E. The effect of E_{t1}/E for fixed E_{t2}/E increases with the mode II component. It is easy to note that each of these plots contains a curve which matches that of Figure 3.11 when the corresponding tangent modulus ratios are equal. Also, the first plot of Figure 4.6 contains the curve from the master plot of Figure 3.9. When the tangent modulus ratio of one material is unity, no plastic work is done in that material. As a result, all the work is being done in the softer material. It should be noted that even with a unity tangent modulus ratio in one material, the quantity of plastic work is not half the work when the tangent moduli are equal. The interface is still elastic and acts to distribute the loads differently than if the two materials were the same. As the tangent modulus ratio of one material departs from unity, more plastic work is allowed to occur.

In Figure 4.6, the effects on $\frac{dW}{dN}^*$ become more pronounced as the hardening of the

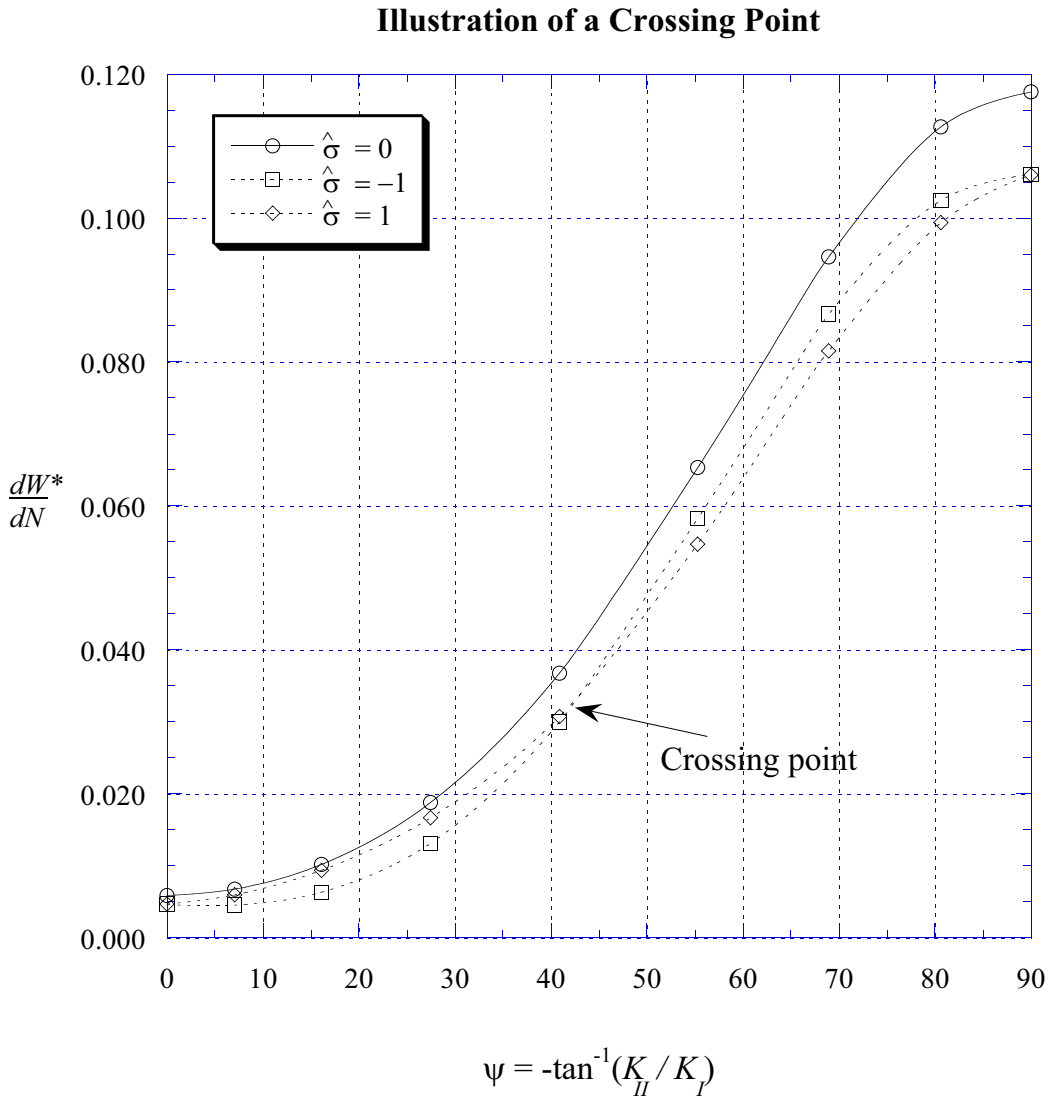


Figure 4.4: An illustration of asymmetric properties of a strength mismatch when $E_t/E = 0$ and $\nu = 0.3$.

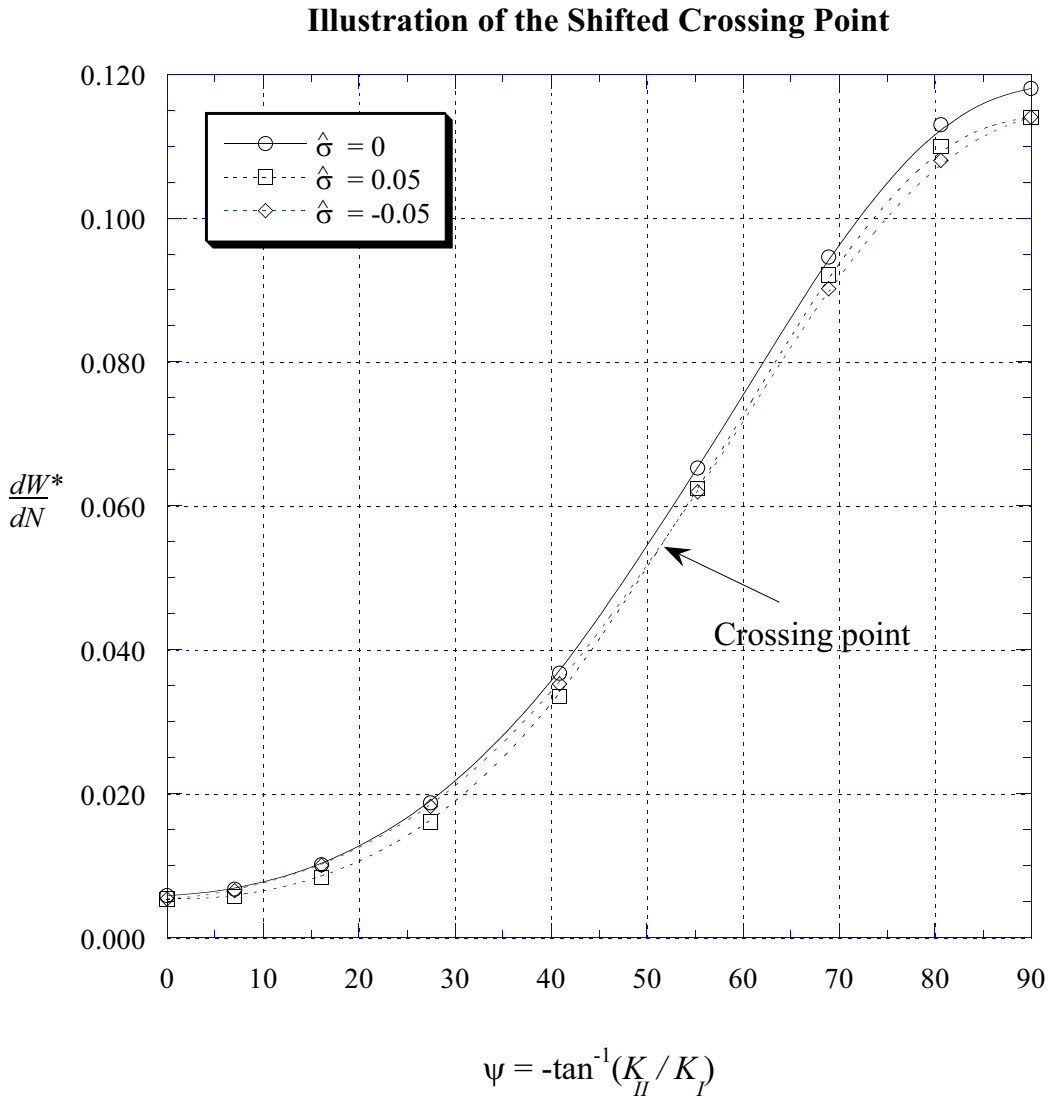


Figure 4.5: Translation of crossing points for different strength mismatches when $E_t/E = 0$ and $\nu = 0.3$.

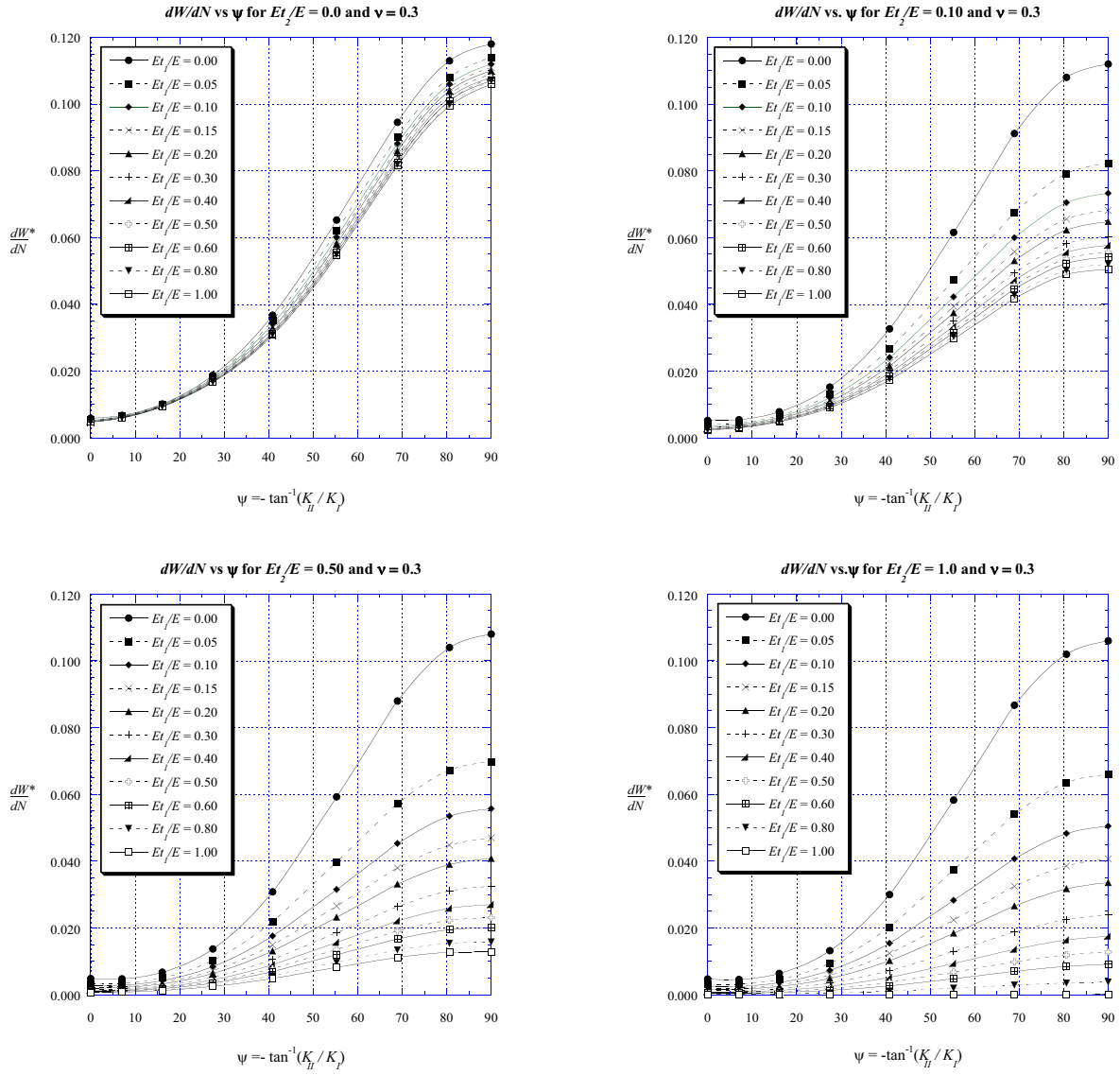


Figure 4.6: Family of plots showing $\frac{dW}{dN}^*$ in mixed mode with differing tangent modulus ratio (E_{t1}/E and E_{t2}/E) values.

bottom layer is increased, i.e. $E_{t2}/E \to 1$. In the extreme case when $E_{t2}/E = 1$, the bottom material does not yield. This is the same physical result as having $\hat{\sigma} = -1$, just as $E_{t1}/E = 1$ corresponds to $\hat{\sigma} = +1$. This is confirmed by comparison of Figures 4.4 and 4.7. These plots are of different FEA runs, the first being an analysis of strength mismatches and the second being an analysis of a hardening modulus mismatch. It can also be noted from Figure 4.7 that the same "crossing point" phenomenon exists for mismatches in hardening modulus as that for strength mismatches.

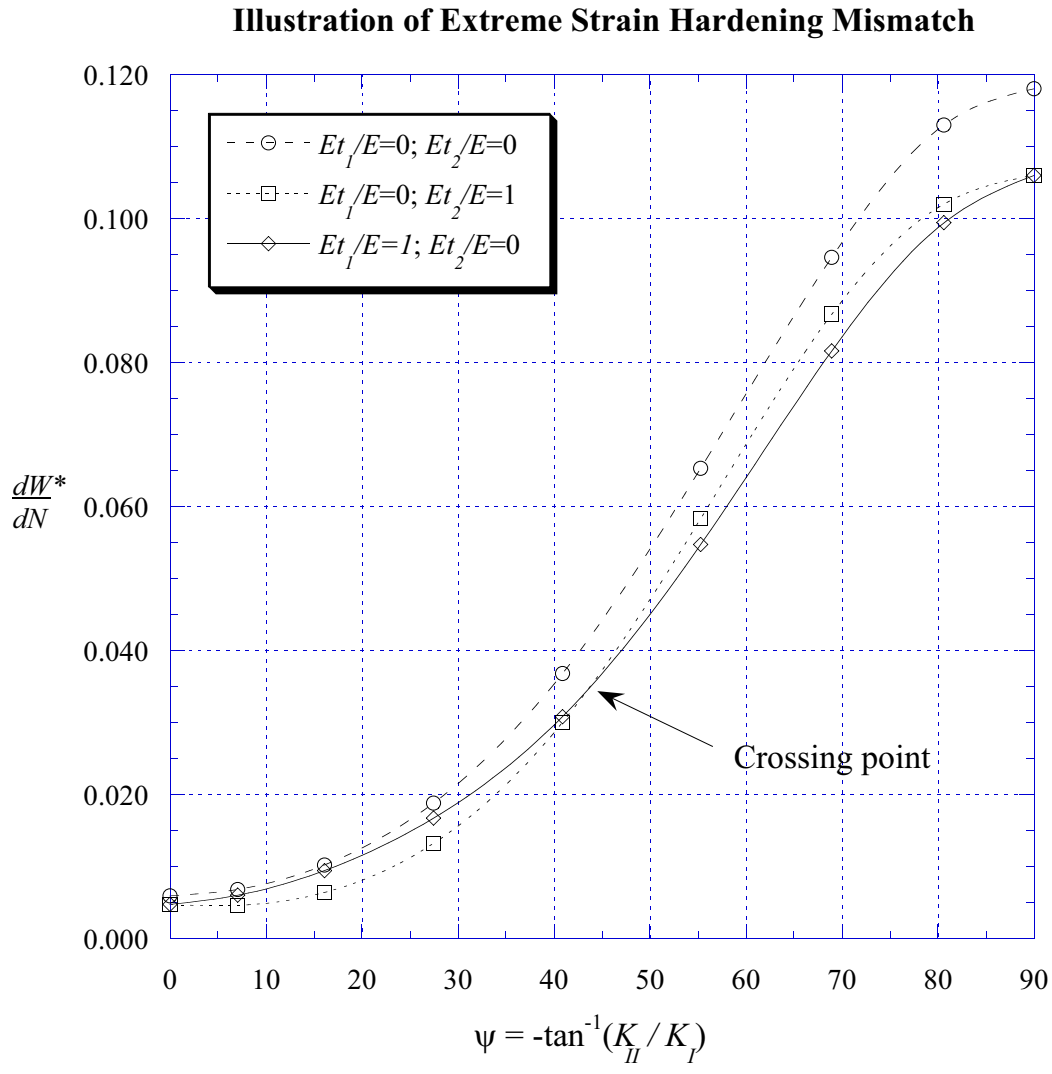


Figure 4.7: Effect of an extreme hardening modulus mismatch when $\hat{\sigma} = 0$ and $\nu = 0.3$.

Chapter 5

Elastic Mismatches

5.1 Analytical Models

The previous chapters considered an elastically homogeneous material with mismatches in only plastic properties. An elastic mismatch (shown in Figure 5.1) is significantly more complicated. Like any other plane elasticity problem, the elastic solution must satisfy the equations of elasticity. To this end, numerous studies have been performed concerning the asymptotic analysis of bonded bimaterial interfaces. The accepted conclusion is that the stress fields near an interface crack tip have oscillatory behavior in the singular dominated zone. This result, however non-physical it may seem, is important to quantifying a mode-mix ratio applicable to the general case. The solution procedure used to report the mode-mix is presented by Suo and Hutchinson [43, 44].

When there exists a mismatch in the elastic properties, the elastic modulus, shear modulus, and Poisson's ratio can all differ. Since only two of these parameters are independent, an elastic mismatch can be rewritten in terms of two parameters, as originally observed by

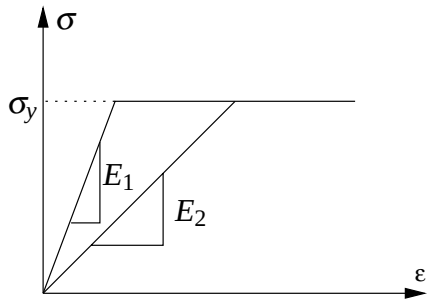


Figure 5.1: Schematic of an elastic mismatch with equal plastic properties (both $E_t/E = 0$ and $\hat{\sigma} = 0$).

Dundurs [50]. The Dundurs' parameters used in this research are defined as

$$\alpha = \frac{\Gamma(\kappa_2+1)-(\kappa_1+1)}{\Gamma(\kappa_2+1)+(\kappa_1+1)} \tag{5.1}$$

$$\beta = \frac{\Gamma(\kappa_2-1)-(\kappa_1-1)}{\Gamma(\kappa_2+1)+(\kappa_1+1)} \tag{5.2}$$

where $\kappa = 3-4\nu$ for plane strain and $\kappa = (3-\nu)/(1+\nu)$ for plane stress. Furthermore, the ratio of the shear moduli is defined as $\Gamma = \frac{\mu_1}{\mu_2}$ where the subscripts refer to the respective layers in Figure 3.1. The parameter α can also be written as

$$\alpha = \frac{\bar{E}_1 - \bar{E}_2}{\bar{E}_1 + \bar{E}_2}, \tag{5.3}$$

where $\bar{E} = E/(1-\nu^2)$ for plane strain and $\bar{E} = E$ for plane stress. The permissible values describing an elastic mismatch are $-1 < \alpha < 1$ and $\frac{\alpha-1}{4} < \beta < \frac{\alpha+1}{4}$. Whenever $\alpha < 0$ the top material is more compliant. Likewise, whenever $\alpha > 0$ the top material is more stiff, and there is no mismatch when $\alpha = 0$. Also, switching the layers results in a sign change of α. The design space for all possible elastic material mismatches is shown in Figure 5.2.

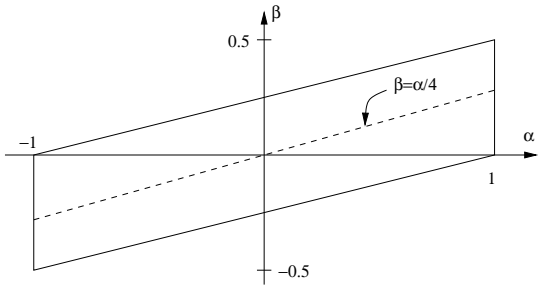


Figure 5.2: Permissible values for Dundurs' parameters in plane strain.

The quantity $\beta = \frac{\alpha}{4}$ corresponds to both materials having a Poisson's ratio $\nu = \frac{1}{3}$ in plane strain, which is convenient since the results shown in Figure 3.10 indicate no significant difference in $\frac{dW}{dN}^*$ after $\nu \geq 0.3$. For all subsequent analyses, $\nu = \frac{1}{3}$ and for both layers so $\beta = \frac{\alpha}{4}$. The following table gives a physical interpretation of the Dundurs parameter α that is useful in interpreting the data presented in this chapter.

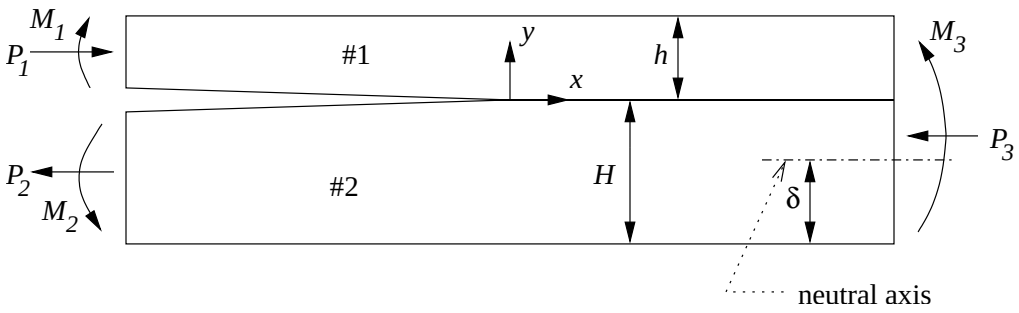
α	-0.8	-0.6	-0.4	-0.2	0.0	0.2	0.4	0.6	0.8
$\frac{E_1}{E_2}$	$\frac{1}{9}$	$\frac{1}{4}$	$\frac{3}{7}$	$\frac{2}{3}$	1	$\frac{3}{2}$	$\frac{7}{3}$	4	9

5.1.1 Equivalent loading and superposition

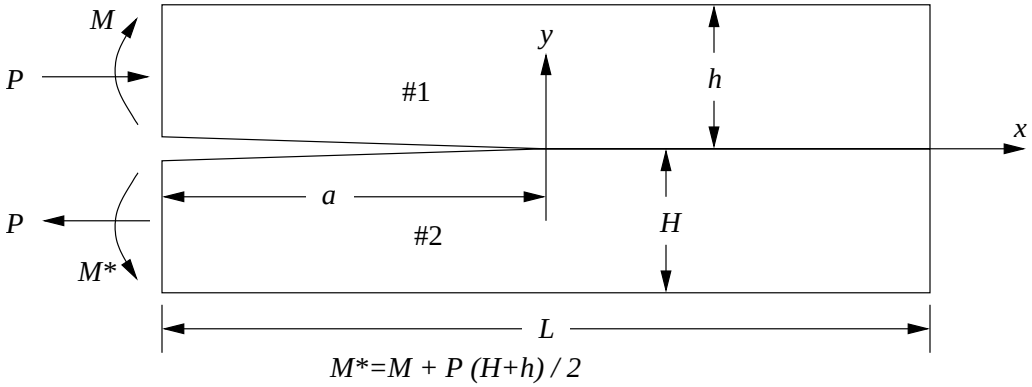
Consider the geometry of Figure 5.3b as an equivalent loading on the same specimen used in the finite element runs (Figure 3.1). Superposition arguments can be used to simplify the loading into two load parameters (P and M) as shown in equations (5.4) and (5.5) [43].

$$P = P_1 - C_1 P_3 - C_2 \frac{M_3}{h} \tag{5.4}$$

$$M = M_1 - C_3 M_3 \tag{5.5}$$



(a) General loading conditions for an interface crack.



(b) Generalized self equilibrating specimen shown with equal layer thickness.

Figure 5.3: A generalized mixed mode specimen and corresponding equivalent loading obtained by superposition [43].

where,

$$\begin{aligned} C_1 &= \frac{\Sigma}{A_o} \\ C_2 &= \frac{\Sigma}{I_o}\left(\frac{1}{\eta}-\Delta+\frac{1}{2}\right) \\ C_3 &= \frac{\Sigma}{12I_o}. \end{aligned} \tag{5.6}$$

Here Σ is an elastic modulus mismatch defined as $\Sigma = \frac{c_2}{c_1} = \frac{1-\alpha}{1+\alpha}$, where $c_i = \frac{\kappa_i+1}{\mu_i}$. Also, the following terms are derived from elementary beam theory for a composite beam:

$$\begin{aligned} A_o &= \frac{1}{\eta}+\Sigma \\ I_o &= \frac{1}{3}\left\{\Sigma\left[3\left(\Delta-\frac{1}{\eta}\right)^2-3\left(\Delta-\frac{1}{\eta}\right)+1\right]+3\frac{\Delta}{\eta}\left(\Delta-\frac{1}{\eta}\right)+\frac{1}{\eta^3}\right\} \\ \Delta &= \frac{1+2\Sigma\eta+\Sigma\eta^2}{2\eta(1+\Sigma\eta)} = \frac{\delta}{h} \end{aligned} \tag{5.7}$$

where $\eta = \frac{h}{H} = 1$ for the case under consideration and δ refers to the offset of the neutral axis in a composite beam. Using equations (5.6) and (5.7) in equations (5.4) and (5.5) will give an equivalent loading for any generalized specimen.

5.1.2 Strain Energy Release Rate

The strain energy release rate reported in [43] for the geometry of Figure 5.3b is

$$\mathscr{G} = \frac{c_1}{16}\left[\frac{P^2}{Ah}+\frac{M^2}{Ih^3}+2\frac{PM}{\sqrt{AI}h^2}\sin\gamma\right] \tag{5.8}$$

with

$$\sin\gamma = 6\Sigma\eta^2(1+\eta)\sqrt{AI}$$

where $A = \frac{1}{1+\Sigma(4\eta+6\eta^2+3\eta^3)}$ and $I = \frac{1}{12(1+\Sigma\eta^3)}$. For an elastically matched geometry, $\Sigma = 1$ and $\eta = 1$ which yields $\sin\gamma = \sqrt{\frac{3}{7}}$. Making the appropriate substitutions yields

the previous results of equation (3.3).

5.1.3 Interface Stress Intensity Factors

In order to discuss interface stress intensity factors, it is necessary to introduce a bimaterial constant or an oscillation index ε defined as:

$$\varepsilon = \frac{1}{2\pi} \ln \frac{1-\beta}{1+\beta}. \tag{5.9}$$

The interface crack tip stress field have the form

$$(\sigma_{yy} + i\sigma_{xy}) \mid_{\theta=0} = \frac{(K_1 + iK_2) r^{i\varepsilon}}{\sqrt{2\pi r}} \tag{5.10}$$

The oscillations in the stress field occur because the asymptotic solution for the stress fields yield a complex solution in the form $K = K_1 + iK_2 = |K| e^{i\psi}$. The stress fields oscillate for nonzero values of ε, therefore, the ratio K_1/K_2 does not correspond directly to the ratio σ_{yy}/σ_{xy}. In order to define a consistent measure of mode-mix, it is useful to consider the quantity $Kh^{i\varepsilon}$, where

$$Kh^{i\varepsilon} = K_1 \cos(\varepsilon \ln h) - K_2 \sin(\varepsilon \ln h) + i[K_2 \cos(\varepsilon \ln h) + K_1 \sin(\varepsilon \ln h)]. \tag{5.11}$$

Here, h is any characteristic length, usually taken as the thickness of the thinnest layer. For the problem of Figure 5.3, the real and imaginary parts of $Kh^{i\varepsilon}$ are [32, 43]:

$$\mathrm{Re}[Kh^{i\varepsilon}] = \frac{p}{\sqrt{2}} \left[\frac{P}{\sqrt{Ah}} \cos\omega + \frac{M}{\sqrt{Ih^3}} \sin(\omega + \gamma) \right] \tag{5.12}$$

$$\mathrm{Im}[Kh^{i\varepsilon}] = \frac{p}{\sqrt{2}}\left[\frac{P}{\sqrt{Ah}}\sin\omega - \frac{M}{\sqrt{Ih^3}}\cos(\omega+\gamma)\right] \tag{5.13}$$

where $p = \sqrt{\frac{1-\alpha}{1-\beta^2}}$. Equations (5.12) and (5.13) contain a function $\omega(\alpha,\beta,\eta)$ that is tabulated in [43] and ranges from $37^\circ \leq \omega \leq 65^\circ$.

The phase angle or mode-mix ratio ψ is redefined as

$$\psi = -\tan^{-1}\left(\frac{\mathrm{Im}[Kh^{i\varepsilon}]}{\mathrm{Re}[Kh^{i\varepsilon}]}\right). \tag{5.14}$$

When there is no elastic mismatch or when $\beta = 0$, then the computation of equation (5.14) yields the same result as the previous definition of the mode-mix ratio in equation (2.5). It is also important to note that the value of the phase angle depends on the reference distance h. This implies that the mode will change depending on size of the specimen or if the stress intensity factors are normalized with respect to some other value (e.g. the length or the total height). However, this freedom of choice for the normalization length comes from the ability to define a phase transformation to change the normalization from one length h_1 to another length h_2 using a formula from [44],

$$\psi_2 = \psi_1 + \varepsilon \ln\left(\frac{h_2}{h_1}\right). \tag{5.15}$$

5.2 Numerical Models

The basic extraction of the plastic dissipation energy remains the same as in previous chapters, however, defining the mode is not as straight forward. The first challenge is extracting the $\omega(\alpha,\beta,\eta)$ function for use in the finite element code. The tabulated data in [43] con-

tains only values of α at intervals of 0.2 and values of β at intervals of 0.1 for $\eta = 1$. When using $\beta = \frac{\alpha}{4}$, the published values of ω only exist at 5 locations. The published values of ω were tabulated using numerical solutions to cumbersome integral equations, and an alternative method of approximating them was desired. Some built in functions in ABAQUS allowed for the extractions of the stress intensity factors, from which ω can be inferred.

5.2.1 Interaction Integrals

ABAQUS has a built in feature called the interaction integral. It is invoked using the *CONTOUR INTEGRAL, CONTOURS = N, TYPE = K FACTORS option similar to the computation of the J-Integral. The output of the interaction integral is given as K_1 and K_2 which are the mode I and mode II stress intensity factors. However, when there is an elastic mismatch, this computation gives the real and imaginary parts of the complex stress intensity factors defined in equations (5.12) and (5.13).

The interaction integral method uses a normalization length of 1 as defined in the ABAQUS Theory Manual [51] and requires a transformation according to equation (5.15). Since the model reference height used herein is $h = 5$, the phase transformation is $\psi_2 = \psi_1 + \varepsilon \ln(h)$ where ψ_2 is the normalized mode-mix ratio and $\psi_1 = -\tan^{-1}\left(\frac{K_2}{K_1}\right)$ from the interaction integrals. Once ψ, K_1, and K_2 are known, ω can be found using a root finding method in conjunction with equations 5.12 and 5.13. Incidentally, this method for calculating ω agreed closely with the tabulated data found in [43].

5.2.2 Spanning All Modes

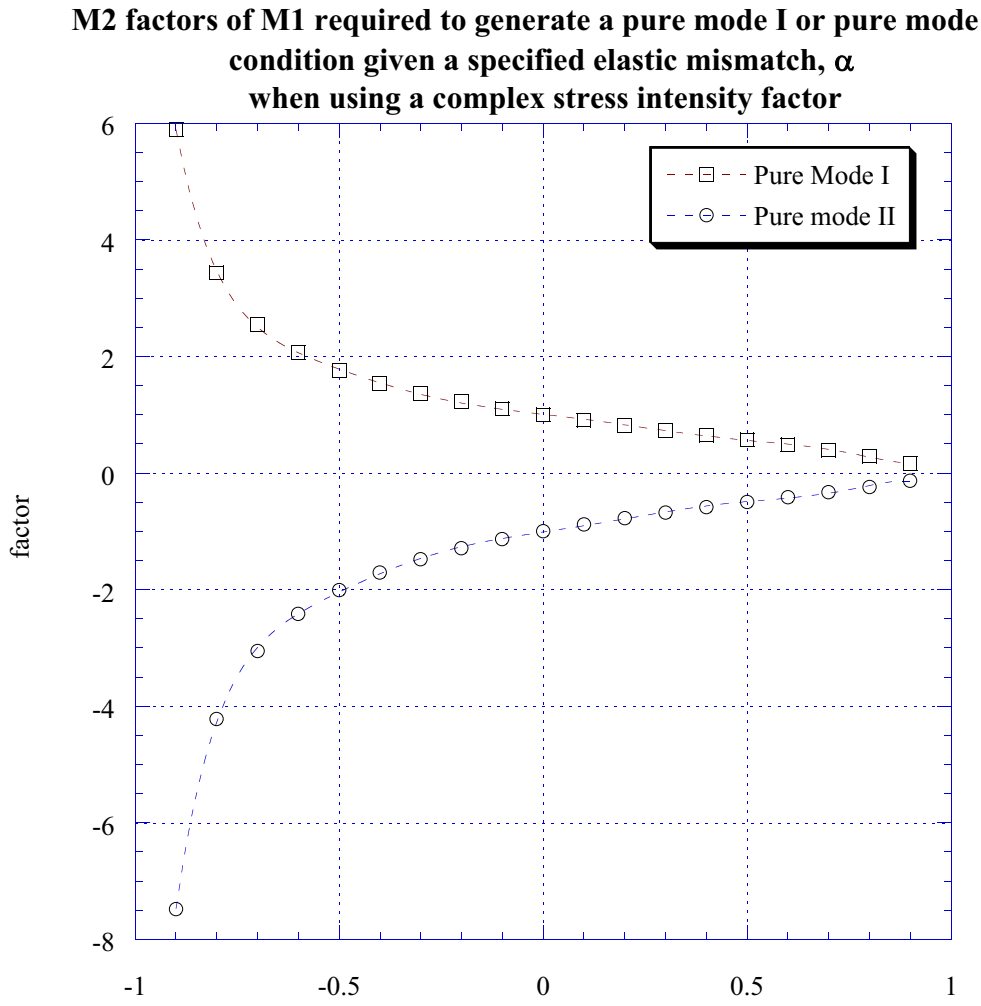
While being able to determine ω is a nice result, the real interest lies in trying to span all the modes for the current problem. Doing this requires applying bending moments on the

end of each layer in a ratio unique to each mode. There is no simplified expression for $\mathrm{Re}[Kh^{i\varepsilon}]$ or $\mathrm{Im}[Kh^{i\varepsilon}]$, so the choices for loading conditions for the pure mode I and pure mode II cases are not straightforward as they were in Chapter 3.

To determine a ratio of the loads $f = \frac{M_2}{M_1}$, a search algorithm, shown in the `findfactor` function of the master script in Appendix C, was used to determine the moments for each value of α. Only half the values of α had to be determined because values of f were inverted when α changed sign (i.e. $f(\alpha) = f(-\alpha)^{-1}$). Once the factor data were gathered, an 8th order polynomial regression was performed to quantify the trends. The results are interpreted as a factor of the bending moment M_1 (i.e. $f = \frac{M_2}{M_1}$) and shown in Figure 5.4. The regression data can then be used in a script to solve for the extreme loading conditions for any value of α. This technique can calculate the loading factors for any given mode accurately, however, once the results were fitted with the polynomial, some of the accuracy was sacrificed for convenience. Overall, the pure modes were calculated within a degree for all combinations of elastic mismatch. Dividing the difference between the mode II and mode I load ratio into even sections allowed generation of the full range of the mode-mix values reported in the plots. Table 5.1 shows the values of the mode-mix ratio (ψ) for each load ratio (f) and elastic mismatch (α). The case for no elastic mismatch corresponds closely to the results of Chapter 3, with the only difference coming from the slight errors due to the regression.

5.2.3 Constant Loading Amplitude

For comparison across modes for a given elastic mismatch, a constant magnitude for the stress intensity factors (or strain energy release rate) is necessary. To that end, some algebraic manipulation of the equations presented in this chapter will give a formula for the



Y = M0 + M1*x + ... M8*x^8 + M9*x^9	
M0	1.01
M1	-0.831
M2	-0.00187
M3	-3.01
M4	5.03
M5	8.32
M6	-12.7
M7	-10.1
M8	12.7
R	1.000

Y = M0 + M1*x + ... M8*x^8 + M9*x^9	
M0	-1.01
M1	1.06
M2	-0.143
M3	3.92
M4	-6.70
M5	-11.1
M6	17.0
M7	13.4
M8	-17.0
R	1.000

Figure 5.4: Loading conditions to generate pure mode I and pure mode II for different values of α in plane strain.

$\alpha = -0.9$		$\alpha = -0.8$		$\alpha = -0.7$		$\alpha = -0.6$		$\alpha = -0.5$	
f	$\psi(^\circ)$	f	$\psi(^\circ)$	f	$\psi(^\circ)$	f	$\psi(^\circ)$	f	$\psi(^\circ)$
-7.47	89.9	-4.269	89.6	-2.993	89.3	-2.417	90.0	-4.269	89.6
-5.99	82.0	-3.409	82.5	-2.382	81.5	-1.919	82.2	-3.409	82.5
-4.50	72.2	-2.549	72.9	-1.772	72.0	-1.420	72.6	-2.549	72.9
-3.02	60.8	-1.689	61.4	-1.161	60.8	-0.922	61.3	-1.689	61.4
-1.53	48.2	-0.829	48.8	-0.550	48.5	-4.24	48.8	-0.829	48.8
-0.049	35.6	0.031	36.0	0.061	36.1	0.075	36.1	0.031	36.0
1.436	24.1	0.890	24.3	1.672	24.7	0.573	24.5	0.890	24.3
2.920	14.4	1.751	14.4	1.283	14.9	1.070	14.6	1.751	14.4
4.405	6.42	2.611	6.2	1.893	6.9	1.570	6.5	2.611	6.2
5.889	0.04	3.471	0.2	2.504	0.4	2.068	0.0	3.471	0.2

$\alpha = -0.4$		$\alpha = -0.3$		$\alpha = -0.2$		$\alpha = -0.1$		$\alpha = 0$	
f	$\psi(^\circ)$	f	$\psi(^\circ)$	f	$\psi(^\circ)$	f	$\psi(^\circ)$	f	$\psi(^\circ)$
-1.729	89.6	-1.466	89.7	-1.265	89.3	-1.122	89.6	-1.010	89.7
-1.365	82.5	-1.152	81.8	-0.991	81.3	-0.875	81.5	-0.786	82.1
-1.000	72.8	-0.839	72.1	-0.716	71.5	-0.629	71.6	-0.561	72.0
-0.635	61.4	-0.525	60.7	-0.442	60.1	-0.382	60.0	-0.337	60.2
-0.271	48.7	-0.212	48.1	-0.167	47.7	-0.136	47.4	-0.112	47.3
0.094	35.9	0.101	35.6	0.107	35.3	0.111	35.0	0.112	34.7
0.458	24.2	0.415	24.2	0.382	24.1	0.357	23.7	0.337	23.3
0.823	14.3	0.728	14.5	0.656	14.5	0.604	14.2	0.561	13.7
1.187	6.2	1.042	6.5	0.931	6.7	0.850	6.4	0.786	5.9
1.552	0.3	1.355	0.1	1.205	0.4	1.097	0.2	1.010	0.2

Table 5.1: Load ratio and corresponding mode-mix ratio. The load ratios corresponding to the positive values of α can be determined by inverting the load ratio shown in this table.

bending moment M_1 as

$$M_1 = \frac{h^3|K|^2(3\alpha^2-4)(\beta^2-1)}{\sqrt{3}\sqrt{-h^3|K|^2(3\alpha^2-4)(-(6\alpha+7)(\alpha-1)^2+f^2(\alpha+1)^2(6\alpha-7)+2f(\alpha^2-1))(\beta^2-1)}}. \quad (5.16)$$

Similarly, equation (5.16) can be written in terms of the strain energy release rate because $|K|^2 = \mathscr{G}\bar{E}$. This equation is particularly useful when comparing the plastic zones because each picture of a plastic zone will have the same magnitude of loading for a given value of α. When $\alpha = \beta = 0$, equation (5.16) was reduced and used to generate the plots in Figures 3.4 - 3.6 in Chapter 3.

5.3 Numerical Results and Discussion

5.3.1 Normalizing the Data

Equation (3.11) uses the elastic modulus to define the dimensionless plastic dissipation energy, thus creating a choice of which moduli to use in the case of an elastic mismatch. Either modulus is useful and the choice of one or the other only changes the way the full range of the data is represented. For consistency, the top layer is used to normalize the data.

5.3.2 Effects of Elastic Mismatch

Figure 5.5 shows the response of the dimensionless plastic dissipation energy in plane strain and with $\beta = \alpha/4$ as a function of mode for $-0.8 \leq \alpha \leq 0.8$, and normalized with respect to the top layer. Figure 5.5 contains a curve for $\alpha = 0$ which is identical to the master curve from Figure 3.8. The effect of mode-mix is seemingly uniform across the values of

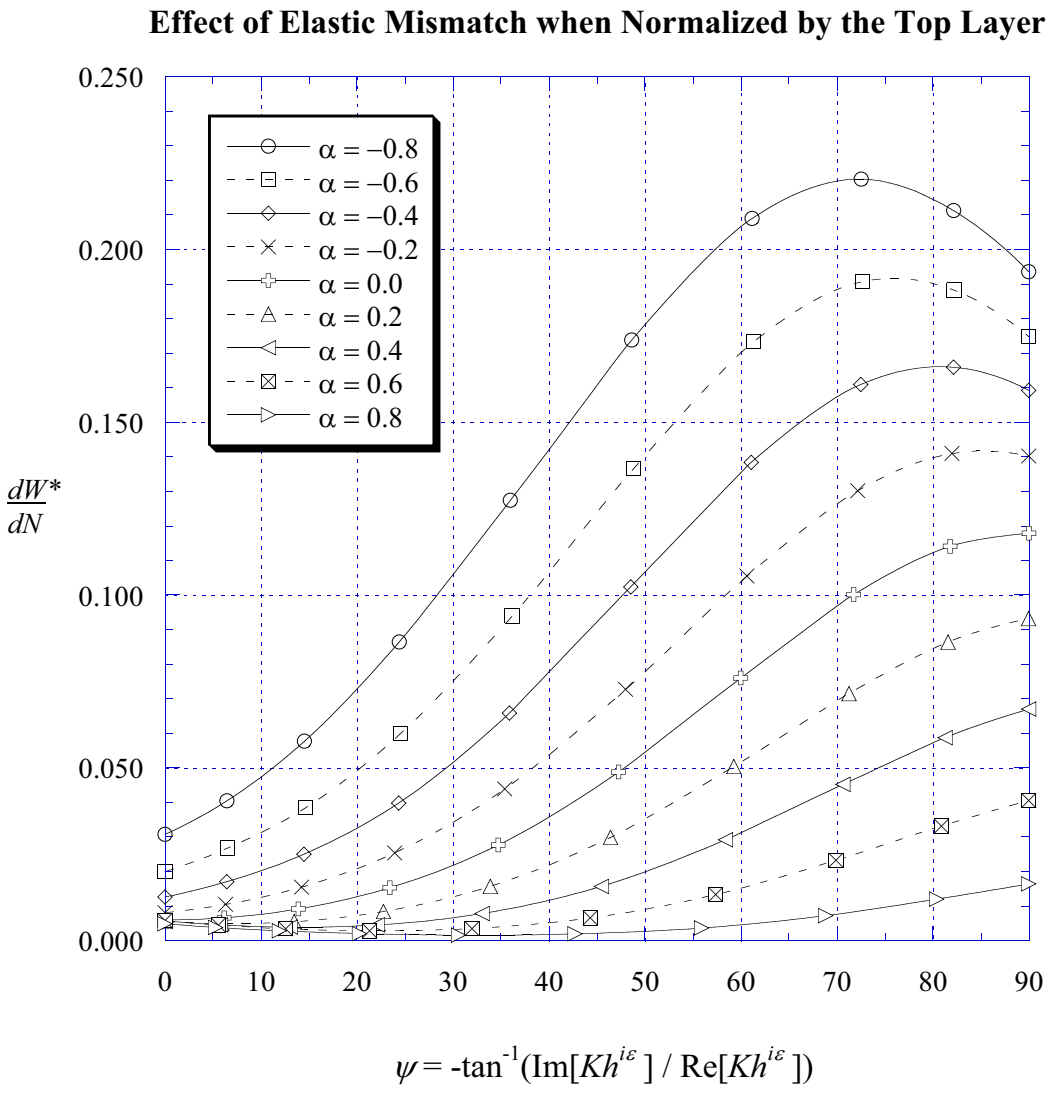


Figure 5.5: Dependence of plane strain $\frac{dW}{dN}^*$ on mode for a complete range of elastic modulus mismatch when $\beta = \frac{\alpha}{4}$ and normalized with respect to the elastic modulus of the top layer.

α for large mode II components. However, this uniformity diminishes as $\psi \to 0$. There are distinctive peaks for negative values of α when the top layer is more compliant than the bottom layer. Also, there are less distinguishable minima for the cases when $\alpha > 0$. Those maxima and minima locations are best explained with the aid of illustrations of the plastic zones from the finite element run.[1] The pictures of the plastic zones for α = -0.8, -0.4, 0.0, 0.4, 0.8 are found in Appendix A. The case of no elastic mismatch shows a geometrically symmetric plastic zone about the x-axis for both $\psi = 90^\circ$ and $\psi = 0^\circ$. Those plastic zones are symmetric only at the pure modes (when $\alpha = 0$) and occur when the plastic work is at either a maximum or a minimum value. The shape of the pure mode II plastic zone (as shown in Figure 3.6) corresponds to a maximum in the dimensionless plastic dissipation energy. Likewise, the shape of the pure mode I plastic zone (as shown in Figure 3.4) corresponds to a minimum in the dimensionless plastic dissipation.

The degree of symmetry of the plastic zone determines the proximity to an extrema of the plastic work, given no plastic mismatches. To show that this statement has validity, examination of the pictures in Appendix A.2 shows a shift in the symmetry when $72^\circ < \psi < 82^\circ$, with a closer resemblance to Figure 3.6 when $\psi = 82^\circ$. Remarkably, Figure 5.5 shows a maximum value of $\frac{dW}{dN}^*$ near $\psi = 82^\circ$. As a result, the extrema can be inferred from the geometry of the plastic zone. This technique works on all the cases presented in this chapter and provides some physical explanation for the extrema.

Figure 5.5 is normalized such that if the stiffness of the top layer is fixed, then as α increases, the bottom layer is becoming more compliant and the plastic dissipation energy is decreasing. Similarly, If the bottom layer material properties are fixed, then increasing the stiffness of the top layer will decrease the plastic dissipation. These conclusions are

[1] It should be noted that the plastic zones just illustrate the onset of yielding and do not necessarily predict the amount of plastic dissipation energy.

directly applicable to a scenario when a designer is given the choice of a substrate for a given material to be deposited.

The elastic mismatch and the mode-mix are the dominant factors driving the plastic dissipation energy, whereas the plastic mismatch plays a secondary role. This observation is apparent when comparing the amount of fluctuation of $\frac{dW}{dN}^*$ in the figures of Chapter 4 to that of Figure 5.5. However, the pictures of the plastic zones indicate that the effect of a high strength mismatch will influence the shape of the elastic mismatch curves (Figure 5.5) by eliminating the contribution of the plasticity of one layer or the other. A complete examination of all possible elastic and plastic mismatches is cumbersome and the results are to be determined through studies of specific layered material systems.

Chapter 6

Application of Results

6.1 Combined Elastic-Plastic Mismatch

Two common bimaterial systems, one of a brass-solder interface [39] and the other of a stainless steel-copper interface [32], have been analyzed to determine the response of $\frac{dW}{dN}^*$ under mixed-mode loading for both elastic and plastic mismatch conditions. Table 6.1 shows the material properties for these bimaterial systems, and the numerical results are plotted in Figure 6.1, where the first entry in the legend corresponds to the top layer. Since the yield strength mismatch $\hat{\sigma} > 1.67$, the yield strength mismatch has the same effect for each example case (see Section 4.2.2). Because no strain hardening rates were reported in [39, 32] the analysis neglected strain hardening and assumed an elastic-perfectly plastic response ($E_t = 0$).

The curves of Figure 6.1 are similar in shape to the curves of Figure 5.5. This similarity leads to the conclusion that the elastic mismatch is the dominant parameter influencing the shape and magnitude of the $\frac{dW}{dN}^*$ vs ψ curve when both elastic and plastic mismatches are present. The yield strength mismatch slightly changes the shape of the curve, so it does not

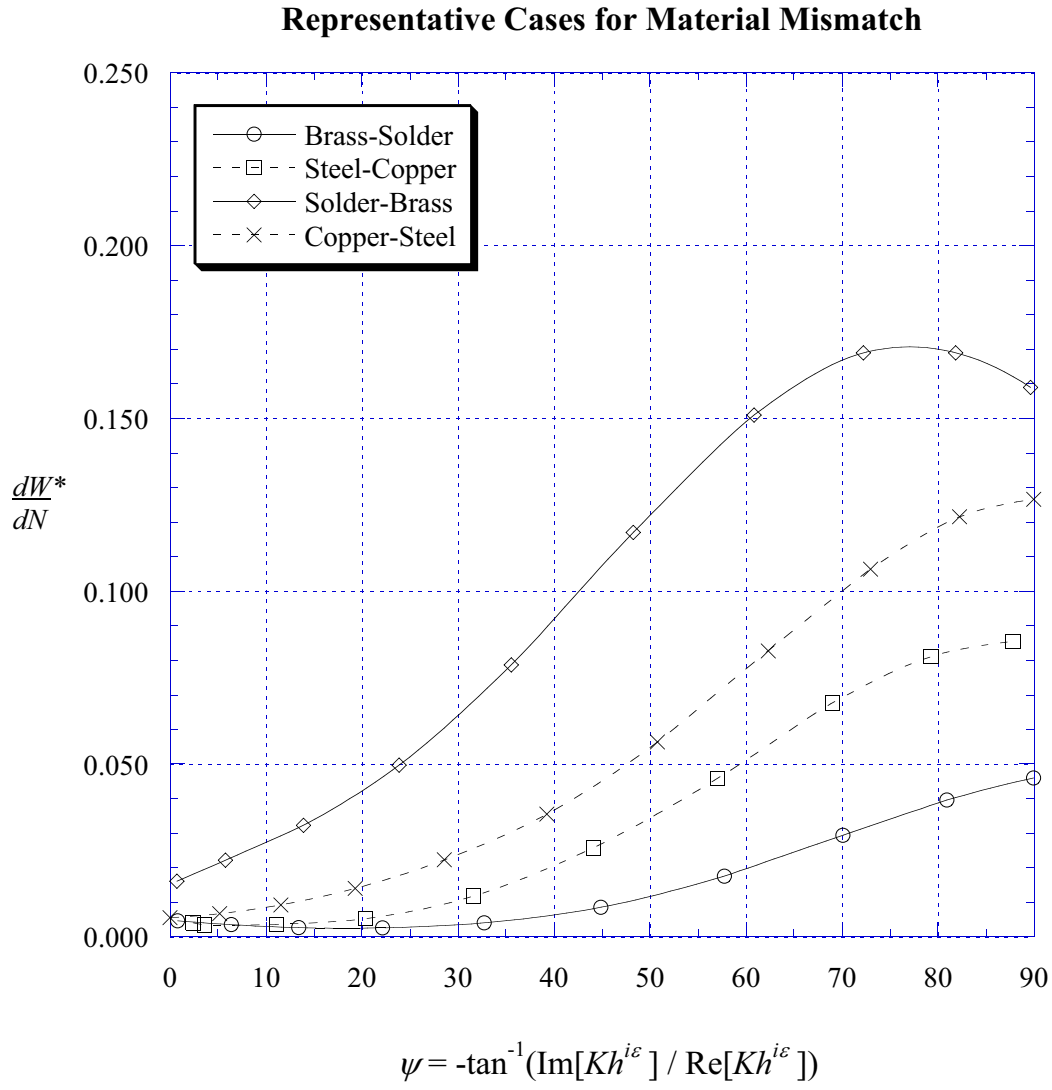


Figure 6.1: Representative curves of $\frac{dW^*}{dN}$ vs. ψ for two different material interfaces in plane strain when normalized by the top layer.

	Brass	63Sn–37Pb Solder
E	101 GPa	30 GPa
ν	0.35	0.324
σ_y	70 MPa	22.6 MPa
	α	± 0.3166
	β	± 0.0309
	$\hat{\sigma}$	± 0.576

	Stainless-Steel	Copper
E	190 GPa	120 GPa
ν	0.267	0.364
σ_y	260 MPa	70 MPa
	α	± 0.193
	β	$\pm 3.12 \times 10^{-4}$
	$\hat{\sigma}$	± 0.521

Table 6.1: Material properties for the bimaterial layers. The sign on the Dundurs' parameters and yield strength mismatch change depending on which layer is labeled as the top layer.

fit evenly within the curves of Figure 5.5. The effect of the yield strength mismatch on $\frac{dW}{dN}^*$ is small compared to the variation to the elastic modulus mismatch and the mode-mix. The results shown in Figure 5.5 also reaffirm the conclusion that the plastic dissipation energy decreases when the softer material is on the top layer ($\alpha < 0$). The interpretaion of this problem is reversed if the sign of the mode mix changes , since this is equivalent to flipping the problem upside down and switching M_1 and M_2. The magnitude of ψ will change when a problem is reversed due to the asymmetry of the elastic layers.

6.2 Comparison with Previous Literature

The numerical results show that the dominant factor in the dimensionless plastic dissipation energy is the mode-mix ratio. Higher modes have more plastic dissipation energy, which would lead us to believe, according to equation (1.1), that the fatigue crack growth rate would be higher in a mixed mode condition than in mode I. It is known that this is not true and that the mode I fatigue crack growth rates typically represent a worst case scenario in terms of fatigue crack growth rate. What is missing from equation (1.1) is the interfacial fracture toughness $\mathcal{G}_c$, which also increases sharply with higher mode-mix ratios. There

is a limited amount of fatigue crack growth rate and fracture toughness data for mixed-mode loading, and comparison of experimental results to this theory would certainly be of interest.

In order to compare the results found in the literature, it is convenient to define a "universal" fatigue crack growth law [1] as

$$\frac{da}{dN}^{*} = (\Delta\mathscr{G}^{*})^{2} \tag{6.1}$$

where

$$\frac{da}{dN}^{*} = \frac{\frac{da}{dN}}{\left(\frac{\mathscr{G}_c}{\sigma_y^2}\right)\frac{dW}{dN}^{*}} \tag{6.2}$$

and $\Delta\mathscr{G}^{*} = \frac{\Delta\mathscr{G}}{\mathscr{G}_c}$. These equations stem from equation (3.11) and provide a means of comparing the result of this thesis, through equation (1.1), to physical results in the literature. Another method of comparing data is by predicting the fatigue crack growth rate using equation (3.12) and plotting the predicted curves on the existing data.

There is a recent paper by Nayeb-Hashemi and Yang [39] that reports the critical strain energy release rate $\mathscr{G}_c$, mode-mix ψ, and fatigue crack growth rate $\frac{da}{dN}$ for a brass-solder interface. The geometry used in [39] (shown in Figure 6.2) is not the same as the layered specimen geometry used in this thesis. Instead, a sandwich specimen was used with a thin middle layer of solder between two brass substrates. However, as the middle layer of the sandwich became larger, the local stress field will be more similar to that of the problem herein. Nayeb-Hashemi and Yang only reported three modes corresponding to $\psi = (0^{\circ}, 14^{\circ}, 26^{\circ})$. The results of that study showed that in pure mode I the fracture toughness is low and the fatigue crack growth rate is high. As the mode increases, the fracture toughness becomes similar and, as predicted by equation (1.1), the fatigue crack growth

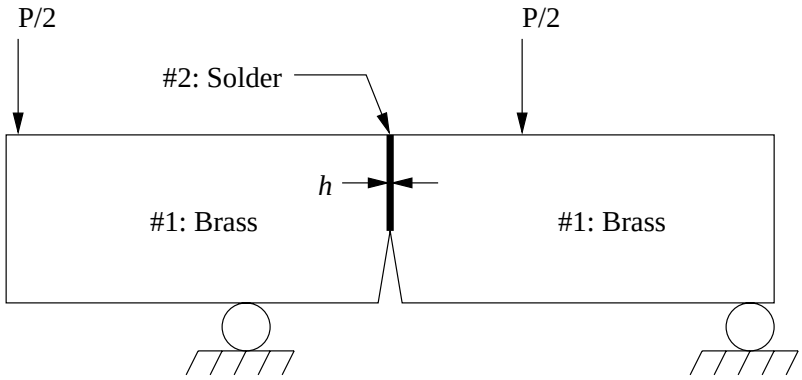


Figure 6.2: Sandwich specimen geometry considered by Nayeb-Hashemi and Yang [39].

rate is higher for the higher mode. This only happened when the sandwiched layer was 1.0mm. Incidentally, the other layer thicknesses reported in [39] were 0.1mm, 0.25mm and 0.5 mm. The correlation with this experimental study is promising, since the plastic dissipation energy approach is able to account for some less intuitive results in the reported data. A conclusion from [39] is that the exponent of the power law relationship $1 < m < 3.5$ which indicates an agreement with the $\Delta\mathscr{G}^2$ relationship predicted in Chapter 3.

Even in the absence of a comparison with sustained mixed-mode crack growth data, the results of this work are useful for comparison with dissipated energy measurements during fatigue crack growth. In particular, Ranganathan [22] has reported dissipated energy measurements under mode I loading which are substantially higher than those predicted by finite element models. Such discrepancies had been attributed in part to a mix of crack extension modes associated with the deformation mechanism at the crack tip. In light of the current work, the presence of a mode II component can significantly increase the dissipated energy at a fatigue crack tip, which tends to support the observation in [22]. Perhaps subsequent experimental studies of dissipated energy under sustained mixed-mode crack growth will allow a more thorough comparison with the results of this work.

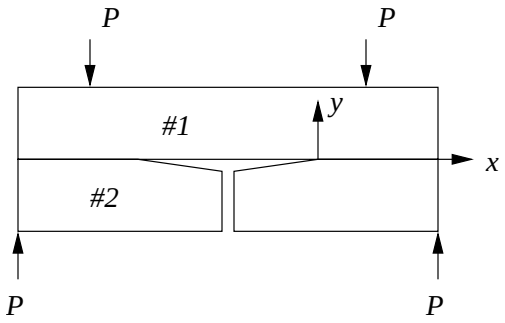


Figure 6.3: Four point bend test specimen.

6.3 Design Guidelines

The results presented in this thesis can be used to provide design guidelines for debond-resistant bimaterial systems. A debond resistant bimaterial will have the least amount of plastic dissipated energy and the highest interfacial fracture toughness. Assume the interfacial fracture toughness plateaus and remains fairly consistent for $\psi > 10^{\circ}$, then the only driving force will be the plastic dissipation energy. Minimizing the plastic dissipation energy can be achieved by stacking the layer in the correct sequence relative to the applied load. An applied load will give a specific value of ψ which can used to locate and predict the amount of plastic dissipation per cycle. For positive values of ψ, if the bottom layer is made more stiff with respect to the top layer (with respect to the loads), then the plastic dissipation will increase thus making the interface more likely to debond.

If a layered system interface is subjected to a four point bending test as shown in Figure 6.3, then layer #1 should be the stiffer layer to have a lower plastic dissipation energy per cycle and thus a slower fatigue crack growth rate. This, of course, assumes the same magnitude of loading would be applied to either case. A complete set of design guidelines would require the interface fracture toughness to also be known. With both these values, a

design could be optimized with the fatigue crack growth rate as a cost function. The educated hypothesis would be the midrange values of mode ($15^\circ \leq \psi \leq 50^\circ$) would have the highest fracture toughness and the least amount of plastic dissipation energy, thus making that loading combination optimal.

6.4 Future Research

The extent of this study is currently constrained to planar numerical models. The goal was to determine trends in the plastic dissipation energy for all possible combinations of ductile metals. Equation (1.1) asserts that the fatigue crack growth rate is directly proportional to the plastic work per cycle. To validate this assertion, some fatigue crack growth studies need to be conducted in mixed-mode. The difficulty, of course is measuring a pure mode II condition and accounting for the energy due to friction between the faces in the wake of the crack. The end result is to be able to use continuum finite element models (i.e. the results of this thesis) to model and predict the fatigue crack growth rate when given only material property data ($E, \mathscr{G}_c, \sigma_y, E_t$) for each layer. This has promising application to accelerated introduction of next-generation materials.

To this end, physical experiments can be conducted to determine the critical strain energy release rate, yield strengths, and the fatigue crack growth rate for the four point bend test specimen. Recall this specimen is a special case of the geometry analyzed herein with $\psi \approx 41^\circ$. The goal of such research would be to generate mixed mode fatigue and fracture data with the hopes of validating equation (1.1) for mixed mode fatigue of ductile bimaterial interfaces.

Chapter 7

Conclusion

This study reports previously unpublished numerical results for the cyclic plastic dissipation energy in ductile materials under mixed-mode loading conditions. These conditions occur in layer manufactured systems, welding, soldering, or any other application where a material is deposited onto a substrate. From the numerical results presented herein, it can be concluded that:

- The plastic dissipation energy follows a power law relation with respect to the strain energy release rate with the power law exponent $m = 2$.
- The plastic work can be reported as a dimensionless parameter to account for yield strength, elastic modulus, and the magnitude of loading ($\Delta\mathscr{G}$).
- The mode-mix ratio is the dominant factor influencing the plastic work per cycle for all cases.
- Changes in Poisson's ratio affect each mode equally with a substantial increase in dissipated energy for plane stress conditions. In plane strain, the effect of Poisson's

ratio is negligible for $\nu \geq 0.3$, which is typical of most ductile metals.

- The effect of specimen geometry in mode I under plane strain is small (less than 0.5%).

- Increasing the hardening (tangent) modulus will decrease the plastic dissipation energy. Also, an increased hardening modulus will mitigate the effect the mode-mix ratio has on $\frac{dW}{dN}^*$.

- Introducing a yield strength mismatch shows a decrease in the normalized plastic dissipation energy when normalizing with respect to the smallest yield strength.

- The effect of a yield strength mismatch is confined to relatively small ratios of mismatch. In other words, the effect of a yield strength mismatch is the same if one layer is stronger than the other by a factor of about 1.67 or more ($\hat{\sigma} \geq 0.25$).

- A hardening modulus mismatch will decrease the plastic dissipation energy per cycle, but not to the same degree as if there were no mismatch. The softer hardening material becomes the dominant producer of dissipated energy.

- Given a plastic mismatch, the response of $\frac{dW}{dN}^*$ to ψ depends on the order of the layers (or the direction of the moments) for all but three cases: pure mode I, pure mode II and a mode in the middle that is a function of the mismatch parameters.

- Introducing an elastic mismatch required redefining the mode-mix ratio with respect to the complex elastic solution for a bimaterial crack.

- The dimensionless plastic dissipation energy will increase or decrease with an elastic modulus mismatch depending by which layer $\frac{dW}{dN}^*$ is normalized.

- Only an elastic mismatch will change the general shape of the $\frac{dW}{dN}^*$ vs ψ curve by introducing extrema away from the pure modes.

- The minimum and maximum values of $\frac{dW}{dN}^*$ are at the pure modes I and II only for no elastic mismatches. The extrema will shift with the introduction of an elastic mismatch.

- The effects of plastic mismatches seem negligible compared to an elastic mismatch when investigating real world specimens.

- For a given mode-mix and equal plastic properties, increasing the elastic modulus of the bottom layer with respect to the top layer (decrease α) will increase the plastic dissipation energy. Likewise, decreasing the bottom layer stiffness with respect to the top layer (increase α) will result in a decrease of the plastic dissipation energy. The bottom layer defined herein is the more “contracting” layer at the interface.

- The maximum plastic work occurs in a bimaterial system when the plastic zones are geometrically symmetrical about the crack plane.

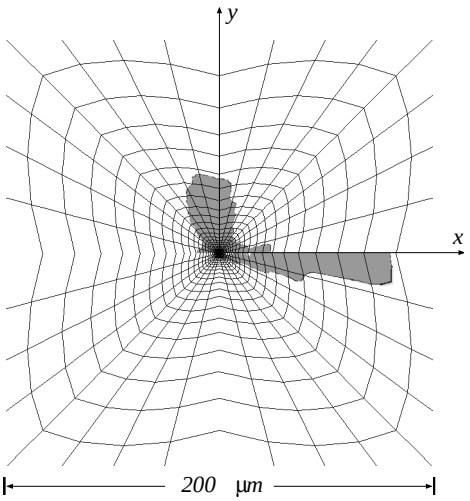
Appendix A

Plane Strain Plastic Zones

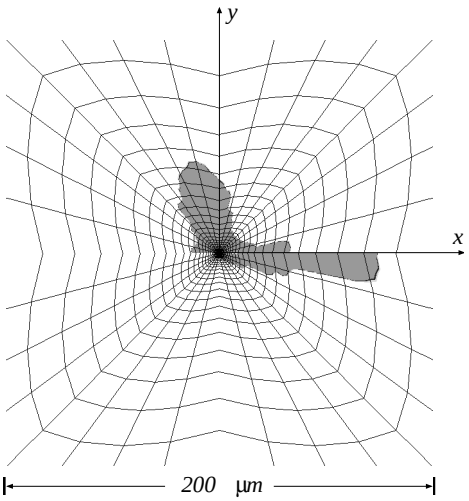
Plastic zones are generated in ABAQUS by plotting contours of the active yield flag. In an elastic-plastic analysis, each element has a bit (or flag) that is set whenever the stress exceeds the yield stress given in the material definition. These flags only indicate the onset of yielding and do not quantify the magnitude. For example, the plastic zone shapes are the same for any value of E_t/E, given everything else is the same. Still, the plastic zones give some insight into the physical aspects of the interface crack problem.

These plastic zones are representatives of the the actual data corresponding to the elastic mismatch results in section 5.3. The forward plastic zones are about four times as large as the reverse zones shown on the following pages. The interesting feature of these plastic zones is how they shift as they go through the modes. The case where there is no mismatch ($\alpha = 0$) shows a geometrically symmetric plastic zone for pure mode I and pure mode II. These modes are also the minima and maxima for the plastic work. As seen by Figure 5.5, the maxima and minima are not at the pure modes. Comparing the phase ψ of the maxima from Figure 5.5 to the plastic zones shows the maxima correspond to the case where the plastic zone is symmetric about the x-axis.

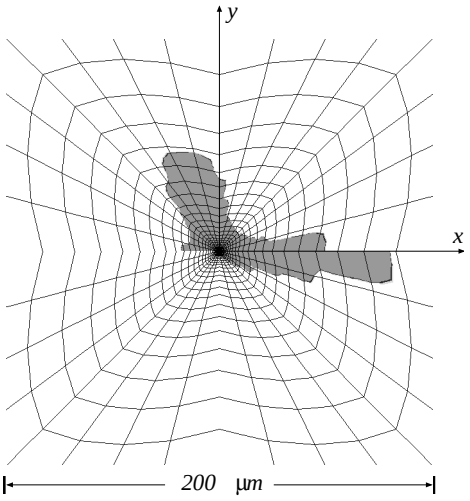
A.1 Reverse Plastic Zones when $\alpha = -0.8$



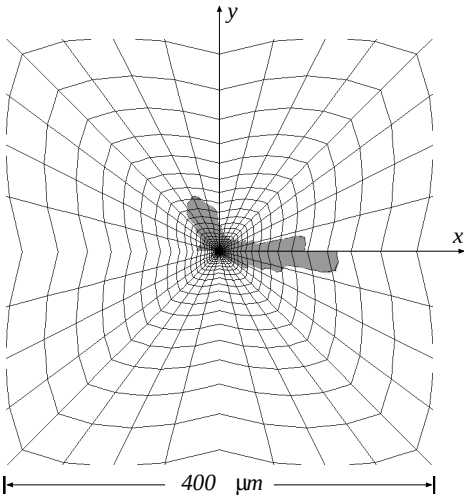
Rev. Zone: α = -0.8
ψ = 0.000
G = 10.667 *J/m/m*



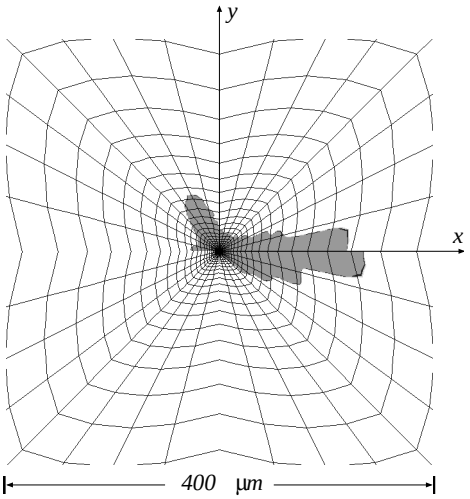
Rev. Zone: α = -0.8
ψ = 6.475
G = 10.667 *J/m/m*



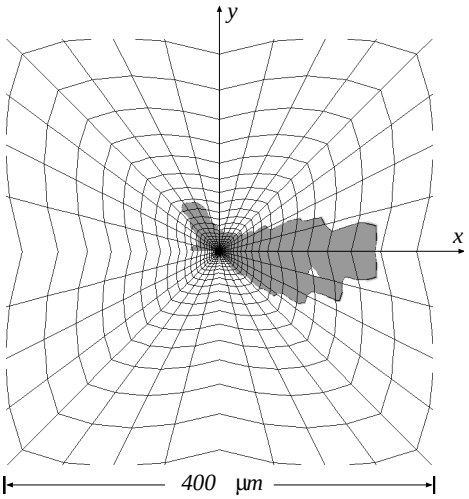
Rev. Zone: α = -0.8
ψ = 14.543
G = 10.667 *J/m/m*



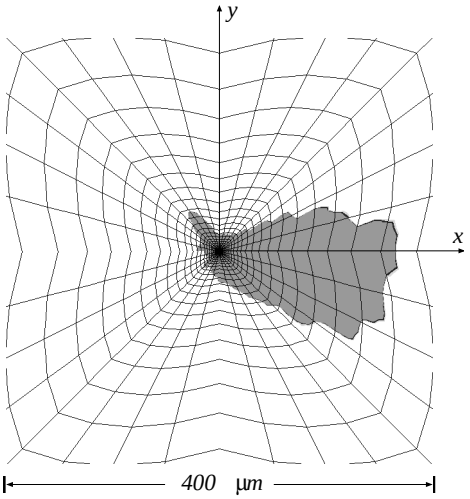
Rev. Zone: α = -0.8
ψ = 24.418
G = 10.667 *J/m/m*



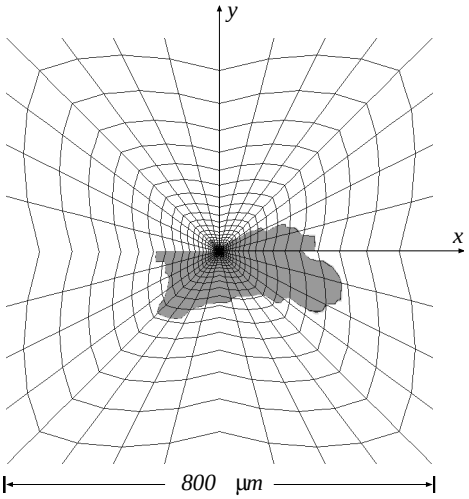
Rev. Zone: $\alpha = -0.8$
$\psi = 35.994$
G = 10.667 *J/m/m*



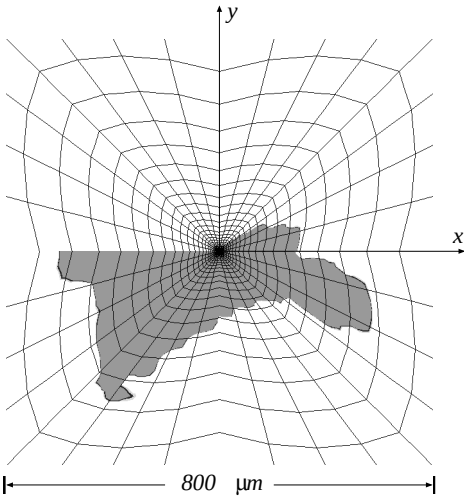
Rev. Zone: $\alpha = -0.8$
$\psi = 48.618$
G = 10.667 *J/m/m*



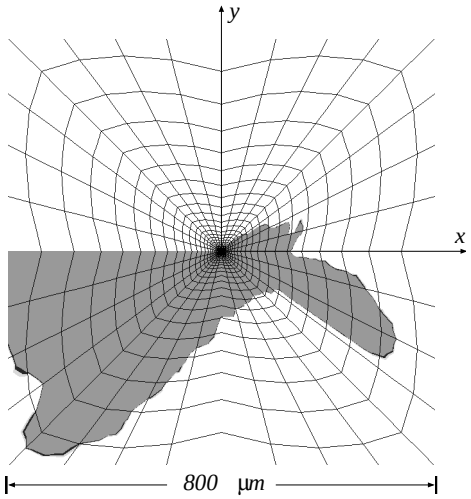
Rev. Zone: $\alpha = -0.8$
$\psi = 61.160$
G = 10.667 *J/m/m*



Rev. Zone: $\alpha = -0.8$
$\psi = 72.533$
G = 10.667 *J/m/m*

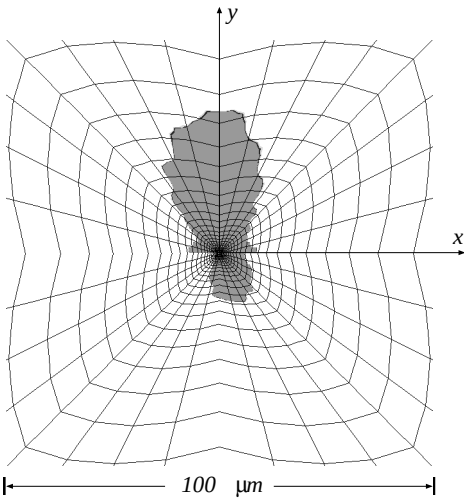


Rev. Zone: $\alpha = -0.8$
$\psi = 82.162$
G = 10.667 *J/m/m*

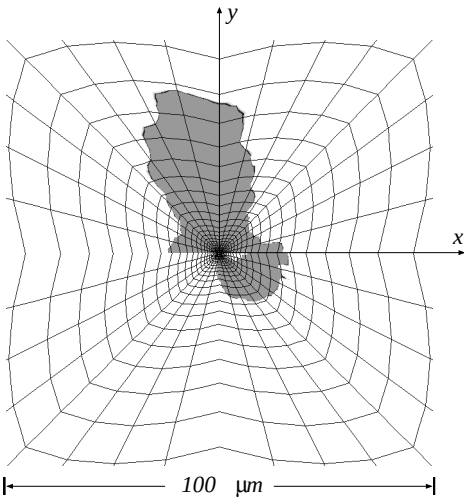


Rev. Zone: $\alpha = -0.8$
$\psi = 90.000$
G = 10.667 *J/m/m*

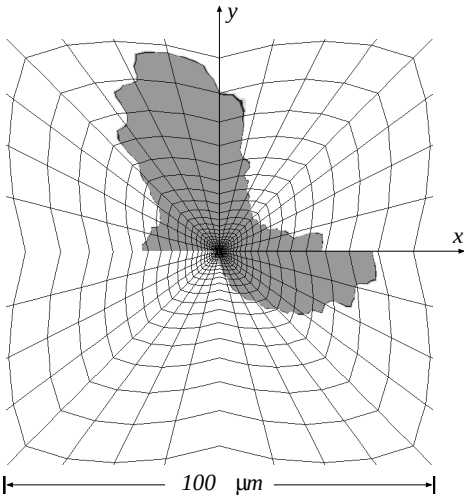
A.2 Reverse Plastic Zones when $\alpha = -0.4$



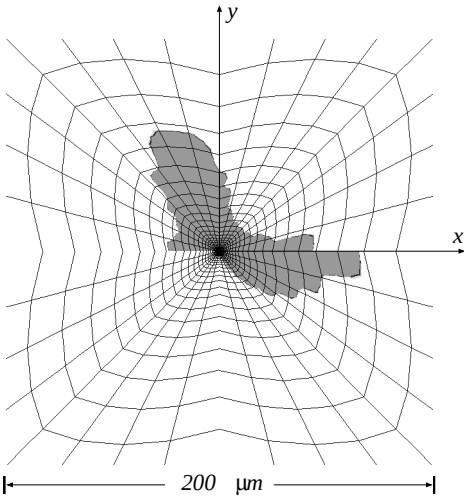
Rev. Zone: α = -0.4
ψ = 0.000
G = 14.143 *J/m/m*



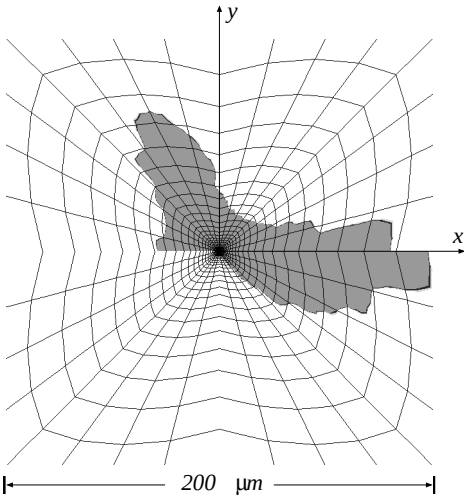
Rev. Zone: α = -0.4
ψ = 6.455
G = 14.143 *J/m/m*



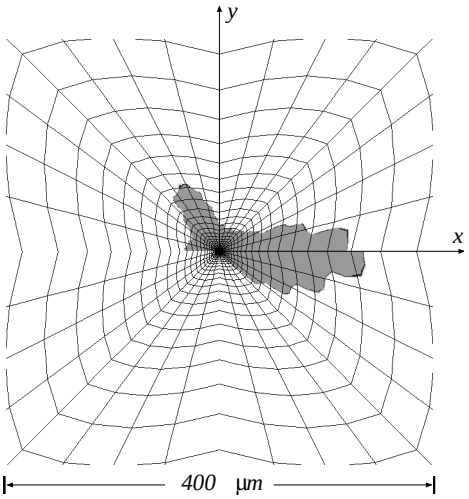
Rev. Zone: α = -0.4
ψ = 14.498
G = 14.143 *J/m/m*



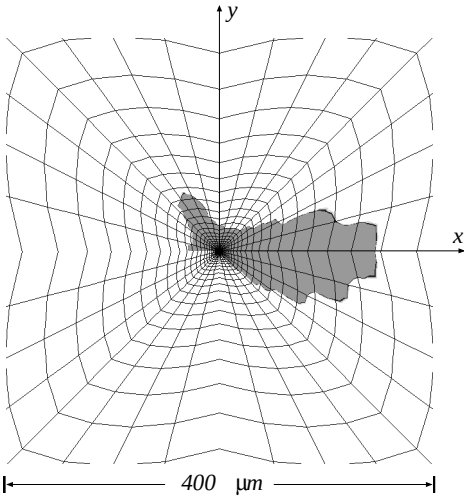
Rev. Zone: α = -0.4
ψ = 24.348
G = 14.143 *J/m/m*



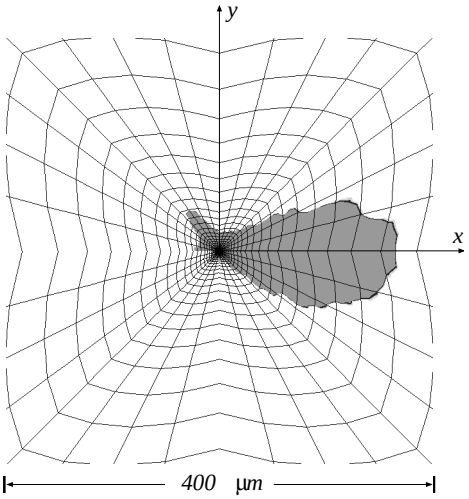
Rev. Zone: $\alpha = -0.4$
$\psi = 35.906$
G = 14.143 *J/m/m*



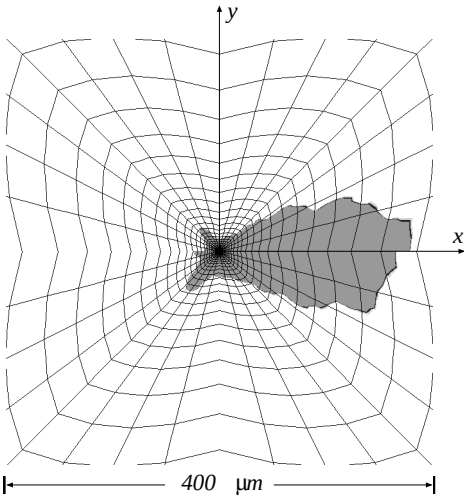
Rev. Zone: $\alpha = -0.4$
$\psi = 48.526$
G = 14.143 *J/m/m*



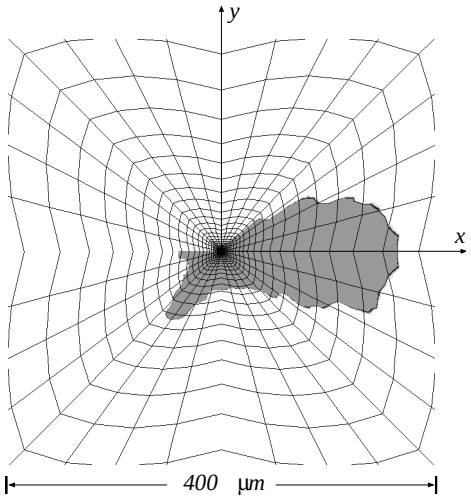
Rev. Zone: $\alpha = -0.4$
$\psi = 61.082$
G = 14.143 *J/m/m*



Rev. Zone: $\alpha = -0.4$
$\psi = 72.480$
G = 14.143 *J/m/m*

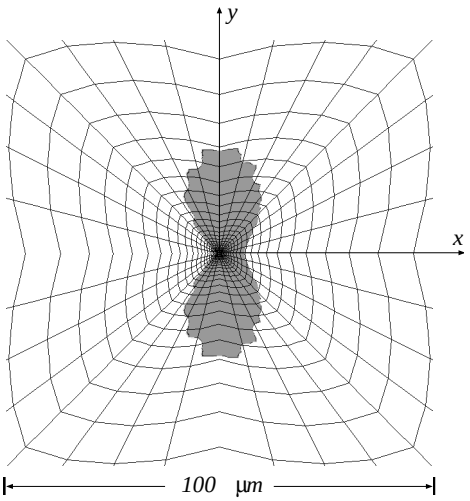


Rev. Zone: α = -0.4
ψ = 82.136
G = 14.143 *J/m/m*

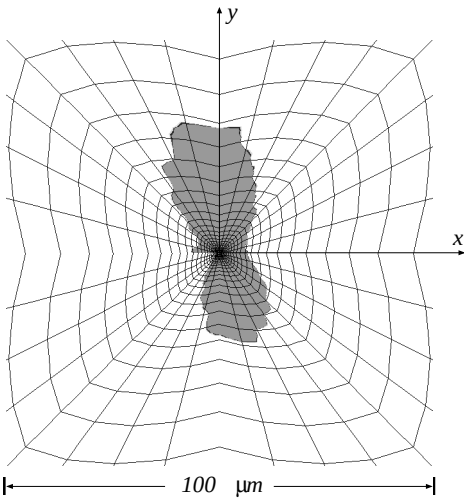


Rev. Zone: α = -0.4
ψ = 90.000
G = 14.143 *J/m/m*

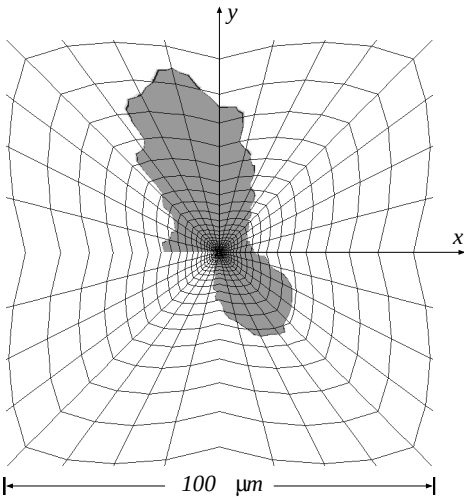
A.3 Reverse Plastic Zones when $\alpha = 0.0$



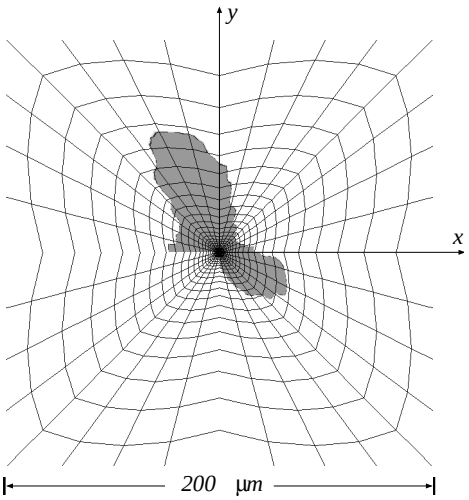
Rev. Zone: α = 0.0
ψ = 0.000
G = 20.000 *J/m/m*



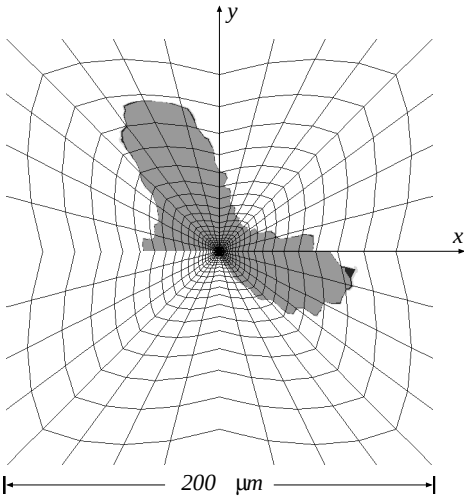
Rev. Zone: α = 0.0
ψ = 6.178
G = 20.000 *J/m/m*



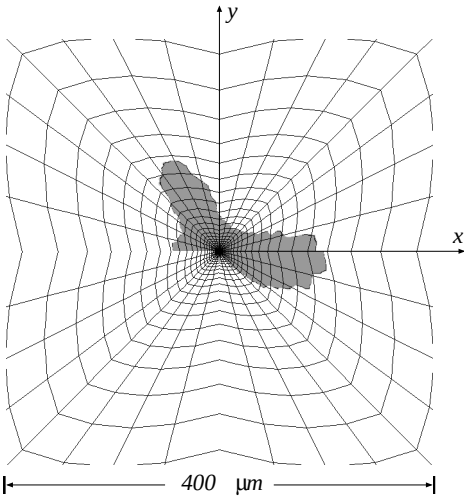
Rev. Zone: α = 0.0
ψ = 13.898
G = 20.000 *J/m/m*



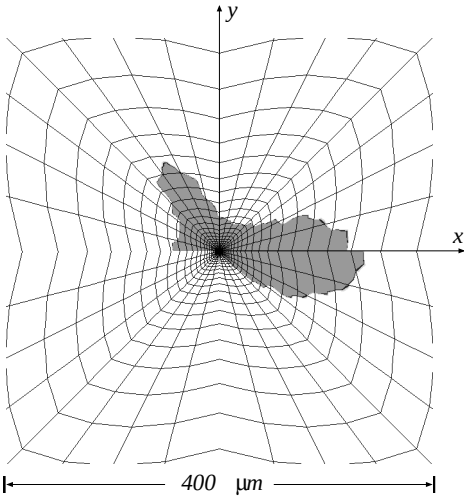
Rev. Zone: α = 0.0
ψ = 23.413
G = 20.000 *J/m/m*



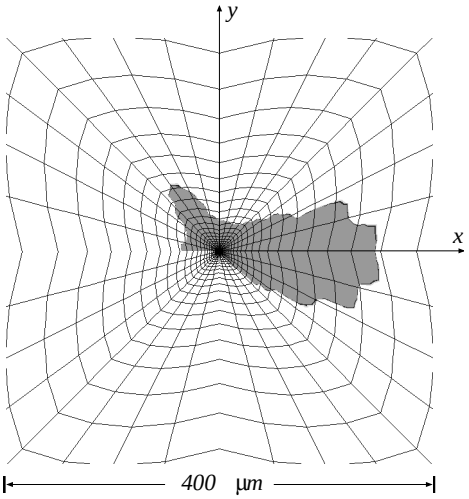
Rev. Zone: $\alpha = 0.0$
$\psi = 34.715$
G = 20.000 *J/m/m*



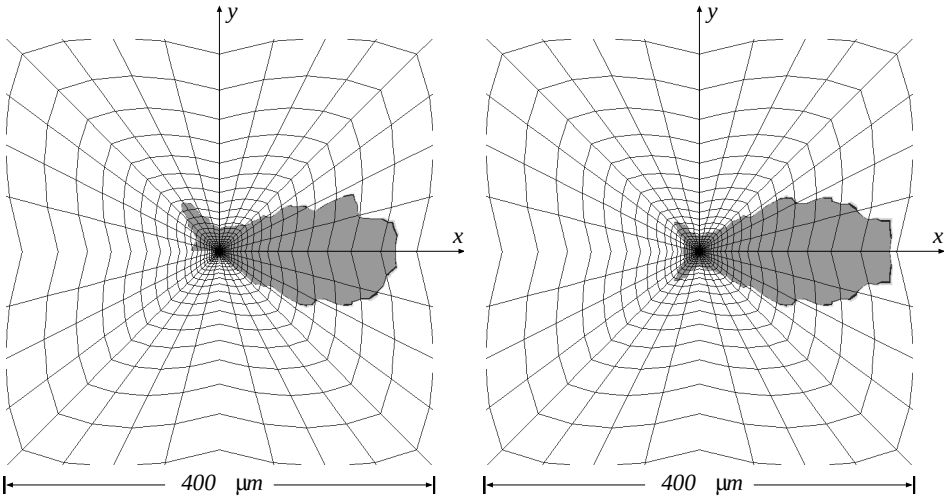
Rev. Zone: $\alpha = 0.0$
$\psi = 47.269$
G = 20.000 *J/m/m*



Rev. Zone: $\alpha = 0.0$
$\psi = 60.000$
G = 20.000 *J/m/m*



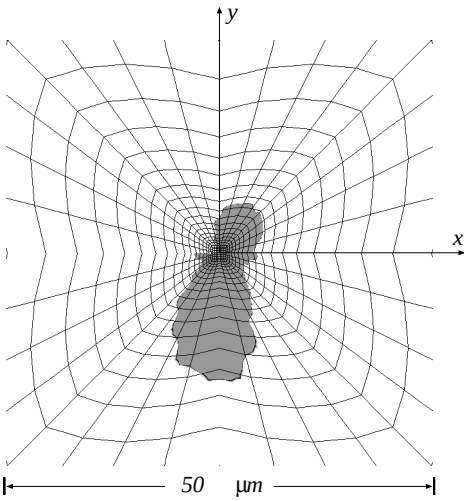
Rev. Zone: $\alpha = 0.0$
$\psi = 71.742$
G = 20.000 *J/m/m*



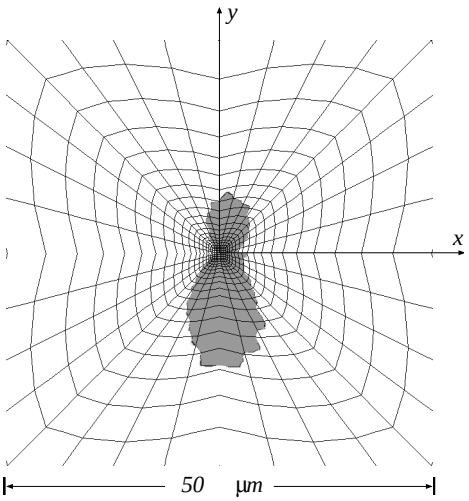
Rev. Zone: $\alpha = 0.0$
$\psi = 81.787$
G = 20.000 *J/m/m*

Rev. Zone: $\alpha = 0.0$
$\psi = 90.000$
G = 20.000 *J/m/m*

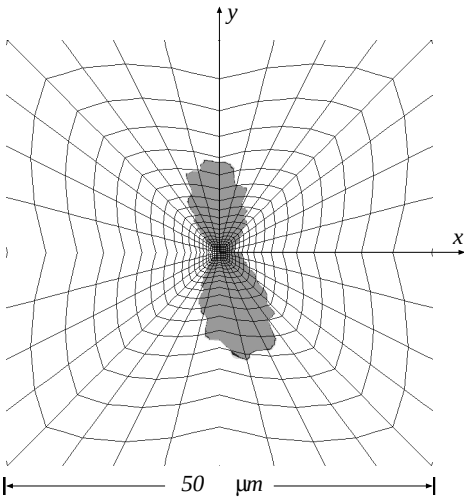
A.4 Reverse Plastic Zones when $\alpha = 0.4$



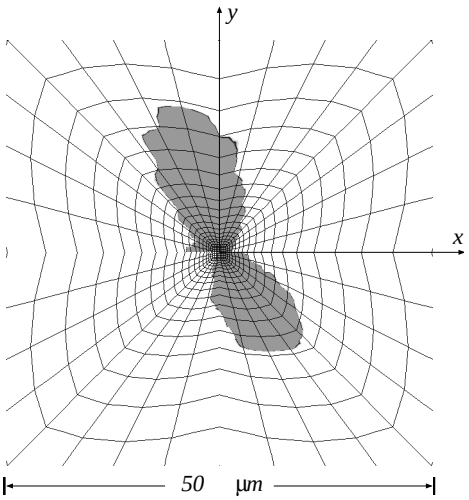
Rev. Zone: α = 0.4
ψ = 0.000
G = 14.143 *J/m/m*



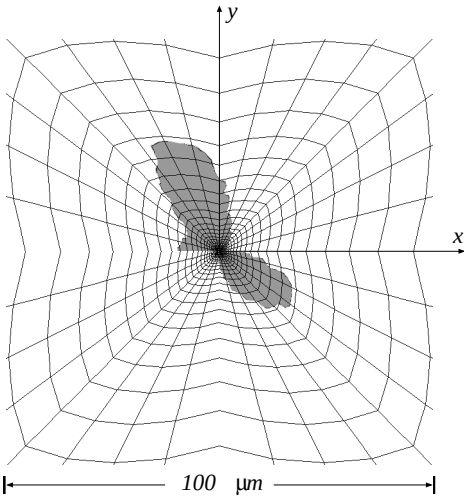
Rev. Zone: α = 0.4
ψ = 5.802
G = 14.143 *J/m/m*



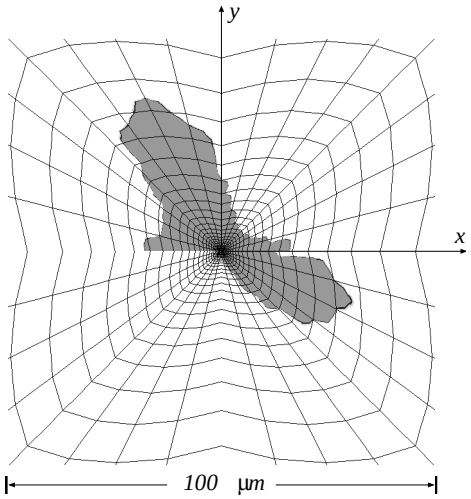
Rev. Zone: α = 0.4
ψ = 13.075
G = 14.143 *J/m/m*



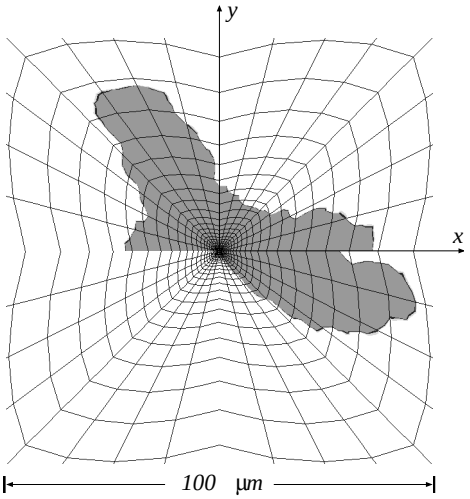
Rev. Zone: α = 0.4
ψ = 22.118
G = 14.143 *J/m/m*



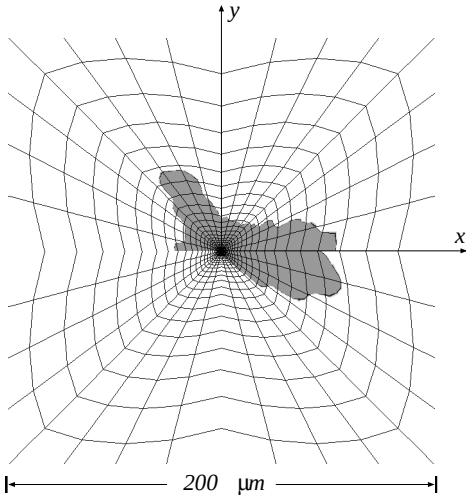
Rev. Zone: $\alpha = 0.4$
$\psi = 33.035$
G = 14.143 *J/m/m*



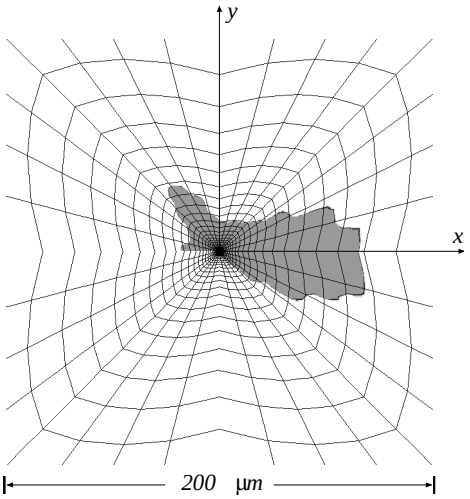
Rev. Zone: $\alpha = 0.4$
$\psi = 45.457$
G = 14.143 *J/m/m*



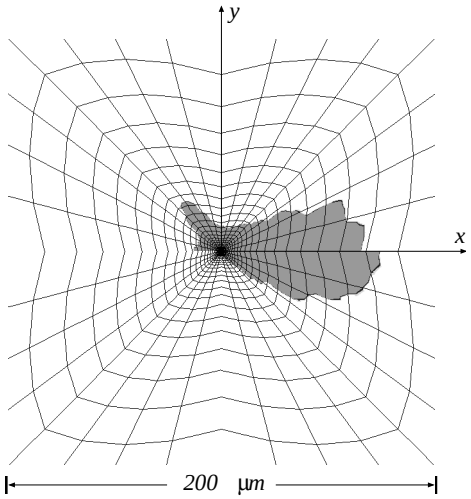
Rev. Zone: $\alpha = 0.4$
$\psi = 58.404$
G = 14.143 *J/m/m*



Rev. Zone: $\alpha = 0.4$
$\psi = 70.634$
G = 14.143 *J/m/m*

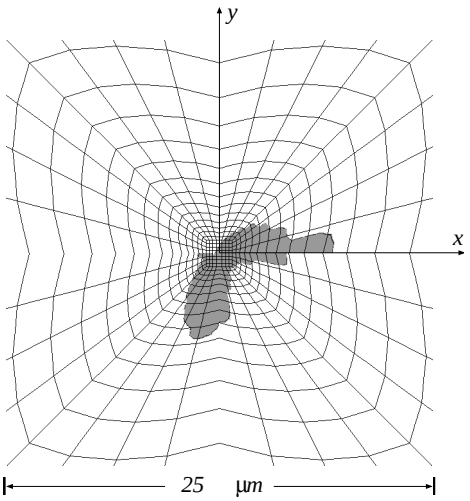


Rev. Zone: $\alpha = 0.4$
$\psi = 81.258$
G = 14.143 *J/m/m*

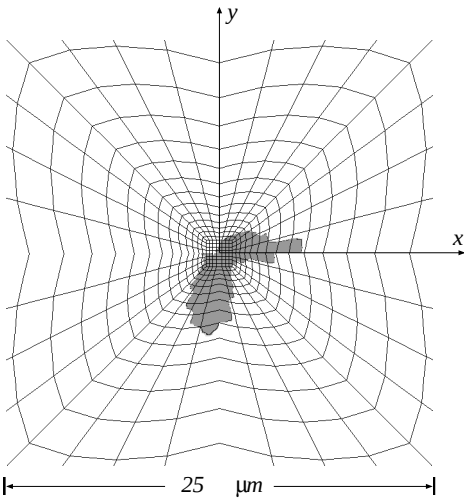


Rev. Zone: $\alpha = 0.4$
$\psi = 90.000$
G = 14.143 *J/m/m*

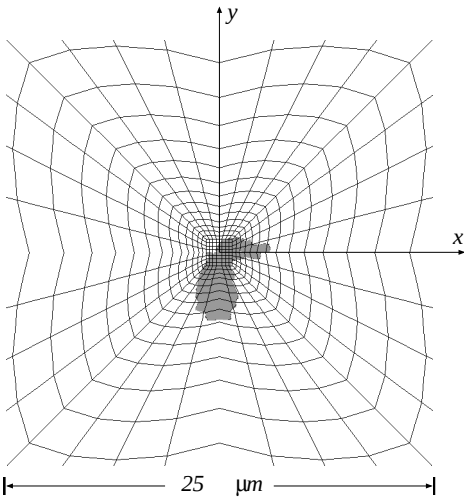
A.5 Reverse Plastic Zones when $\alpha = 0.8$



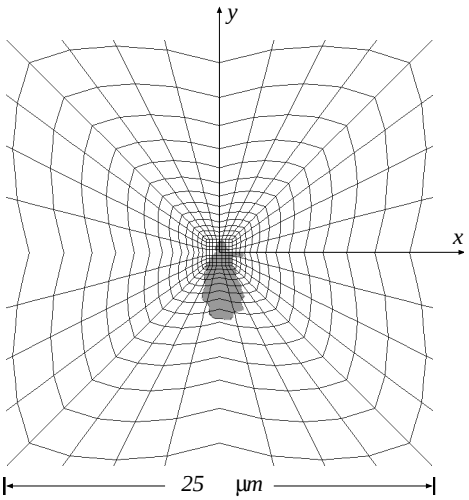
Rev. Zone: α = 0.8
ψ = 0.000
G = 10.667 *J/m/m*



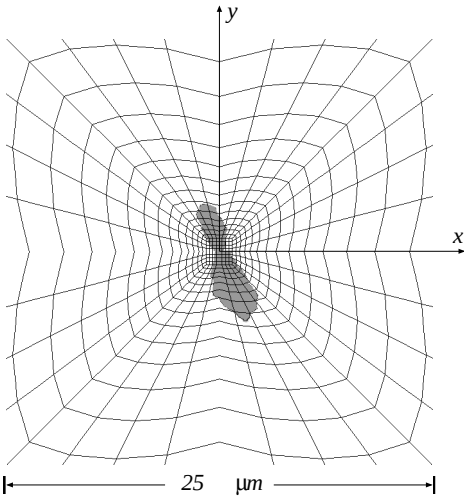
Rev. Zone: α = 0.8
ψ = 5.274
G = 10.667 *J/m/m*



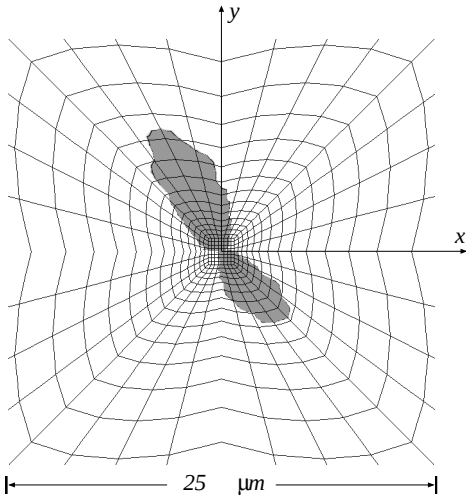
Rev. Zone: α = 0.8
ψ = 11.914
G = 10.667 *J/m/m*



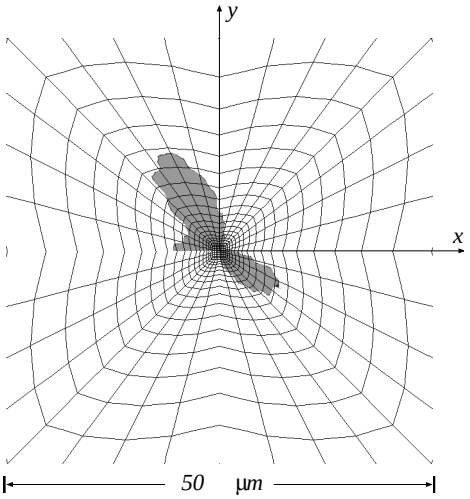
Rev. Zone: α = 0.8
ψ = 20.265
G = 10.667 *J/m/m*



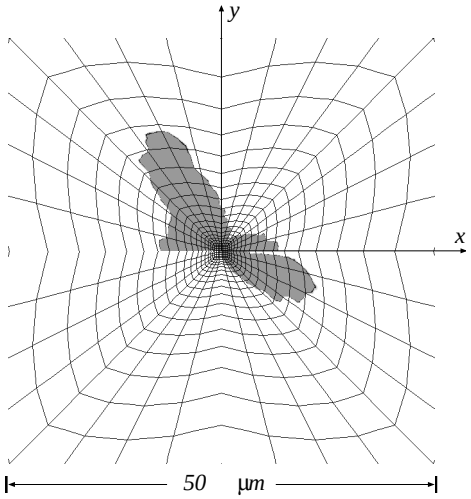
Rev. Zone: $\alpha = 0.8$
$\psi = 30.573$
G = 10.667 *J/m/m*



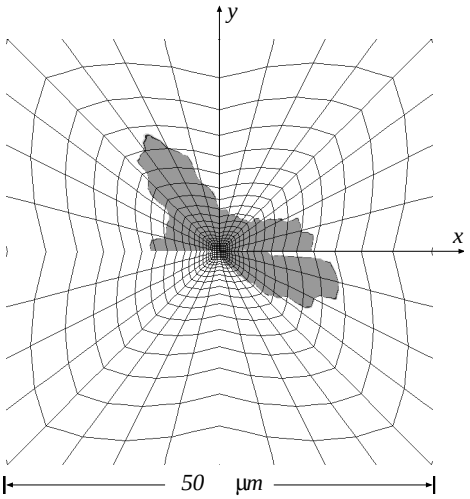
Rev. Zone: $\alpha = 0.8$
$\psi = 42.708$
G = 10.667 *J/m/m*



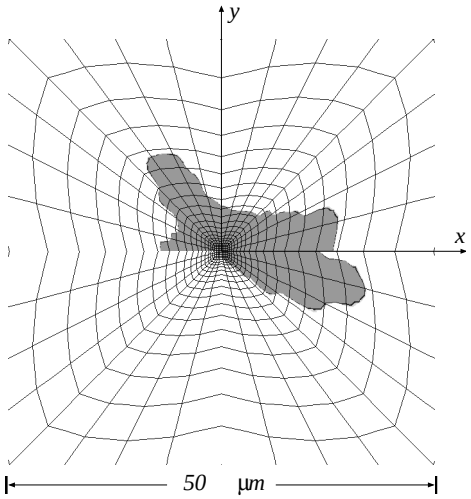
Rev. Zone: $\alpha = 0.8$
$\psi = 55.898$
G = 10.667 *J/m/m*



Rev. Zone: $\alpha = 0.8$
$\psi = 68.847$
G = 10.667 *J/m/m*



Rev. Zone: $\alpha = 0.8$
$\psi = 80.392$
G = 10.667 *J/m/m*



Rev. Zone: $\alpha = 0.8$
$\psi = 90.000$
G = 10.667 *J/m/m*

Appendix B

Raw Data from an Elastic Mismatch

These data were used in generating the plots and conclusions in Chapter 5.

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	7.40E+05	-0.8	-0.2	0.333333	0.333333	5	5	300	300	7.31E-21	6.58E-20

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha1		-0.8	-0.2	0	1.00E-25	300.3776	1031.506	3.434032	0.106667	0.106666	0.106666	0.000032	4.40E-05	0.003416608	0.030749475
alpha2		-0.8	-0.2	0	1.00E-25	335.7687	867.3954	2.583312	0.106667	0.106666	0.106666	0.000421	-6.47524	0.00449914	0.040492257
alpha3		-0.8	-0.2	0	1.00E-25	373.8248	647.6857	1.732591	0.106667	0.106667	0.106667	0.0006	-14.5433	0.006416566	0.057749094
alpha4		-0.8	-0.2	0	1.00E-25	410.2663	361.8019	0.881871	0.106667	0.106667	0.106667	0.000899	-24.4176	0.009608589	0.086477299
alpha5		-0.8	-0.2	0	1.00E-25	437.4533	13.62688	0.03115	0.106667	0.106667	0.106667	0.001325	-35.9941	0.014162378	0.1274614
alpha6		-0.8	-0.2	0	1.00E-25	446.7962	-366.181	-0.81957	0.106667	0.106667	0.106667	0.001807	-48.6175	0.019311855	0.173806692
alpha7		-0.8	-0.2	0	1.00E-25	434.6864	-726.053	-1.67029	0.106667	0.106667	0.106667	0.002173	-61.1597	0.023218427	0.208965841
alpha8		-0.8	-0.2	0	1.00E-25	405.734	-1022.86	-2.52101	0.106667	0.106667	0.106667	0.002291	-72.5328	0.024487091	0.220383819
alpha9		-0.8	-0.2	0	1.00E-25	368.7026	-1243.17	-3.37173	0.106667	0.106667	0.106667	0.002196	-82.1616	0.02347422	0.21126798
alpha10		-0.8	-0.2	0	1.00E-25	330.8275	-1396.9	-4.22245	0.106667	0.106666	0.106666	0.002013	-90	0.021513305	0.193619742

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	3.29E+05	-0.6	-0.15	0.333333	0.333333	5	5	300	300	7.31E-21	2.92E-20

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha11	-0.6		-0.15	0	1.00E-25	324.3544	670.3257	2.066646	0.122188	0.122187	0.122187	0.000273	-4.57E-05	0.005002683	0.020010732
alpha12	-0.6		-0.15	0	1.00E-25	362.5398	568.5541	1.568253	0.122188	0.122188	0.122188	0.000365	-6.51762	0.006693301	0.026773203
alpha13	-0.6		-0.15	0	1.00E-25	403.4951	431.6834	1.06986	0.122188	0.122187	0.122187	0.000525	-14.6352	0.009614007	0.038456028
alpha14	-0.6		-0.15	0	1.00E-25	442.5125	252.8815	0.571468	0.122188	0.122188	0.122188	0.000818	-24.5599	0.014993446	0.059973783
alpha15	-0.6		-0.15	0	1.00E-25	471.2921	34.43956	0.073075	0.122188	0.122187	0.122187	0.001282	-36.1736	0.023491707	0.09396683
alpha16	-0.6		-0.15	0	1.00E-25	480.6677	-204.437	-0.42532	0.122188	0.122187	0.122187	0.001863	-48.8045	0.034145456	0.136581823
alpha17	-0.6		-0.15	0	1.00E-25	467.0122	-431.384	-0.92371	0.122188	0.122188	0.122188	0.002365	-61.3186	0.043335412	0.173341648
alpha18	-0.6		-0.15	0	1.00E-25	435.4995	-619.325	-1.4221	0.122188	0.122187	0.122187	0.002601	-72.6405	0.047671423	0.190685691
alpha19	-0.6		-0.15	0	1.00E-25	395.5665	-759.684	-1.9205	0.122188	0.122187	0.122187	0.002568	-82.2122	0.047061906	0.188247625
alpha20	-0.6		-0.15	0	1.00E-25	354.8887	-858.436	-2.41889	0.122188	0.122188	0.122188	0.002385	-90	0.043700238	0.174800952

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	1.92E+05	-0.4	-0.1	0.333333	0.333333	5	5	300	300	7.31E-21	1.71E-20

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha21	-0.4		-0.1	0	1.00E-25	349.4248	536.115	1.534279	0.141429	0.141429	0.141429	0.000231	-8.34E-05	0.005424414	0.012656965
alpha22	-0.4		-0.1	0	1.00E-25	390.6106	458.5731	1.17399	0.141429	0.141429	0.141429	0.000312	-6.45454	0.007324062	0.017089478
alpha23	-0.4		-0.1	0	1.00E-25	434.9522	353.9216	0.813702	0.141429	0.141429	0.141429	0.000458	-14.4984	0.010732229	0.025041868
alpha24	-0.4		-0.1	0	1.00E-25	477.5169	216.5129	0.453414	0.141429	0.141429	0.141429	0.000728	-24.3482	0.017072202	0.039835138
alpha25	-0.4		-0.1	0	1.00E-25	509.4445	47.44257	0.093126	0.141429	0.141429	0.141429	0.001205	-35.9064	0.028244618	0.065904108
alpha26	-0.4		-0.1	0	1.00E-25	520.6877	-139.108	-0.26716	0.141429	0.141429	0.141429	0.001871	-48.5259	0.043877841	0.102381628
alpha27	-0.4		-0.1	0	1.00E-25	506.9073	-318.059	-0.62745	0.141429	0.141429	0.141429	0.002529	-61.0816	0.059313448	0.138398045
alpha28	-0.4		-0.1	0	1.00E-25	473.3615	-467.557	-0.98774	0.141429	0.141428	0.141428	0.002942	-72.4798	0.068976849	0.160945981
alpha29	-0.4		-0.1	0	1.00E-25	430.2567	-579.997	-1.34803	0.141429	0.141429	0.141429	0.003033	-82.1364	0.071119244	0.165944903
alpha30	-0.4		-0.1	0	1.00E-25	386.0816	-659.549	-1.70831	0.141429	0.141429	0.141429	0.00291	-89.9999	0.068242029	0.159231402

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	1.23E+05	-0.2	-0.05	0.333333	0.333333	5	5	300	300	7.31E-21	1.10E-20

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha31	-0.2	-0.05	0	1.00E-25	378.1782	462.4051	1.222718	0.16625	0.16625	0.16625	0.000205	9.00E-05	0.005413355	0.008120032
alpha32	-0.2	-0.05	0	1.00E-25	422.8532	398.9122	0.943382	0.16625	0.16625	0.16625	0.000265	-6.33381	0.006991116	0.010486675
alpha33	-0.2	-0.05	0	1.00E-25	471.2964	312.9629	0.664047	0.16625	0.16625	0.16625	0.000392	-14.2361	0.010352607	0.015528911
alpha34	-0.2	-0.05	0	1.00E-25	518.4609	199.4579	0.384712	0.16625	0.16625	0.16625	0.000642	-23.9408	0.01694543	0.025418145
alpha35	-0.2	-0.05	0	1.00E-25	554.9445	58.47794	0.105376	0.16625	0.16625	0.16625	0.001109	-35.3895	0.029262972	0.043894459
alpha36	-0.2	-0.05	0	1.00E-25	569.5405	-99.0768	-0.17396	0.16625	0.16625	0.16625	0.001837	-47.9838	0.048494707	0.072742061
alpha37	-0.2	-0.05	0	1.00E-25	556.6465	-252.325	-0.45329	0.16625	0.16625	0.16625	0.002665	-60.6173	0.070343086	0.105514629
alpha38	-0.2	-0.05	0	1.00E-25	521.2462	-381.881	-0.73263	0.16625	0.16625	0.16625	0.003289	-72.1641	0.086817357	0.130226036
alpha39	-0.2	-0.05	0	1.00E-25	474.4407	-480.118	-1.01197	0.16625	0.16625	0.16625	0.00356	-81.9872	0.093985702	0.140978553
alpha40	-0.2	-0.05	0	1.00E-25	425.8834	-549.943	-1.2913	0.16625	0.16625	0.16625	0.003541	-90	0.093483641	0.140225462

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	8.22E+04	0	0	0.333333	0.333333	5	5	300	300	7.31E-21	7.31E-21

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha41	0	0	0	1.00E-25	413.918	413.918	1	0.2	0.2	0.2	0.000218	-1.72E-07	0.005966672	0.005966672
alpha42	0	0	0	1.00E-25	462.9531	360.0746	0.777778	0.2	0.2	0.2	0.000242	-6.1784	0.006609995	0.006609995
alpha43	0	0	0	1.00E-25	516.6009	287.0005	0.555556	0.2	0.2	0.2	0.000336	-13.8979	0.009196986	0.009196986
alpha44	0	0	0	1.00E-25	569.7558	189.9186	0.333333	0.2	0.2	0.2	0.000559	-23.4132	0.015304286	0.015304286
alpha45	0	0	0	1.00E-25	612.4293	68.0477	0.111111	0.2	0.2	0.2	0.001012	-34.715	0.027688602	0.027688602
alpha46	0	0	0	1.00E-25	631.9452	-70.2161	-0.11111	0.2	0.199999	0.199999	0.001784	-47.2694	0.048814604	0.048814604
alpha47	0	0	0	1.00E-25	620.877	-206.959	-0.33333	0.2	0.2	0.2	0.002779	-60	0.076022804	0.076022804
alpha48	0	0	0	1.00E-25	583.5701	-324.206	-0.55556	0.2	0.2	0.2	0.003661	-71.7416	0.100173209	0.100173209
alpha49	0	0	0	1.00E-25	532.1803	-413.918	-0.77778	0.2	0.2	0.2	0.004171	-81.7868	0.11410875	0.11410875
alpha50	0	0	0	1.00E-25	477.9514	-477.951	-1	0.2	0.2	0.2	0.004309	-90	0.117887805	0.117887805

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	5.48E+04	0.2	0.05	0.333333	0.333333	5	5	300	300	7.31E-21	4.87E-21

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha51	0.2	0.05	0	1.00E-25	377.1386	309.1012	0.819596	0.16625	0.16625	0.16625	0.000139	0.052708	0.008260482	0.005506988
alpha52	0.2	0.05	0	1.00E-25	422.0324	271.1019	0.642372	0.16625	0.16625	0.16625	0.000124	-5.94547	0.007373939	0.004915959
alpha53	0.2	0.05	0	1.00E-25	471.6697	219.3965	0.465149	0.16625	0.16625	0.16625	0.000139	-13.4549	0.008264064	0.005509376
alpha54	0.2	0.05	0	1.00E-25	521.8449	150.2523	0.287925	0.16625	0.16625	0.16625	0.000213	-22.7546	0.012664035	0.00844269
alpha55	0.2	0.05	0	1.00E-25	563.7939	62.41288	0.110702	0.16625	0.16625	0.16625	0.000398	-33.8918	0.023612331	0.015741554
alpha56	0.2	0.05	0	1.00E-25	585.548	-38.9518	-0.06652	0.16625	0.16625	0.16625	0.000755	-46.4094	0.044816315	0.029877543
alpha57	0.2	0.05	0	1.00E-25	578.9082	-141.106	-0.24375	0.16625	0.16625	0.16625	0.001274	-59.2682	0.07563986	0.050426573
alpha58	0.2	0.05	0	1.00E-25	546.5326	-230.073	-0.42097	0.16625	0.16625	0.16625	0.001806	-71.2569	0.107260435	0.071506957
alpha59	0.2	0.05	0	1.00E-25	499.4661	-298.777	-0.59819	0.16625	0.16625	0.16625	0.002182	-81.581	0.129620303	0.086413535
alpha60	0.2	0.05	0	1.00E-25	448.7442	-347.963	-0.77542	0.16625	0.16625	0.16625	0.002355	89.95715	0.139861767	0.093241178

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	3.52E+04	0.4	0.1	0.333333	0.333333	5	5	300	300	7.31E-21	3.13E-21

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha61	0.4	0.1	0	1.00E-25	350.9699	228.7521	0.651771	0.141429	0.141428	0.141428	0.000105	-0.0001	0.013416519	0.005749937
alpha62	0.4	0.1	0	1.00E-25	392.8192	202.0311	0.514311	0.141429	0.141429	0.141429	8.35E-05	-5.80187	0.010655064	0.004566456
alpha63	0.4	0.1	0	1.00E-25	439.55	165.6446	0.37685	0.141429	0.141428	0.141428	7.19E-05	-13.075	0.009178315	0.003933564
alpha64	0.4	0.1	0	1.00E-25	487.7152	116.7541	0.23939	0.141429	0.141429	0.141429	8.34E-05	-22.1183	0.010644676	0.004562004
alpha65	0.4	0.1	0	1.00E-25	529.6183	53.98374	0.10193	0.141429	0.141429	0.141429	0.000144	-33.0355	0.018334382	0.007857592
alpha66	0.4	0.1	0	1.00E-25	553.9263	-19.6815	-0.03553	0.141429	0.141429	0.141429	0.000287	-45.4568	0.036669344	0.015715433
alpha67	0.4	0.1	0	1.00E-25	551.6572	-95.4319	-0.17299	0.141429	0.141429	0.141429	0.000533	-58.4038	0.068099856	0.029185652
alpha68	0.4	0.1	0	1.00E-25	523.7314	-162.593	-0.31045	0.141429	0.141428	0.141428	0.000826	-70.6337	0.105477253	0.045204537
alpha69	0.4	0.1	0	1.00E-25	480.1053	-215.045	-0.44791	0.141429	0.141428	0.141428	0.001074	-81.2577	0.137097571	0.058756102
alpha70	0.4	0.1	0	1.00E-25	431.7759	-252.75	-0.58537	0.141429	0.141429	0.141429	0.001225	89.99993	0.156330838	0.066998931

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	2.06E+04	0.6	0.15	0.333333	0.333333	5	5	300	300	7.31E-21	1.83E-21

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha71	0.6	0.15	0	1.00E-25	335.1631	162.1771	0.483875	0.122188	0.122187	0.122187	7.94E-05	4.53E-05	0.023292407	0.005823102
alpha72	0.6	0.15	0	1.00E-25	375.2756	144.1721	0.384177	0.122188	0.122187	0.122187	6.18E-05	-5.57497	0.018133835	0.004533459
alpha73	0.6	0.15	0	1.00E-25	420.585	119.6471	0.284478	0.122188	0.122187	0.122187	4.70E-05	-12.5773	0.013779964	0.003444991
alpha74	0.6	0.15	0	1.00E-25	468.318	86.53539	0.184779	0.122188	0.122188	0.122188	3.93E-05	-21.3277	0.01150866	0.002877165
alpha75	0.6	0.15	0	1.00E-25	511.6658	43.53275	0.08508	0.122188	0.122187	0.122187	4.80E-05	-31.9933	0.014083834	0.003520959
alpha76	0.6	0.15	0	1.00E-25	539.6592	-7.88887	-0.01462	0.122188	0.122188	0.122188	8.96E-05	-44.3071	0.026282533	0.006570633
alpha77	0.6	0.15	0	1.00E-25	542.2044	-61.9831	-0.11432	0.122188	0.122188	0.122188	0.000182	-57.3685	0.05335575	0.013338937
alpha78	0.6	0.15	0	1.00E-25	518.2529	-110.914	-0.21402	0.122188	0.122188	0.122188	0.000317	-69.903	0.092949529	0.023237382
alpha79	0.6	0.15	0	1.00E-25	476.8013	-149.579	-0.31371	0.122188	0.122188	0.122188	0.000453	-80.9056	0.132702476	0.033175619
alpha80	0.6	0.15	0	1.00E-25	429.2182	-177.444	-0.41341	0.122188	0.122188	0.122188	0.000554	89.99999	0.16230744	0.04057686

Material Properties:

Etop	Ebot	alpha	beta	v1	v2	h1	h1	yield1	yield2	Ettop	Etbot
8.22E+04	9.14E+03	0.8	0.2	0.333333	0.333333	5	5	300	300	7.31E-21	8.12E-22

Job	alpha	beta	sighat	Et/E	M1	M2	M2 Ratio	G	J1	J2	dW/dN	ModeMix	dW/dN* (bottom)	dW/dN* (top)
alpha81	0.8	0.2	0	1.00E-25	343.835	100.126	0.291204	0.106667	0.106667	0.106667	5.08E-05	0.000114	0.043941951	0.004882439
alpha82	0.8	0.2	0	1.00E-25	385.1773	89.56659	0.232533	0.106667	0.106667	0.106667	4.00E-05	-5.27361	0.034649028	0.003849892
alpha83	0.8	0.2	0	1.00E-25	432.5523	75.20494	0.173863	0.106667	0.106666	0.106666	2.99E-05	-11.9137	0.02588707	0.002876341
alpha84	0.8	0.2	0	1.00E-25	483.8301	55.73387	0.115193	0.106667	0.106666	0.106666	2.14E-05	-20.2651	0.018567113	0.002063013
alpha85	0.8	0.2	0	1.00E-25	532.8722	30.11947	0.056523	0.106667	0.106666	0.106666	1.66E-05	-30.5725	0.014348897	0.001594322
alpha86	0.8	0.2	0	1.00E-25	568.4819	-1.2207	-0.00215	0.106667	0.106666	0.106666	2.04E-05	-42.7085	0.017638282	0.001959809
alpha87	0.8	0.2	0	1.00E-25	578.345	-35.1735	-0.06082	0.106667	0.106666	0.106666	3.82E-05	-55.8978	0.033030967	0.003670107
alpha88	0.8	0.2	0	1.00E-25	558.3381	-66.7145	-0.11949	0.106667	0.106666	0.106666	7.54E-05	-68.8475	0.065238019	0.007248669
alpha89	0.8	0.2	0	1.00E-25	516.4926	-92.0172	-0.17816	0.106667	0.106667	0.106667	0.000125	-80.392	0.108208376	0.012023153
alpha90	0.8	0.2	0	1.00E-25	465.6354	-110.275	-0.23683	0.106667	0.106667	0.106667	0.000171	-89.9997	0.147992295	0.016443588

Appendix C

Scripts

The finite element software, ABAQUS version 6, was written to be used with the python language. The graphical user interface called ABAQUS/CAE (Complete ABAQUS Environment) uses the python language to record the user inputs. These recorded files can be replayed to recover from system crashes, run macros for repeated tasks, or be modified and run independently. This capability of building unique scripts allows users to iterate design processes and use the finite element analysis as a "module" in part of a larger script to analyze data. This automation, while convenient, must also be scrutinized at all the means and extremes of the results to insure validity and convergence of the output. All the scripts used in this appendix are written in python and can be run in ABAQUS/CAE version 6.3. The scripts have been commented with a hash (#) in order to understand their functionality. Python uses indentation for its loop and conditional structures and that structure is preserved as best as possible given the limitations of the present format.

Master Script

```
from abaqus import *
from sketch import *
from part import *
from material import *
from section import *
from assembly import *
from load import *
from visualization import *
from interaction import *
from step import *
from mesh import *
from job import *
from odbAccess import *
from shutil import *

import assembly
import regionToolset
import displayGroupMdbToolset as dgm
import part
import step
import interaction
import load
import mesh
import job
import visualization
import xyPlot
import displayGroupOdbToolset as dgo
import material
import section
#.....................................................................................#
#.....................................................................................#
```

#This is the main module that sets all the input parameters and executes the elastic function to determine the mode-mix first, followed by the plastic case to do the actual analysis for each iteration.#

```
#.....................................................................................#
```

def main():

#The first part of the name of all the files

```
name = 'alpha'
```

#Set all the result files with a blank file (overwrites with a 'w' and appends with an 'a')

```
results = open(name + '.txt', 'w')
results.close()
results = open(name + '-elastic.txt', 'w')
results.close()
```

```
    jobnumber = 0 #initialize the job counter
    spaces = 9. # Spaces between the pure mode 1 and pure mode 2
    plasticzones = 'no' #bit to toggle printing the plastic zones
    h = 5. #layer height in mm
    L = 50. #specimen length in mm
    elm = 0.0002 #smallest element in mm
    Einit = 73.1E3 #Initial elastic modulus in newtons per mm
    vinit = 1/3. #Poisson's Ratio
    initialyieldstr = 300. #newtons per mm
    etlist = [1e-25] #A list of desired tangent modulus ratios (can't be zero)
    sighatlist = [0] #A list of desired strength mismatches
    alphalist = [-.8, -.6, -.4, -.2, 0., .2, .4, .6, .8] #A list of desired
        elastic modulus mismatches
    for meshsize in [2]: #Add more values to do a convergence study in space


            for timestep in [1.]: #Add more values to do a convergence study
                in time


                elasticCase(h, L, meshsize, timestep, elm,
                   alphalist, name, jobnumber, initialyieldstr,
                   spaces, Einit, vinit)
                plasticCase(h, L, meshsize, timestep, elm,
                   alphalist, name, jobnumber, sighatlist,
                   etlist, initialyieldstr, spaces, Einit,
                   vinit, plasticzones)


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def submitJobs(name, number, G, factor, mode):


    "This function submits jobs"

    mymodel = mdb.model['ModeMix']
    a = mdb.model['ModeMix'].rootAssembly

    print('The job with the name ' + name + '-' + `number` + ' is processing...')

    myjob = mdb.Job(name = name + '-' + `number`, model = mymodel.name,
       type = ANALYSIS, explicitPrecision = SINGLE, nodalOutputPrecision
       = SINGLE, description = 'G = ' + `G` + ', Mode2factor = ' + `factor`
       + ', Phase = ' + `mode`, userSubroutine = '', numCpus = 1, preMemory
       = 740.0, standardMemory = 740.0, standardMemoryPolicy = MODERATE,
```

```
        scratch = '', echoPrint = OFF, modelPrint = OFF, contactPrint
        = OFF, historyPrint = OFF)
    a.regenerate()
    myjob.submit()
    myjob.waitForCompletion()
    print ('Done!')


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def plasticCase(h, L, meshsize, timestep, elm, alphalist, name, jobnumber,
sighatlist, etlist, initialyieldstr, spaces, Einit, vinit, plasticzones):


    initjobnum = jobnumber
    #the following lines open a file and read the contents that were written during the elastic
        analysis. The reason for doing this is to eliminate the need to run the elastic case if only
        the plastic properties are changing or a n analysis isn't completed.
    data = open('mode1ratio', 'r')
    Mode1ratioarray = []
    line = data.readline()
    line = float(line[:-1])
    Mode1ratioarray.append(line)
    while line:


            line = data.readline()
            if line:


                line = float(line[:-1])
                Mode1ratioarray.append(line)


        data.close()
    print ('Mode1ratioarray = ' + `Mode1ratioarray`)
    data = open('mode2ratio', 'r')
    Mode2ratioarray = []
    line = data.readline()
    line = float(line[:-1])
    Mode2ratioarray.append(line)
    while line:
```

```
        line = data.readline()
        if line:

              line = float(line[:-1])
              Mode2ratioarray.append(line)


   data.close()
print ('Mode2ratioarray = ' + `Mode2ratioarray`)
data = open('modearray', 'r')
modearray = []
line = data.readline()
line = float(line[:-1])
modearray.append(line)
while line:

        line = data.readline()
        if line:

              line = float(line[:-1])
              modearray.append(line)


   data.close()
print ('modearray = ' + `modearray`)
elastic = 'no' #Turn the elastic bit off because the analyses are slightly different in
   some modules
h1, h2 = buildModel(h, L, meshsize, timestep, elm, elastic)#call the
   function to build the model in ABAQUS. Building the model each time guarantees unifor-
   mity for each analysis.
alphacount = 0
results = open(name + '.txt', 'a')
results.write('Job \talpha \tbeta \tsighat \tEt/E \tM1 \tM2 \tM2
   Ratio \tG \tJ1 \tJ2 \tdW/dN \tModeMix \tdW/dN* (bottom) \tdW/dN*
   (top)\n')
results.close()
header = ''
data = []
for alpha in alphalist:
```

```
beta = alpha/4.
epsilon = 1/(2*pi)*log((1-beta)/(1 + beta))
eta = h1/h2
inveta = 1 / eta
SIGMA = (1 + alpha)/(1-alpha)
A = 1 / ( 1 + SIGMA*( 4*eta + 6*eta**2 + 3*eta**3 )
   )
I = 1 / ( 12*( 1 + SIGMA*eta**3 ) )
#Elastic Properties
vtop = vinit
vbot = (alpha*(2 - 3*vtop) + 4*beta*(vtop - 1) + vtop)/(1
   + alpha*(3 - 4*vtop) + 4*beta*(vtop - 1))
Etopbar = Einit/(1-vtop**2)
Ebotbar = Etopbar/SIGMA
Etop = (1-vtop**2)*Etopbar
Ebot = (1-vbot**2)*Ebotbar
header = header + 'phase\ta = ' + `alpha` + '\t'
for sighat in sighatlist:
   for et in etlist:

       assignMaterials(elastic, et, sighat, initialyieldstr,
          Etop, vtop, Ebot, vbot)
       MyLoads = createLoads(elastic, initialyieldstr,
          spaces,
       Mode1ratioarray[alphacount], Mode2ratioarray[alphacount],
          name)
       line = 0
       for load in MyLoads:

             jobnumber = jobnumber + 1
             line = line + 1
             results = open(name + '.txt', 'a')

             results.write(name + `jobnumber`
                + '\t' + `alpha` + '\t' + `beta`
                + '\t' + `sighat` + '\t' + `et`
                + '\t')
             results.close()
             G, factor = setLoads(name, jobnumber,
                elastic, load)
             submitJobs(name, jobnumber, G, factor,
                modearray[jobnumber-1-initjobnum])
             dwdn, dwdnstar, dwdnstarbot = extractWork(name,
```

```
                                jobnumber, Etop, Ebot, G)
                            if plasticzones == 'yes':

                                 printfig(name, jobnumber,
                                    alpha, factor, modearray[jobnumber-1-initjobnum],
                                    G)

                                results = open(name + '.txt',
                                'a')
                            results.write(`dwdn` + '\t' + `modearray[jobnumber-1-
                            initjobnum]` + '\t' + `dwdnstarbot`

                            + '\t' + `dwdnstar` + '\n')
                            results.close()
                            #Change this line to extract normalizations wrt
                                the top or bottom.
                            data.append(`modearray[jobnumber-1-initjobnum]`
                                + '\t' + `dwdnstar` + '\t')
                            cleanfiles(name, jobnumber)

                    alphacount = alphacount + 1


                mdb.saveAs(name + '.cae')
            graph = open('kgraph.txt', 'a')
            graph.write(header + '\n')
            index = 0
            for row in range(len(Myloads)):


                    index = row
                    for column in range(alphacount):

                            graph.write(data[index])
                            index = index + len(MyLoads)


                graph.close()

#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def elasticCase(h, L, meshsize, timestep, elm, alphalist, name, elasticjobnumber,
initialyieldstr, spaces, Einit, vinit):
```

```
results = open(name + '-elastic.txt', 'a')
results.write('\nTime increment = ' + `0.05/timestep` + '\nMesh
   resolution = ' + `meshsize/2.` + 'x\n')
results.close()

results = open(name + '-elastic.txt', 'a')
results.write('Job \tM1 \tM2 \tRatio \talpha \tbeta \tomega (mode1)
   \tomega (mode2) \tG \tJ-Integral \tJ from K \tK1 \tK2 \tmodemix
   \tepsilon \trealKh \timagKh \tphase \tadjustedmode\n')
results.close()

elastic = 'yes'

h1, h2 = buildModel(h, L, meshsize, timestep, elm, elastic)

modearray = []
Mode1ratioarray = []
Mode2ratioarray = []
alphacount = 0
for alpha in alphalist:


        beta = alpha/4.
        epsilon = 1/(2*pi)*log((1-beta)/(1 + beta))
        eta = h1/h2
        inveta = 1 / eta
        SIGMA = (1 + alpha)/(1-alpha)
        A = 1 / ( 1 + SIGMA*( 4*eta + 6*eta**2 + 3*eta**3 )
           )
        I = 1 / ( 12*( 1 + SIGMA*eta**3 ) )

        #Elastic Properties
        vtop = vinit
        vbot = (alpha*(2 - 3*vtop) + 4*beta*(vtop - 1) + vtop)/(1
           + alpha*(3 - 4*vtop) + 4*beta*(vtop - 1))
        Etopbar = Einit/(1-vtop**2)
        Ebotbar = Etopbar/SIGMA
        Etop = (1-vtop**2)*Etopbar
        Ebot = (1-vbot**2)*Ebotbar
        assignMaterials(elastic, 1, 0, initialyieldstr, Etop,
           vtop, Ebot, vbot)
        if alpha == 0.:

               Mode1ratio, Mode2ratio = 1., -1.

           else:
```

```
                Mode1ratio, Mode2ratio = findfactor(epsilon,
                   alpha)

                omega1, omega2 = determineOmega(Mode1ratio, Mode2ratio,
                alpha)
             Mode1ratioarray.append(Mode1ratio)
             Mode2ratioarray.append(Mode2ratio)
             MyLoads = createLoads(elastic, initialyieldstr, spaces,
                Mode1ratio, Mode2ratio, name + '-elastic')
             for load in MyLoads:

                elasticjobnumber = elasticjobnumber + 1
                results = open(name + '-elastic.txt', 'a')

                results.write(name + '-elastic-' + `elasticjobnumber`
                   + '\t' + `alpha` + '\t' + `beta` + '\t'
                   + `omega1` + '\t' + `omega2` + '\t')
                results.close()
                G, factor = setLoads(name + '-elastic', elasticjobnumber,
                   elastic, load)
                submitJobs(name + '-elastic', elasticjobnumber,
                   G, factor, 9999)
                modemix, realKh, imagKh = extractMode(name
                   + '-elastic', elasticjobnumber, h, epsilon)
                modearray.append(modemix)
                cleanfiles(name + '-elastic', elasticjobnumber)


          alphacount = 1 + alphacount
       mdb.saveAs(name + '-elastic.cae')

       data = open('mode1ratio', 'w')
       for item in Mode1ratioarray:


             data.write(`item` + '\n')


          data.close()

       data = open('mode2ratio', 'w')
       for item in Mode2ratioarray:


             data.write(`item` + '\n')
```

```
        data.close()

    data = open('modearray', 'w')
    for item in modearray:


            data.write('item' + '\n')


        data.close()

#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def determineOmega(mode1ratio, mode2ratio, alpha):


    mymodel = mdb.model['ModeMix']
    a = mdb.model['ModeMix'].rootAssembly

    #extract y-coordinates and measure height
    y_top = a.sets['top-right-corner'].vertices[0].pointOn[0][1]
    y_bot = a.sets['bottom-right-corner'].vertices[0].pointOn[0][1]
    y_tip = a.sets['Tip'].vertices[0].pointOn[0][1]
    h1 = y_top - y_tip
    h2 = y_tip - y_bot
    h = h1
    eta = h1/h2

    #extract elastic material properties
    top_material = mdb.model['ModeMix'].material['Top']
    bottom_material = mdb.model['ModeMix'].material['Bot']
    E1 = top_material.elastic.table[0][0]
    E2 = bottom_material.elastic.table[0][0]
    v1 = top_material.elastic.table[0][1]
    v2 = bottom_material.elastic.table[0][1]
    define the shear modulus
    shear1 = E1 / (2*(1 + v1))
    shear2 = E2 / (2*(1 + v2))
    #kappa for plane strain
    k1 = 3-4*v1
    k2 = 3-4*v2
    #Modulus from plane stress to plane strain
    E1 = E1 / (1-v1**2)
    E2 = E2 / (1-v2**2)
```

```
#Evaluate the quantities reported in Suo and Hutchinson [43]
c1 = (k1 + 1)/shear1
c2 = (k2 + 1)/shear2
SIGMA = c2/c1
A = 1.  / ( 1.  + SIGMA*( 4*eta + 6*eta**2 + 3*eta**3 ) )
I = 1.  / ( 12*( 1.  + SIGMA*eta**3 ) )
delta = h1*( 1 + 2*SIGMA*eta + SIGMA*eta**2 ) / ( 2*eta* ( 1 + SIGMA*eta
   ) )
DELTA = delta/h1
Ao = 1 / eta + SIGMA
Io = (1/3.)*( SIGMA*(3*(DELTA - 1/eta)**2 - 3*(DELTA - 1/eta) +
   1) + 3*DELTA/eta*(DELTA-1/eta) + 1/eta**3 )
C1 = SIGMA / Ao
C2 = (SIGMA / Io )*( 1/eta - DELTA + 1./2.)
C3 = SIGMA / (12*Io)
siny = 6*SIGMA*eta**2*( 1 + eta)*(A*I)**0.5
gamma = asin(siny)
P1 = 0
P2 = 0
M1 = 1.
for M2 in [mode2ratio, mode1ratio]:


        P3 = P1 - P2 #eqn (1.1)
        M3 = ( M1 - M2 + P1*(h1/2.  + h2 - delta) + P2*(delta
           - h2/2.)  ) #eqn (1.1)
        P = P1 - C1*P3 - C2*M3/h1 # eqn (1.2)
        M = M1 - C3*M3
        lamda = (I/A)**0.5*P*h/M
        err = 100
        tol = 0.0001
        omega = [49.0*pi/180., 50.0*pi/180.]
        lhs = []
        for k in range(2):

               if M2 == mode2ratio:

                      lhs.append(lamda*cos(omega[k]) +
                         sin(omega[k] + gamma))
                      err = abs(lhs[k])

                  if M2 == mode1ratio:
```

```
                    lhs.append(lamda*sin(omega[k])-cos(omega[k]
                       + gamma))
                    err = abs(lhs[k])

                if err < tol:

                    break

            while err > tol:

                omega.append(omega[k] - (lhs[k]*(omega[k]
                   - omega[k-1]))/(lhs[k]-lhs[k-1]) )
                k = k + 1
                if M2 == mode2ratio:

                    lhs.append(lamda*cos(omega[k]) +
                       sin(omega[k] + gamma))
                    print 'mode2'

                   if M2 == mode1ratio:

                    lhs.append(lamda*sin(omega[k])-cos(omega[k]
                       + gamma))
                    print 'mode1'

                   print ('k = ' + `k`)
                print ('lhs = ' + `lhs`)
                err = abs(lhs[k])

            if M2 == mode2ratio:

                omega2 = omega[k]*180/pi
                print ('omega2 = ' + `omega2`)

            if M2 == mode1ratio:

                omega1 = omega[k]*180/pi
                print ('omega1 = ' + `omega1`)


       omega = (omega1 + omega2)/2.
    print ('OMEGA = ' + `omega`)
    return omega1, omega2
```

```
#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def findfactor(epsilon, alpha):


    print "Finding Pure Moment Ratios Using the Secant Method..."
    mymodel = mdb.model['ModeMix']
    a = mdb.model['ModeMix'].rootAssembly
    #extract y-coordinates and measure height
    y_top = a.sets['top-right-corner'].vertices[0].pointOn[0][1]
    y_bot = a.sets['bottom-right-corner'].vertices[0].pointOn[0][1]
    y_tip = a.sets['Tip'].vertices[0].pointOn[0][1]
    h1 = y_top - y_tip
    h2 = y_tip - y_bot
    h = h1
    #generate Loading conditions
    m1 = 1.e3
    for mode in [1, 2]:


        err = 100.
        tol = .5
        if mode == 1:

            point1 = (( 1.0 - 0.831*alpha + 0.00187*alpha**2
               - 3.01*alpha**3 + 5.03*alpha**4 + 8.32*alpha**5
               -12.7*alpha**6 - 10.1*alpha**7 + 12.7*alpha**8))

        else:

            point1 = (( -1.0 + 1.06*alpha - 0.143*alpha**2
               + 3.92*alpha**3 -6.70*alpha**4 -11.1*alpha**5
               + 17*alpha**6 + 3.4*alpha**7 -17.0*alpha**8))

        point = [point1, point1*.95]
        MyLoads = [[m1, point[0]*m1], [m1, point[1]*m1]]
        K1 = []
        K2 = []
        realKh = []
        imagKh = []
        k = 0
        for load in MyLoads:  #vary loads to change modes
```

```
M1 = load[0]
M2 = load[1]
factor = M2/M1
print('alpha = ' + `alpha`)
print('k = ' + `k`)
print('M1 = ' + `M1`)
print('M2 = ' + `M2`)
print('M2 factor = ' + `factor`)
if M2 == 0.:

      M2 = 1e-15

   # Begin Applying forces to model
topcouple = 4*M1/h1**2
bottomcouple = 4*M2/h2**2
mymodel.load['Load-1'].setValues(magnitude
   = topcouple)
mymodel.load['Load-2'].setValues(magnitude
   = -topcouple)
mymodel.load['Load-3'].setValues(magnitude
   = -bottomcouple)
mymodel.load['Load-4'].setValues(magnitude
   = bottomcouple)
#submit the job and wait for completion
myjob = mdb.Job(name = 'findfactor', model
   = mymodel.name, type = ANALYSIS, explicitPrecision
   = SINGLE, nodalOutputPrecision = SINGLE,
   description = 'G = ' + `G` + ' mode2factor
   = ' + `factor`, userSubroutine = '',numCpus
   = 1, preMemory = 512.0, standardMemory
   = 512.0, standardMemoryPolicy = MODERATE,
   scratch = '', echoPrint = OFF, modelPrint
   = OFF, contactPrint = OFF, historyPrint
   = OFF)
name = myjob.name
a.regenerate()
myjob.submit()
myjob.waitForCompletion()
o3 = session.openOdb(name + '.odb')
odb = session.odb[name + '.odb']
#J Integrals
```

```
session.XYDataFromHistory(name = 'Jintegral',
   odb = odb, outputVariableName = 'J-integral
   at of contour 10 on
crackfront node set L 5:  J for Whole Model',
   steps = ('Load', ))
session.XYDataFromHistory(name = 'JfromK',
   odb = odb, outputVariableName = 'J-integral
   estimated from Ks at of contour 10 on crackfront
   node set L 5:  JfK for Whole Model', steps
   = ('Load', ))
session.XYDataFromHistory(name = 'K1', odb
   = odb, outputVariableName = 'Stress intensity
   factor K1 at of contour 10 on crackfront
   node set L 5:  K1 for Whole Model', steps
   = ('Load', ))
session.XYDataFromHistory(name = 'K2', odb
   = odb, outputVariableName = 'Stress intensity
   factor K2 at of contour 10 on crackfront
   node set L 5:  K2 for Whole Model', steps
   = ('Load', ))
JfromK = session.xyDataObjects['JfromK'].data[0][1]
J = session.xyDataObjects['Jintegral'].data[0][1]
K1.append(session.xyDataObjects['K1'].data[0][1])
K2.append(session.xyDataObjects['K2'].data[0][1])
odb.close()
o3.close()
print ('G analyical = ' + `G`)
print('J Integral = ' + `J`)
print('J from K = ' + `JfromK`)
print('K1 = ' + `K1[k]`)
print('K2 = ' + `K2[k]`)
tanpsi = (K2[k]/K1[k])
modemix = atan(K2[k]/K1[k])*180/pi
print('mode-mix = ' + `modemix`)
realKh.append(K1[k]*cos(epsilon*log(h)) -
   K2[k]*sin(epsilon*log(h)))
imagKh.append(K2[k]*cos(epsilon*log(h)) +
   K1[k]*sin(epsilon*log(h)))
phase = atan(imagKh[k]/realKh[k])*180/pi
adjustedmode = modemix + epsilon*log(h)*180/pi
```

```
            print('realKh = ' + `realKh[k]`)
            print('imagKh = ' + `imagKh[k]`)
            print('phase = ' + `phase`)
            print('adjustedmode = ' + `adjustedmode`)
            #clean files
            home = 'c:/Abaqus/'
            filetypes = ('.stt', '.023', '.res', '.sta',
               '.log', '.prt', '.inp', '.ipm', '.mdl',
               '.com')
            for extension in filetypes:

                  file = open(name + extension, 'w')
                  file.close()
                  os.remove(name + extension)

               morefiletypes = ('.odb', '.msg', '.dat')
            for extension in morefiletypes:

                  file = open(name + extension, 'w')
                  file.close()
                  os.remove(name + extension)

               if mode == 1:

                  err = (abs(imagKh[k]))

               else:

                  err = (abs(realKh[k]))

               print ('err = ' + `err`)
            print ('tol = ' + `tol`)
            if err < tol:

                  print ('this should break!!!!!!!!!!!!!!')
                  break

               if len(realKh)<2:

                  k = k + 1

         while err > tol:

            print('alpha = ' + `alpha`)
```

```
print('k = ' + `k`)
if mode == 2:

      point.append( point[k] - (realKh[k]*(point[k]
         - point[k-1]))/(realKh[k]-realKh[k-1])
         )

   else:

      point.append( point[k] - (imagKh[k]*(point[k]
         - point[k-1]))/(imagKh[k]-imagKh[k-1])
         )

   k = k + 1
M1 = m1
M2 = point[k]*m1
factor = M2/M1
print('M1 = ' + `M1`)
print('M2 = ' + `M2`)
print('M2 factor = ' + `factor`)
if M2 == 0.:

      M2 = 1e-15

   # Begin Applying forces to model
topcouple = 4*M1/h1**2
bottomcouple = 4*M2/h2**2
mymodel.load['Load-1'].setValues(magnitude
   = topcouple)
mymodel.load['Load-2'].setValues(magnitude
   = -topcouple)
mymodel.load['Load-3'].setValues(magnitude
   = -bottomcouple)
mymodel.load['Load-4'].setValues(magnitude
   = bottomcouple)
#submit the job and wait for completion
myjob = mdb.Job(name = 'findfactor', model
   = mymodel.name, type = ANALYSIS, explicitPrecision
   = SINGLE, nodalOutputPrecision = SINGLE,
   description = 'G = ' + `G` + ' mode2factor
   = ' + `factor`, userSubroutine = '', numCpus
   = 1, preMemory = 512.0, standardMemory
   = 512.0, standardMemoryPolicy = MODERATE,
```

```
   scratch = '', echoPrint = OFF, modelPrint
   = OFF, contactPrint = OFF, historyPrint
   = OFF)
name = myjob.name
a.regenerate()
myjob.submit()
myjob.waitForCompletion()
o3 = session.openOdb(name + '.odb')
odb = session.odb[name + '.odb']
#J Integrals
session.XYDataFromHistory(name = 'Jintegral',
   odb = odb, outputVariableName = 'J-integral
   at of contour 10 on crackfront node set
   L 5:  J for Whole Model', steps = ('Load',
   ))
session.XYDataFromHistory(name = 'JfromK',
   odb = odb, outputVariableName = 'J-integral
   estimated from Ks at of contour 10 on crackfront
   node set L 5:  JfK for Whole Model', steps
   = ('Load', ))
session.XYDataFromHistory(name = 'K1', odb
   = odb, outputVariableName = 'Stress intensity
   factor K1 at of contour 10 on crackfront
   node set L 5:  K1 for Whole Model', steps
   = ('Load', ))
session.XYDataFromHistory(name = 'K2', odb
   = odb, outputVariableName = 'Stress intensity
   factor K2 at of contour 10 on crackfront
   node set L 5:  K2 for Whole Model', steps
   = ('Load', ))
JfromK = session.xyDataObjects['JfromK'].data[0][1]
J = session.xyDataObjects['Jintegral'].data[0][1]
K1.append(session.xyDataObjects['K1'].data[0][1])
K2.append(session.xyDataObjects['K2'].data[0][1])
odb.close()
o3.close()
print ('G analyical = ' + `G`)
print('J Integral = ' + `J`)
print('J from K = ' + `JfromK`)
print('K1 = ' + `K1[k]`)
```

```
print('K2 = ' + `K2[k]`)
tanpsi = (K2[k]/K1[k])
modemix = atan(K2[k]/K1[k])*180/pi
print('mode-mix = ' + `modemix`)
realKh.append(K1[k]*cos(epsilon*log(h)) -
   K2[k]*sin(epsilon*log(h)))
imagKh.append(K2[k]*cos(epsilon*log(h)) +
   K1[k]*sin(epsilon*log(h)))
phase = atan(imagKh[k]/realKh[k])*180/pi
adjustedmode = modemix + epsilon*log(h)*180/pi
print('realKh = ' + `realKh[k]`)
print('imagKh = ' + `imagKh[k]`)
print('phase = ' + `phase`)
print('adjustedmode = ' + `adjustedmode`)
#clean files
home = 'c:/Abaqus/'
filetypes = ('.stt', '.023', '.res', '.sta',
   '.log', '.prt', '.inp', '.ipm', '.mdl',
   '.com')
for extension in filetypes:

      file = open(name + extension, 'w')
      file.close()
      os.remove(name + extension)

   morefiletypes = ('.odb', '.msg', '.dat')
for extension in morefiletypes:

      file = open(name + extension, 'w')
      file.close()
      os.remove(name + extension)

   if mode == 1:

      err = (abs(imagKh[k]))

   else:

      err = (abs(realKh[k]))

   print ('points = ' + `point`)
print ('RealK = ' + `realKh`)
print ('ImagK = ' + `imagKh`)
```

```
                    if mode == 1:

                          Mode1Ratio = point[k]
                          print('Mode1Ratio = ' + `Mode1Ratio`)

                       else:

                          Mode2Ratio = point[k]
                          print('Mode2Ratio = ' + `Mode2Ratio`)


         print "Done!"
      return Mode1Ratio, Mode2Ratio


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def setLoads(name, number, elastic, load):


      print "Setting Loads..."
      mymodel = mdb.model['ModeMix']
      a = mdb.model['ModeMix'].rootAssembly
      #extract y-coordinates and measure height
      y_top = a.sets['top-right-corner'].vertices[0].pointOn[0][1]
      y_bot = a.sets['bottom-right-corner'].vertices[0].pointOn[0][1]
      y_tip = a.sets['Tip'].vertices[0].pointOn[0][1]
      h1 = y_top - y_tip
      h2 = y_tip - y_bot
      eta = h1/h2
      #extract elastic material properties
      top_material = mdb.model['ModeMix'].material['Top']
      bottom_material = mdb.model['ModeMix'].material['Bot']
      E1 = top_material.elastic.table[0][0]
      E2 = bottom_material.elastic.table[0][0]
      v1 = top_material.elastic.table[0][1]
      v2 = bottom_material.elastic.table[0][1]
      #define the shear modulus
      shear1 = E1 / (2*(1 + v1))
      shear2 = E2 / (2*(1 + v2))
      #kappa for plane strain
      k1 = 3-4*v1
```

```
k2 = 3-4*v2
#Modulus from plane stress to plane strain
E1 = E1 / (1-v1**2)
E2 = E2 / (1-v2**2)
c1 = (k1 + 1)/shear1
c2 = (k2 + 1)/shear2
SIGMA = c2/c1
A = 1 / ( 1 + SIGMA*( 4*eta + 6*eta**2 + 3*eta**3 ) )
I = 1 / ( 12*( 1 + SIGMA*eta**3 ) )
delta = h1*( 1 + 2*SIGMA*eta + SIGMA*eta**2 ) / ( 2*eta* ( 1 + SIGMA*eta
   ) )
DELTA = delta/h1
M1 = load[0]
M2 = load[1]
if M2 == 0.:

       M2 = 1e-15

   if M1 == 0.:

       M1 = 1e-15


   factor = M2/M1
print('M2 factor = ' + `factor`)
P1 = 0
P2 = 0
P3 = P1 - P2 #eqn (1.1)
M3 = ( M1 - M2 + P1*(h1/2.  + h2 - delta) + P2*(delta - h2/2.)  )
   #eqn (1.1)
Ao = 1 / eta + SIGMA
Io = (1/3.)*( SIGMA*(3*(DELTA - 1/eta)**2 - 3*(DELTA - 1/eta) +
   1) + 3*DELTA/eta*(DELTA-1/eta) + 1/eta**3 )
C1 = SIGMA / Ao
C2 = (SIGMA / Io )*( 1/eta - DELTA + 1./2.)
C3 = SIGMA / (12*Io)
P = P1 - C1*P3 - C2*M3/h1 # eqn (1.2)
M = M1 - C3*M3
siny = 6*SIGMA*eta**2*( 1 + eta)*(A*I)**0.5
gamma = asin(siny)
```

```
    #eqn (2.7)
    G = (c1/16.0)*( P**2/(A*h1) + M**2/(I*h1**3) + 2*P*M*siny/((A*I)**0.5*h1**2)
       )
    print ('G = ' + 'G')

    topcouple = 4*M1/h1**2
    bottomcouple = 4*M2/h2**2
    mdb.model['ModeMix'].load['Load-1'].setValues(magnitude = topcouple)
    mdb.model['ModeMix'].load['Load-2'].setValues(magnitude = -topcouple)
    mdb.model['ModeMix'].load['Load-3'].setValues(magnitude = -bottomcouple)
    mdb.model['ModeMix'].load['Load-4'].setValues(magnitude = bottomcouple)
    if elastic!  = 'yes':


            mdb.model['ModeMix'].load['Reload-1'].setValues(magnitude
               = topcouple)
            mdb.model['ModeMix'].load['Reload-2'].setValues(magnitude
               = -topcouple)
            mdb.model['ModeMix'].load['Reload-3'].setValues(magnitude
               = -bottomcouple)
            mdb.model['ModeMix'].load['Reload-4'].setValues(magnitude
               = bottomcouple)


       results = open(name + '.txt', 'a')
    results.write('M1' + '\t' + 'M2' + '\t' + 'factor' + '\t' + 'G'
       + '\t')
    results.close()
    print "Done..."
    return G, factor


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def assignMaterials(elastic, Et, sighat, initialyieldstr, Etop, vtop, Ebot,
vbot):


    "This Assigns the materials according to the properties"
    print ('Assigning Materials...')
    mymodel = mdb.model['ModeMix']
    if elastic == 'yes':
```

```
        mymodel.material['Top'].Elastic(table = ((Etop, vtop),
           ))
        mymodel.material['Bot'].Elastic(table = ((Ebot, vbot),
           ))

    else:

        mymodel.material['Top'].Elastic(table = ((Etop, vtop),
           ))
        mymodel.material['Bot'].Elastic(table = ((Ebot, vbot),
           ))
        #Plastic Properties
        EttopoverE = Et
        EtbotoverE = Et
        if sighat > 0:

               yieldstr = initialyieldstr*(1.  + sighat)/(1-sighat)

           else:

               yieldstr = initialyieldstr

           stress = yieldstr*1.0002
        topplastictable = [(yieldstr, 0.0)]
        plasticstrain = ((stress-yieldstr)/EttopoverE - stress
           + yieldstr)/Etop
        totalstrain = stress/Etop + plasticstrain
        topplastictable.append((stress, plasticstrain))
        top_material = mymodel.Material(name = 'Top')
        mymodel.material['Top'].Elastic(table = ((Etop, vtop),
           ))
        mymodel.material['Top'].Plastic(table = topplastictable
           )
        mymodel.material['Top'].plastic.setValues(hardening
           = KINEMATIC)
        if sighat < 0:

               yieldstr = initialyieldstr*(1.-sighat)/(1
                  + sighat)

           else:
```

```
                yieldstr = initialyieldstr

                stress = yieldstr*1.0002
            botplastictable = [(yieldstr, 0.0)]
            plasticstrain = ((stress-yieldstr)/EtbotoverE - stress
                + yieldstr)/Ebot
            totalstrain = stress/Ebot + plasticstrain
            botplastictable.append((stress, plasticstrain))
            bottom_material = mymodel.Material(name = 'Bot')
            mymodel.material['Bot'].Elastic(table = ((Ebot, vbot),
                ))
            mymodel.material['Bot'].Plastic(table = botplastictable
                )
            mymodel.material['Bot'].plastic.setValues(hardening
                = KINEMATIC)


        #assign material to section
    section1 = mymodel.section['Top']
    section2 = mymodel.section['Bottom']
    section1.setValues(material = top_material.name, thickness = 1.0)
    section2.setValues(material = bottom_material.name, thickness =
        1.0)
    print ('Done!')


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def createLoads(elastic, initialyield, spaces, Mode1ratio, Mode2ratio, name):


    print ('Creating Loads...')
    mymodel = mdb.model['ModeMix']
    a = mdb.model['ModeMix'].rootAssembly
    top_material = mdb.model['ModeMix'].material['Top']
    bottom_material = mdb.model['ModeMix'].material['Bot']
    #extract elastic material properties
    E1 = top_material.elastic.table[0][0]
    E2 = bottom_material.elastic.table[0][0]
    v1 = top_material.elastic.table[0][1]
    v2 = bottom_material.elastic.table[0][1]
    #extract y-coordinates and measure height
```

```
y_top = a.sets['top-right-corner'].vertices[0].pointOn[0][1]
y_bot = a.sets['bottom-right-corner'].vertices[0].pointOn[0][1]
y_tip = a.sets['Tip'].vertices[0].pointOn[0][1]
h1 = y_top - y_tip
h2 = y_tip - y_bot
if elastic == "yes":


        yield1 = initialyield
        yield2 = initialyield


   else:


        yield1 = top_material.plastic.table[0][0]
        yield2 = bottom_material.plastic.table[0][0]
        sigmahat = (yield1 - yield2) / (yield1 + yield2)
        #extract plastic material properties for top layer
        stress1top = top_material.plastic.table[1][0]
        stress0top = top_material.plastic.table[0][0]
        plstrain1top = top_material.plastic.table[1][1]
        plstrain0top = top_material.plastic.table[0][1]
        totstrain1top = plstrain1top + stress1top/E1
        totstrain0top = plstrain0top + stress0top/E1
        Ettop = (stress1top-stress0top)/(totstrain1top-totstrain0top)
        EttopoverE = Ettop/E1
        #extract plastic material properties for bottom layer
        stress1bot = bottom_material.plastic.table[1][0]
        stress0bot = bottom_material.plastic.table[0][0]
        plstrain1bot = bottom_material.plastic.table[1][1]
        plstrain0bot = bottom_material.plastic.table[0][1]
        totstrain1bot = plstrain1bot + stress1bot/E2
        totstrain0bot = plstrain0bot + stress0bot/E2
        Etbot = (stress1bot-stress0bot)/(totstrain1bot-totstrain0bot)
        EtbotoverE = Etbot/E2


    #define the shear modulus
shear1 = E1 / (2*(1 + v1))
shear2 = E2 / (2*(1 + v2))
#kappa for plane strain
k1 = 3-4*v1
```

```
k2 = 3-4*v2
#Modulus from plane stress to plane strain
E1 = E1 / (1-v1**2)
E2 = E2 / (1-v2**2)
#extract heights from the model
#define parts
top = mdb.model['ModeMix'].part['Top']
bottom = mdb.model['ModeMix'].part['Bottom']
#define the quantities in eqn (2.8)
# define the compliance parameters eqn (2.5)
c1 = (k1 + 1)/shear1
c2 = (k2 + 1)/shear2
SIGMA = c2/c1
#Calculate eta
eta = h1/h2
inveta = 1 / eta
#Calculate A, I and siny from eqn (2.8)
A = 1 / ( 1 + SIGMA*( 4*eta + 6*eta**2 + 3*eta**3 ) )
I = 1 / ( 12*( 1 + SIGMA*eta**3 ) )
siny = 6*SIGMA*eta**2*( 1 + eta)*(A*I)**0.5
gamma = asin(siny)
# eqns AIII.1
delta = h1*( 1 + 2*SIGMA*eta + SIGMA*eta**2 ) / ( 2*eta* ( 1 + SIGMA*eta
   ) )
DELTA = delta/h1 #big DELTA is Normalized
# eqn AIII.3
Ao = 1 / eta + SIGMA
Io = (1/3.)*( SIGMA*(3*(DELTA - 1/eta)**2 - 3*(DELTA - 1/eta) +
   1) + 3*DELTA/eta*(DELTA-1/eta) + 1/eta**3 )
#eqns AIII.6
C1 = SIGMA / Ao
C2 = (SIGMA / Io )*( 1/eta - DELTA + 1./2.)
C3 = SIGMA / (12*Io)
#define Dundurs' parameters: (eqn 2.1)
GAMMA = shear1/shear2
alpha = ( GAMMA*(k2 + 1) - (k1 + 1) ) / ( GAMMA*(k2 + 1) + (k1 +
   1) )
beta = ( GAMMA*(k2 - 1) - (k1 - 1) ) / ( GAMMA*(k2 + 1) + (k1 +
   1) )
start = Mode1ratio
```

```
finish = Mode2ratio
delt = (finish - start)/spaces
MyLoads = []
for index in range(spaces + 1): #Add one because range() returns 0 as the first
    value


        for level in [2]: #use this loop to change the load levels


                factor = start + delt*index
                #yield from #1 of separated beam
                m1e = yield1*(h1**3/12.)/(level*h1/2.)
                #yield from #2 of separated beam
                m1f = yield2*(h2**3/12.)/(level*factor*h2/2.
                   + 1e-15)
                m1 = (min(m1e**2, m1f**2))**.5
                #use Mag K as a common factor
                Gin = .2
                magKsquared = Gin*min(E1, E2)
                m1 = (h1**3*magKsquared*(3*alpha**2 - 4)*(beta**2
                   - 1))/((sqrt(3)*(-h1**3*magKsquared*(3*alpha**2
                   - 4)*(-(6*alpha 7)*(alpha - 1)**2 + factor**2*(alpha
                   + 1)**2*(6*alpha - 7) + 2*factor*(alpha**2
                   - 1))*(beta**2 - 1))**0.5))
                MyLoads.append([m1, m1*factor])


    if elastic !  = 'yes':


        results = open(name + '.txt', 'a')
        results.write('\nMaterial Properties:\nEtop \tEbot \talpha
           \tbeta \tv1 \tv2 \th1 \t h1 \tyield1 \t yield2 \tEttop
           \tEtbot \n')
        results.write('%10.4E' %E1 + '\t%10.4E' %E2 + '\t' +
           `alpha` + '\t' + `beta` + '\t' + `v1` + '\t' + `v2`
           + '\t' + `h1` + '\t' + `h2` + '\t' + `yield1` + '\t'
           + `yield2` + '\t' + `Ettop` + '\t' + `Etbot` + '\n\n')
        results.close()


    print ('Done!')
return MyLoads
```

```
#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def extractMode(origname, number, h, epsilon):


    print ('Extractiong Mode...')
    name = origname + '-' + `number`
    o3 = session.openOdb(name + '.odb')
    odb = session.odb[name + '.odb']
    #J Integrals
    session.XYDataFromHistory(name = 'Jintegral', odb = odb, outputVariableName
       = 'J-integral at of contour 10 on crackfront node set L 5:  J
       for Whole Model', steps = ('Load', ))
    session.XYDataFromHistory(name = 'JfromK', odb = odb, outputVariableName
       = 'J-integral estimated from Ks at of contour 10 on crackfront
       node set L 5:  JfK for Whole Model', steps = ('Load', ))
    session.XYDataFromHistory(name = 'K1', odb = odb, outputVariableName
       = 'Stress intensity factor K1 at of contour 10 on crackfront
       node set L 5:  K1 for Whole Model', steps = ('Load', ))
    session.XYDataFromHistory(name = 'K2', odb = odb, outputVariableName
       = 'Stress intensity factor K2 at of contour 10 on crackfront
       node set L 5:  K2 for Whole Model', steps = ('Load', ))
    JfromK = session.xyDataObjects['JfromK'].data[0][1]
    J = session.xyDataObjects['Jintegral'].data[0][1]
    K1 = session.xyDataObjects['K1'].data[0][1]
    K2 = session.xyDataObjects['K2'].data[0][1]
    odb.close()
    o3.close()
    print ('G analyical = ' + `G`)
    print('J Integral = ' + `J`)
    print('J from K = ' + `JfromK`)
    print('K1 = ' + `K1`)
    print('K2 = ' + `K2`)
    tanpsi = (K2/K1)
    modemix = atan(K2/K1)*180/pi
    print('mode-mix = ' + `modemix`)
    realKh = K1*cos(epsilon*log(h))-K2*sin(epsilon*log(h))
    imagKh = K2*cos(epsilon*log(h)) + K1*sin(epsilon*log(h))
    phase = atan(imagKh/realKh)*180/pi
    adjustedmode = modemix + epsilon*log(h)*180/pi
```

```
    print('realKh = ' + `realKh`)
    print('imagKh = ' + `imagKh`)
    print('phase = ' + `phase`)
    print('adjustedmode = ' + `adjustedmode`)
    results = open(origname + '.txt', 'a')
    results.write(`J` + '\t' + `JfromK` + '\t' + `K1` + '\t' + `K2`
       + '\t' + '%4.2f' %modemix + '\t' + `epsilon` + \t' + `realKh`
       + '\t' + `imagKh` + '\t' + `phase` + '\t' + `adjustedmode` +
       '\n')
    results.close()
    print ('Done!')
    return phase, realKh, imagKh


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def extractWork(origname, jobnumber, E1, E2, G):


    print ('Extractiong Work...')
    name = mdb.job[origname + '-' + `jobnumber`].name
    top_material = mdb.model['ModeMix'].material['Top']
    bottom_material = mdb.model['ModeMix'].material['Bot']
    yield1 = top_material.plastic.table[0][0]
    yield2 = bottom_material.plastic.table[0][0]
    myodb = session.openOdb(name + '.odb')
    odb = session.odb[name + '.odb']
    for stepname in myodb.step.keys():


        history = myodb.step[stepname].historyRegion['Assembly
           ASSEMBLY']
        pddata = history.historyOutput['ALLPD'].data
        if stepname == myodb.step.keys()[1]:

            w2 = pddata[-1][-1]

        if stepname == myodb.step.keys()[3]:

            w4 = pddata[-1][-1]


    if yield2<yield1:
```

```
        dWdNStartop = (w4-w2)*yield2**2 / (G**2 * E1)
        dWdNStarbot = (w4-w2)*yield2**2 / (G**2 * E2)


    else:


        dWdNStartop = (w4-w2)*yield1**2 / (G**2 * E1)
        dWdNStarbot = (w4-w2)*yield1**2 / (G**2 * E2)


    dwdn = w4-w2
#open and close the output file so real time access is enabled
#J Integrals
for k in range(1, 50):


        if k < 10:

              session.XYDataFromHistory(name = 'J' + `k`,
                 odb = odb, outputVariableName = 'J-integral
                 at of contour ' + `k` + ' on crackfront
                 node set L 5:  J for Whole Model', steps
                 = ('Load', 'Unload', 'Reload', 'Reunload',
                 ) )

           if k > 9:

              session.XYDataFromHistory(name = 'J' + `k`,
                 odb = odb, outputVariableName = 'J-integral
                 at of contour ' + `k` + ' on crackfront
                 node set L 5:  J for Whole Model', steps
                 = ('Load', 'Unload', 'Reload', 'Reunload',
                 ) )


    for k in range(1, 50):


        J = session.xyData['J' + `k`].data
        for i in range(len(J)):

              if J[i][0] == 1.0:

                    J1 = J[i][1]

                 elif J[i][0] == 3.0:
```

```
                        J2 = J[i][1]


        myodb.close()
    print ('dW/dN = ' + `dwdn`)
    print ('dW*/dN* (top) = ' + `dWdNStartop`)
    print ('dW*/dN* (bot) = ' + `dWdNStarbot`)
    print 'J Integrals'
    print `J1` + ' ' + `J2`
    results = open(origname + '.txt', 'a')
    results.write(`J1` + '\t' + `J2` + '\t')
    results.close()
    print ('Done!')
    return dwdn, dWdNStartop, dWdNStarbot


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def printfig(name, number, alpha, factor, mode, G):


    print ('Printing Plastic Zones...')
    o0 = session.openOdb(name + '-' + `number` + '.odb')
    session.viewports['Viewport:  1'].setValues(displayedObject = o0)
    for key in session.text.keys():


        del session.text[key]


      for key in session.arrow.keys():


        del session.arrow[key]


      session.viewport['Viewport:  1'].odbDisplay.setPlotMode(CONTOUR)
    for zoomfactor in [200, 400, 800., 1600.]:
      for stepnumber in [0, 3]:


        session.viewport['Viewport:  1'].odbDisplay.setFrame(step
          = 0, frame = 1)
        session.viewport['Viewport:  1'].odbDisplay.contourOptions.
          setValues( numIntervals = 2, spectrumType = WHITE_TO_BLACK,
```

```
   outsideLimitsAboveColor = 'Grey60', outsideLimitsBelowColor
   = 'White', deformationScaling = UNIFORM, uniformScaleFactor
   = 1, maxAutoCompute = OFF, maxValue = 0.15, minAutoCompute
   = OFF, minValue = 0.1, )
session.viewport['Viewport:  1'].odbDisplay.contourOptions.
   setValues( modelShape = UNDEFORMED)
session.viewport['Viewport:  1'].odbDisplay.setPrimaryVariable(
   variableLabel = 'AC YIELD', outputPosition = INTEGRATION_POINT
   )
session.viewport['Viewport:  1'].setValues(origin =
   (0.0, 0.0), width = 160, height = 164)
session.viewport['Viewport:  1'].view.zoom(zoomFactor
   = zoomfactor, mode = ABSOLUTE)
session.viewport['Viewport:  1'].viewportAnnotationOptions.
setValues(triad = OFF, legend = OFF,
legendBox = OFF, title = OFF, state = OFF)
session.textDefaults.setValues(color = 'Black')
session.arrowDefaults.setValues(color = 'Black')
session.textDefaults.setValues( font = '-*-times-medium-i-normal
   -*-*-240-*-*-p-*-*-*')
session.Arrow(name = 'Left Dim', startPoint = (60.,
   -6.), endPoint = (2., -6.))
session.Arrow(name = 'Right Dim', startPoint = (100.,
   -6.), endPoint = (158., -6.))
session.Arrow(name = 'X Axis', startPoint = (80., 80.),
   endPoint = (170., 80.))
session.Arrow(name = 'Y Axis', startPoint = (80.0, 80.0),
   endPoint = (80.0, 170.))
session.Text(name = 'X Label', origin = (166., 83.),
   text = 'x')
session.Text(name = 'Y label', origin = (83., 166.),
   text = 'y')
session.Text(name = 'Dimension', origin = (66., -8),
   text = `64000/zoomfactor`)
session.Text(name = 'Unit1', origin = (90., -8.), text
   = 'm')
session.textDefaults.setValues( font = '-*-symbol-medium-i-normal
   -*-*-240-*-*-p-*-*-*')
session.Text(name = 'Unit2', origin = (86., -8.), text
   = 'm')
```

```
session.textDefaults.setValues(font = '-*-arial-medium-r-normal
   -*-*-240-*-*-p-*-*-*')
session.Text(name = 'Left Tick', origin = (0., -9.),
   text = 'I')
session.Text(name = 'Right Tick', origin = (158., -9.
   ), text = 'I')
session.epsOptions.setValues(imageSize = (2.50, 3.0),
   units = INCHES, resolution = DPI_300, fontType =
   PS_IF_AVAILABLE)
session.printOptions.setValues(rendition = GREYSCALE,
   vpDecorations = OFF, vpBackground = OFF)
session.Text(name = 'Text:  1', origin = (44., -18),
   text = 'a')
session.Text(name = 'Text:  2', origin = (51, -18),
   text = ' = ' + `alpha`)
session.Text(name = 'Text:  3', origin = (44, -28),
   text = 'y')
session.Text(name = 'Text:  4', origin = (51, -28),
   text = ' = %4.3f' %mode)
session.Text(name = 'Text:  5', origin = (44, -38),
   text = 'G = %5.3f' %(100.*G))
session.Text(name = 'Text:  6', origin = (83, -38),
   text = 'J/m/m')
if stepnumber == 0:

      session.Text(name = 'Text:  7', origin = (0,
         -18), text = 'Fwd.  Zone:')

   else:

      session.Text(name = 'Text:  7', origin = (0,
         -18), text = 'Rev.  Zone:')

   session.texts['Text:  3'].setValues( font = '-*-symbol
   -medium-r-normal-*-*-240-*-*-p-*-*-*')
session.texts['Text:  1'].setValues( font = '-*-symbol
   -medium-r-normal-*-*-240-*-*-p-*-*-*')
session.texts['Text:  5'].setValues( font = '-*-times
   new roman-medium-r-normal-*-*-240-*-*-p-*-*-*')
session.texts['Text:  4'].setValues( font = '-*-times
   new roman-medium-r-normal-*-*-240-*-*-p-*-*-*')
```

```
            session.texts['Text:  2'].setValues( font = '-*-times
               new roman-medium-r-normal-*-*-240-*-*-p-*-*-*')
            session.texts['Text:  6'].setValues( font = '-*-times
               new roman-medium-i-normal-*-*-240-*-*-p-*-*-*')
            session.texts['Text:  7'].setValues( font = '-*-times
               new roman-medium-r-normal-*-*-240-*-*-p-*-*-*')
            if stepnumber == 0:

                file = name + '-' + `number` + '-load-' +
                   `zoomfactor` + '.eps'

               else:

                file = name + '-' + `number` + '-unload-'
                   + `zoomfactor` + '.eps'

               session.printToFile(fileName = file, format = EPS,
               canvasObjects = ( session.texts['Text:  7'], session.texts['Text:
               6'], session.texts['Text:  5'], session.texts['Text:
               4'], session.texts['Text:  3'], session.texts['Text:
               2'], session.texts['Text:  1'], session.texts['Right
               Tick'], session.texts['Left Tick'], session.texts['Unit2'],
               session.texts['Unit1'], session.texts['Dimension'],
               session.texts['Y label'], session.texts['X Label'],
               session.arrows['Y Axis'], session.arrows['X Axis'],
               session.arrows['Right Dim'], session.arrows['Left
               Dim'], session.viewports['Viewport:  1']))


         o0.close()
      print ('Done!')


#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def buildModel(h, L, meshsize, timestep, elm, elastic):


      "This function builds a model"
      #create the model database
      Mdb
      #create the model
      mymodel = mdb.Model('ModeMix')
      if mdb.model.keys()[0] == 'Model-1':
```

```
        del mdb.model['Model-1']


    #create the sketech profile
s = mymodel.Sketch(name = '__profile__', sheetSize = L)
g, v, d = s.geometry, s.vertex, s.dimension
s.setPrimaryObject(option = STANDALONE)
s.rectangle(point1 = (-L/2., 0.0), point2 = (L/2., h))
#create the top part
p = mymodel.Part(name = 'Top', dimensionality = TWO_D_PLANAR, type
   = DEFORMABLE_BODY)
p.BaseShell(sketch = s)
s.unsetPrimaryObject()
#delete the sketch profile
del mdb.model['ModeMix'].sketch['__profile__']
#begin defining partitions
p0 = mdb.model['ModeMix'].part['Top']
f, e, d0 = p0.face, p0.edge, p0.datum
t = p0.MakeSketchTransform(sketchPlane = f[0], sketchPlaneSide =
   SIDE1)
s0 = mdb.model['ModeMix'].Sketch(name = '__profile__', sheetSize
   = L, gridSpacing = 10.0, transform = t)
g, v, d = s0.geometry, s0.vertex, s0.dimension
s0.setPrimaryObject(option = SUPERIMPOSE)
p0 = mdb.model['ModeMix'].part['Top']
p0.projectReferencesOntoSketch(sketch = s0, filter = COPLANAR_EDGES)
r, r0 = s0.referenceGeometry, s0.referenceVertex
#draw the horizontal partition
s0.Line(point1 = (-L/2., 0.0), point2 = (L/2.0, 0.0))
#draw the vertical partition
s0.Line(point1 = (0.0, h/2.), point2 = (0.0, -h/2.))
#draw the biasing box
s0.rectangle(point1 = (-h/2., h/2.), point2 = (h/2., -h/2.))
#draw the crack tip box
s0.rectangle(point1 = (-4*elm, -h/2.  + 4*elm), point2 = (4*elm,
   -h/2.))
#draw the radials
s0.Line(point1 = (4*elm, -h/2.  + 4*elm), point2 = (h/2., 0.0))
s0.Line(point1 = (-h/2., 0.0), point2 = (-4*elm, -h/2.  + 4*elm))
f, e, d0 = p0.face, p0.edge, p0.datum
faces = (f[0], )
```

```
p0.PartitionFaceBySketch(faces = faces, sketch = s0)
s0.unsetPrimaryObject()
del mdb.model['ModeMix'].sketch['__profile__']
#Copy top to bottom
mdb.model['ModeMix'].Part('Bottom', mdb.model['ModeMix'].part['Top'])
#create-materials
mdb.model['ModeMix'].Material('Top')
mdb.model['ModeMix'].Material('Bot')
mdb.model['ModeMix'].HomogeneousSolidSection(name = 'Top', material
   = 'Top', thickness = 1.0)
mdb.model['ModeMix'].HomogeneousSolidSection(name = 'Bottom', material
   = 'Bot', thickness = 1.0)
#create assembly
a = mdb.model['ModeMix'].rootAssembly
a.DatumCsysByDefault(CARTESIAN)
p = mdb.model['ModeMix'].part['Bottom']
a.Instance(name = 'Bottom-1', part = p)
p2 = a.instance['Bottom-1']
p2.rotateAboutAxis(axisPoint = (0.0, 0.0, 0.0), axisDirection =
   (0.0, 0.0, 1.0), angle = 180.0)
p = mdb.model['ModeMix'].part['Top']
a.Instance(name = 'Top-1', part = p)
p1 = mdb.model['ModeMix'].part['Top']
f = p1.face
faces = f[0:12]
region = (faces, )
p0 = mdb.model['ModeMix'].part['Top']
p0.assignSection(region = region, sectionName = 'Top')
p1 = mdb.model['ModeMix'].part['Bottom']
f = p1.face
faces = f[0:12]
region = (faces, )
p0 = mdb.model['ModeMix'].part['Bottom']
p0.assignSection(region = region, sectionName = 'Bottom')
#define surface sets
e1 = a.instances['Bottom-1'].edges
edges1 = e1[16:17] + e1[19:21]
a.Surface(side1Edges = edges1, name = 'right-bottom-interface')
e1 = a.instances['Bottom-1'].edges
edges1 = e1[6:7] + e1[24:25] + e1[26:27]
```

```
a.Surface(side1Edges = edges1, name = 'left-bottom-interface')
e1 = a.instances['Top-1'].edges
edges1 = e1[6:7] + e1[24:25] + e1[26:27]
a.Surface(side1Edges = edges1, name = 'right-top-interface')
e1 = a.instances['Top-1'].edges
edges1 = e1[16:17] + e1[19:21]
a.Surface(side1Edges = edges1, name = 'left-top-interface')
e1 = a.instances['Top-1'].edges
edges1 = e1[7:9]
e2 = a.instances['Bottom-1'].edges
edges2 = e2[13:14] + e2[15:16]
a.Set(edges = edges1 + edges2, name = 'right-side')
v1 = a.instances['Bottom-1'].vertices
verts1 = v1[11:12]
a.Set(vertices = verts1, name = 'bottom-right-corner')
v1 = a.instances['Top-1'].vertices
verts1 = v1[8:9]
a.Set(vertices = verts1, name = 'top-right-corner')
v1 = a.instances['Top-1'].vertices
verts1 = v1[3:4]
a.Set(vertices = verts1, name = 'top-middle-point')
e1 = a.instances['Top-1'].edges
edges1 = e1[13:14]
a.Surface(side1Edges = edges1, name = 'left-edge-1')
e1 = a.instances['Top-1'].edges
edges1 = e1[15:16]
a.Surface(side1Edges = edges1, name = 'left-edge-2')
e1 = a.instances['Bottom-1'].edges
edges1 = e1[7:8]
a.Surface(side1Edges = edges1, name = 'left-edge-3')
e1 = a.instances['Bottom-1'].edges
edges1 = e1[8:9]
a.Surface(side1Edges = edges1, name = 'left-edge-4')
v1 = a.instance['Top-1'].vertex
verts1 = v1[16:17]
v2 = a.instance['Bottom-1'].vertex
verts2 = v2[16:17]
a.GeometrySet(vertexSeq = (verts1, verts2, ), name = 'Tip')
#extract y-coordinates and measure height
y_top = a.sets['top-right-corner'].vertices[0].pointOn[0][1]
```

```
y_bot = a.sets['bottom-right-corner'].vertices[0].pointOn[0][1]
y_tip = a.sets['Tip'].vertices[0].pointOn[0][1]
h1 = y_top - y_tip
h2 = y_tip - y_bot
print('h1 = ' + `h1`)
print('h2 = ' + `h2`)
if elastic == 'yes':


        mdb.model['ModeMix'].StaticStep(name = 'Load', previous
           = 'Initial', initialInc = 0.05/timestep, maxInc =
           .05/timestep)


    else:


        mdb.model['ModeMix'].StaticStep(name = 'Load', previous
           = 'Initial', initialInc = 0.05/timestep, maxInc =
           0.05/timestep)
        mdb.model['ModeMix'].StaticStep(name = 'Unload', previous
           = 'Load', initialInc = 0.05/timestep, maxInc =0.05/timestep)
        mdb.model['ModeMix'].StaticStep(name = 'Reload', previous
           = 'Unload', initialInc = 0.05/timestep, maxInc =
           .05/timestep)
        mdb.model['ModeMix'].StaticStep(name = 'Reunload', previous
           = 'Reload', initialInc = 0.05/timestep, maxInc =
           .05/timestep)


    #interactions
region1 = a.surfaces['right-top-interface']
region2 = a.surfaces['right-bottom-interface']
mymodel.Tie(name = 'bond', master = region1, slave = region2,
  positionToleranceMethod = COMPUTED, adjust = ON, ieRotations =
   ON)
#Boundary Conditions
region = a.sets['right-side']
mymodel.DisplacementBC(name = 'BC-1', createStepName = 'Initial',
   region = region, u1 = SET, u2 = UNSET, ur3 = UNSET, amplitude
   = UNSET, distribution = UNIFORM, localCsys = None)
region = a.sets['bottom-right-corner']
```

```
mymodel.DisplacementBC(name = 'BC-2', createStepName = 'Initial',
   region = region, u1 = UNSET, u2 = SET, ur3 = UNSET, amplitude
   = UNSET, distribution = UNIFORM, localCsys = None)
#create loading
a = mdb.model['ModeMix'].rootAssembly
region = a.surfaces['left-edge-1']
mdb.model['ModeMix'].Pressure(name = 'Load-1', createStepName =
   'Load', region = region, distribution = UNIFORM, magnitude =
   -1000.0, amplitude = UNSET)
region = a.surfaces['left-edge-2']
mdb.model['ModeMix'].Pressure(name = 'Load-2', createStepName =
   'Load', region = region, distribution = UNIFORM, magnitude =
   -1000.0, amplitude = UNSET)
region = a.surfaces['left-edge-3']
mdb.model['ModeMix'].Pressure(name = 'Load-3', createStepName =
   'Load', region = region, distribution = UNIFORM, magnitude =
   -1000.0, amplitude = UNSET)
region = a.surfaces['left-edge-4']
mdb.model['ModeMix'].Pressure(name = 'Load-4', createStepName =
   'Load', region = region, distribution = UNIFORM, magnitude =
   -1000.0, amplitude = UNSET)
if elastic !  = 'yes':


        mdb.model['ModeMix'].Load('Reload-4', mdb.model['ModeMix'].
           load['Load-4'])
        mdb.model['ModeMix'].Load('Reload-1', mdb.model['ModeMix'].
           load['Load-1'])
        mdb.model['ModeMix'].Load('Reload-3', mdb.model['ModeMix'].
           load['Load-3'])
        mdb.model['ModeMix'].Load('Reload-2', mdb.model['ModeMix'].
           load['Load-2'])
        mdb.model['ModeMix'].load['Load-1'].deactivate('Unload')
        mdb.model['ModeMix'].load['Load-2'].deactivate('Unload')
        mdb.model['ModeMix'].load['Load-3'].deactivate('Unload')
        mdb.model['ModeMix'].load['Load-4'].deactivate('Unload')
        mdb.model['ModeMix'].load['Reload-1'].move('Load', 'Unload')
        mdb.model['ModeMix'].load['Reload-1'].move('Unload',
           'Reload')
        mdb.model['ModeMix'].load['Reload-2'].move('Load', 'Unload')
        mdb.model['ModeMix'].load['Reload-2'].move('Unload',
```

```
   'Reload')
mdb.model['ModeMix'].load['Reload-3'].move('Load', 'Unload')
mdb.model['ModeMix'].load['Reload-3'].move('Unload',
   'Reload')
mdb.model['ModeMix'].load['Reload-4'].move('Load', 'Unload')
mdb.model['ModeMix'].load['Reload-4'].move('Unload',
   'Reload')
mdb.model['ModeMix'].load['Reload-1'].deactivate('Reunload')
mdb.model['ModeMix'].load['Reload-2'].deactivate('Reunload')
mdb.model['ModeMix'].load['Reload-3'].deactivate('Reunload')
mdb.model['ModeMix'].load['Reload-4'].deactivate('Reunload')
```

#create mesh

```
a0 = mdb.model['ModeMix'].rootAssembly
f01 = a0.instance['Bottom-1'].face
f02 = a0.instance['Top-1'].face
regions = (f01[0], f01[1], f01[2], f01[3], f01[4], f01[5], f01[6],
   f01[7], f01[8], f01[9], f01[10], f01[11], f02[0], f02[1], f02[2],
   f02[3], f02[4], f02[5], f02[6], f02[7], f02[8], f02[9], f02[10],
   f02[11])
a0.setMeshControls(regions = regions, technique = FREE)
elemType1 = ElemType(elemCode = CPE8R)
elemType2 = ElemType(elemCode = CPE8R)
f1 = a0.instance['Bottom-1'].face
faces1 = f1[0:12]
f2 = a0.instance['Top-1'].face
faces2 = f2[0:12]
regions = ((faces1, faces2, ), )
a0.setElementType(regions = regions, elemTypes = (elemType1, elemType2))
a0 = mdb.model['ModeMix'].rootAssembly
e01 = a0.instance['Top-1'].edge
e02 = a0.instance['Bottom-1'].edge
edges = (e01[0], e01[2], e01[8], e01[10], e01[13], e01[14], e01[15],
   e01[5], e01[7], e02[14], e02[15], e02[0], e02[2], e02[8], e02[10],
   e02[13], e02[5], e02[7], e02[29], e02[30], e02[1], e02[3], e01[1],
   e01[3], e01[29], e01[30])
a0.seedEdgeByNumber(edges = edges, number = 2*meshsize, constraint
   = FIXED)
e11 = a0.instance['Bottom-1'].edge
e12 = a0.instance['Top-1'].edge
```

```
end1Edges = (e11[4], e11[6], e12[12], e11[12], e12[4], e12[6])
end2Edges = (e11[9], e12[11], e12[16], e11[11], e11[16], e12[9])
edges = ((end1Edges, END1), (end2Edges, END2))
a0.seedEdgeByBias(edges = edges, ratio = 5.0, number = 6*meshsize,
   constraint = FIXED)
e01 = a0.instance['Top-1'].edge
e02 = a0.instance['Bottom-1'].edge
end1Edges = (e01[25], e01[24], e02[25], e02[24])
end2Edges = (e01[18], e01[28], e02[19], e02[18], e02[28], e01[19])
edges = ((end1Edges, END1), (end2Edges, END2))
a0.seedEdgeByBias(edges = edges, ratio = 1.0, number = 20*meshsize,
   constraint = FIXED)
e11 = a0.instance['Bottom-1'].edge
e12 = a0.instance['Top-1'].edge
edges = (e11[26], e11[27], e12[20], e12[22], e12[17], e12[21], e12[23],
   e11[20], e11[22], e12[26], e12[27], e11[17], e11[21], e11[23])
a0.seedEdgeByNumber(edges = edges, number = 2*meshsize, constraint
   = FIXED)
e01 = a0.instance['Top-1'].edge
e02 = a0.instance['Bottom-1'].edge
end1Edges = (e01[25], e01[24], e02[25], e02[24])
end2Edges = (e01[18], e01[28], e02[19], e02[18], e02[28], e01[19])
edges = ((end1Edges, END1), (end2Edges, END2))
a0.seedEdgeByBias(edges = edges, ratio = h/(elm*10.), number = 20*meshsize)
f01 = a0.instances['Bottom-1'].faces
f02 = a0.instances['Top-1'].faces
regions = (f01[1], f01[2], f01[3], f01[4], f02[1], f02[2], f02[3],
   f02[4])
a0.setMeshControls(regions = regions, technique = STRUCTURED)
partInstances = (a0.instance['Bottom-1'], a0.instance['Top-1'],
   )
a0.generateMesh(regions = partInstances)
#Generate output requests
mdb.model['ModeMix'].fieldOutputRequest['F-Output-1'].setValues(
   variables = ( 'S', 'E', 'PE', 'PEEQ', 'U', 'RF', 'CF'), frequency
   = LAST_INCREMENT)
mdb.model['ModeMix'].historyOutputRequest['H-Output-1'].setValues(
   variables = ('ALLPD', ), frequency = LAST_INCREMENT )
#Display the model in the current viewport
```

```
session.viewport['Viewport:  1'].assemblyDisplayOptions.setValues(
   datumPoints = OFF, datumAxes = OFF, datumPlanes = OFF, datumCoordSystems
   = OFF)
session.viewport['Viewport:  1'].setValues(displayedObject = a)

session.viewport['Viewport:  1'].view.fitView()
#Add specific keywords for the interface crack analysis
if elastic == 'yes':


        mymodel.keywordBlock.synchVersions()
        mymodel.keywordBlock.insert(83, """
        *CONTOUR INTEGRAL, CONTOURS = 10, TYPE = K Factors
        Tip, 1, 0""")
        mymodel.keywordBlock.insert(84, """
        *CONTOUR INTEGRAL, CONTOURS = 10, TYPE = J
        Tip, 1, 0""")


   else:


        mdb.model['ModeMix'].keywordBlock.synchVersions()
        mdb.model['ModeMix'].keywordBlock.insert(78, """
        *Contour Integral, contours = 50, type = J
        Tip, 1, 0""")
        mdb.model['ModeMix'].keywordBlock.insert(103, """
        *Contour Integral, contours = 50, type = J
        Tip, 1, 0""")
        mdb.model['ModeMix'].keywordBlock.insert(124, """
        *Contour Integral, contours = 50, type = J
        Tip, 1, 0""")
        mdb.model['ModeMix'].keywordBlock.insert(146, """
        *Contour Integral, contours = 50, type = J
        Tip, 1, 0""")


   return h1, h2

#.....................................................................................#
#.....................................................................................#
#.....................................................................................#
def cleanfiles(name, number):
```

```
    "This Deletea all output files except the ODB file"
    name = name + '-' + `number`
    filetypes = ('.stt', '.023', '.res', '.sta', '.log', '.prt', '.inp',
       '.ipm', '.mdl', '.com')
    for extension in filetypes:


            file = open(name + extension, 'w')
            file.close()
            os.remove(name + extension)


       morefiletypes = ('.msg', '.dat')
    for extension in morefiletypes:


            file = open(name + extension, 'w')
            file.close()
            os.remove(name + extension)

#.....#
#.....#
#.....#
main() #this starts the program after all functions are defined
```

Bibliography

[1] N. W. Klingbeil, "A total dissipated energy theory of fatigue crack growth in ductile solids," *International Journal of Fatigue*, vol. 25, pp. 117–128, 2003.

[2] J. R. Rice, "Mechanics of crack tip deformation and extension by fatigue," *Fatigue Crack Propagation, ASTM STP 415*, pp. 247–311, 1967.

[3] K. N. Raju, "An energy balance criterion for crack growth under fatigue loading from considerations of energy of plastic deformation," *International Journal of Fracture Mechanics*, vol. 8, no. 1, pp. 1–14, 1972.

[4] J. Weertman, "Theory of fatigue crack growth based on a BCS crack theory with work hardening," *International Journal of Fracture*, vol. 9, no. 2, pp. 125–131, 1973.

[5] T. Mura and C. T. Lin, "Theory of fatigue crack growth for work hardening materials," *International Journal of Fracture*, vol. 10, pp. 284–287, 1974.

[6] S. R. Bodner, D. L. Davidson, and J. Lankford, "A description of fatigue crack growth in terms of plastic work," *Engineering Fracture Mechanics*, vol. 17, no. 2, pp. 189–191, 1983.

[7] E. T. Moyer, Jr. and G. C. Sih, "Fatigue analysis of an edge cracked specimen: Hysteresis strain energy density," *Engineering Fracture Mechanics*, vol. 19, no. 2, pp. 643–652, 1984.

[8] D. Kujawski and F. Ellyin, "A fatigue crack propagation model," *Engineering Fracture Mechanics*, vol. 20, pp. 695–704, 1984.

[9] F. Ellyin, "Crack growth rate under cyclic loading and effect of different singularity fields," *Engineering Fracture Mechanics*, vol. 25, pp. 463–473, 1986.

[10] G. C. Sih and D. Y. Jeong, "Fatigue load sequence effect ranked by critical available energy density," *Theoretical and Applied Fracture Mechanics*, vol. 14, pp. 141–151, 1990.

[11] C. L. Chow and T. J. Lu, "Cyclic J-integral in relation to fatigue crack initiation and propagation," *Engineering Fracture Mechanics*, vol. 39, no. 1, pp. 1–20, 1991.

[12] W. Wang and C.-T. T. Hsu, "Fatigue crack growth rate of metal by plastic energy damage accumulation theory," *Journal of Engineering Mechanics*, vol. 120, no. 4, pp. 776–795, 1994.

[13] R. P. Skelton, T. Vilhelmsne, and G. A. Webster, "Energy criteria and cumulative damage during fatigue crack growth," *International Journal of Fatigue*, vol. 20, no. 9, pp. 641–649, 1998.

[14] P. K. Liaw, S. I. Kwun, and M. E. Fine, "Plastic work of fatigue propagation in steels and aluminum alloys," *Metallurgical Transactions A*, vol. 12A, pp. 49–55, 1981.

[15] M. E. Fine and D. L. Davidson, "Quantitative measurement of energy associated with moving a fatigue crack," *Fatigue Mechanisms: Advances in Quantitative Measurement of Physical Damage, ASTM STP 811*, pp. 350–370, 1983.

[16] N. Ranaganathan, J. Petit, and J. de Fouquet, "Energy required for fatigue crack propagation," in *Proceedings of the 7th International Conference on Strength of Metals and Alloys*, (Montreal Canada), pp. 1267–1272, 1986.

[17] N. Ranganathan, N. Jendoubi, M. Benguediab, and J. Petit, "Effect of R-ratio and ΔK level on the hysteretic energy dissipated during fatigue crack propagation," *Scripta Metallurgica*, vol. 21, pp. 1045–1049, 1987.

[18] Y. Birol, "What happens to the energy input during fatigue crack propagation," *Materials Science and Engineering*, vol. A104, pp. 117–124, 1988.

[19] H. P. Stuwe and R. Pippan, "On the energy balance of fatigue crack growth," *Computers and Structures*, vol. 44, no. 1-2, pp. 13–17, 1992.

[20] N. Ranganathan and J. R. Desforges, "Equivalent constant amplitude concepts examined under fatigue crack propagation by block loading," *Fatigue and Fracture of Engineering Materials and Structures*, vol. 19, no. 8, pp. 997–1008, 1996.

[21] N. Rajic, Y. C. Lam, and A. K. Wong, "Evolved heat as a fatigue characterization parameter," *Materials Science Forum*, vol. 210-213, pp. 463–470, 1996.

[22] N. Ranganathan, "Analysis of fatigue crack growth in terms of crack closure and energy," *Advances in Fatigue Crack Closure Measurement and Analysis: Second Volume, ASTM STP 1343*, pp. 14–38, 1999.

[23] J. Qian and A. Fatemi, "Mixed mode fatigue crack growth: a literature survey," *Engineering Fracture Mechanics*, vol. 55, no. 6, pp. 969–990, 1996.

[24] D. F. Socie, C. T. Hua, and D. W. Worthem, "Mixed mode small crack growth," *Fatigue and Fracture of Engineering Materials*, vol. 10, no. 1, pp. 1–16, 1987.

[25] J. P. Campbell and R. O. Ritchie, "Mixed-mode, high cycle fatigue crack growth thresholds in Ti – 6Al – 4V, I. A comparison of large and short crack behavior," *Engineering Fracture Mechanics*, vol. 67, pp. 209–227, 2000.

[26] J. P. Campbell and R. O. Ritchie, "Mixed mode, high cycle fatigue thresholds in Ti – 6Al – 4V, II. Quantification of crack tip shielding," *Engineering Fracture Mechanics*, vol. 67, pp. 229–249, 2000.

[27] A. K. Soh and L. C. Bian, "Mixed mode fatigue crack growth criteria," *International Journal of Fatigue*, vol. 23, pp. 427–439, 2001.

[28] D. A. Faulke and L. P. Pook, "A finite element analysis of three point bend crack path constraint specimens," *OMAE, Materials Engineering, ASME*, vol. 3, pp. 233–238, 1995.

[29] J. Qian and A. Fatemi, "Fatigue crack growth under mixed-mode I and II loading," *Fatigue and Fracture of Engineering Materials and Structures*, vol. 19, no. 10, pp. 1277–1284, 1996.

[30] S. B. Beiner, "Fatigue crack growth studies under mixed-mode loading," *International Journal of Fatigue*, vol. 23, pp. S259–S263, 2001.

[31] M. A. Magill and F. J. Zwerneman, "An analysis of sustained mixed mode fatigue crack growth," *Engineering Fracture Mechanics*, vol. 56, no. 1, pp. 9–24, 1997.

[32] N. W. Klingbeil and J. L. Beuth, "Interfacial fracture testing of deposited metal layers under four point bending," *Engineering Fracture Mechanics*, vol. 56, no. 1, pp. 113–126, 1997.

[33] N. W. Klingbeil and J. L. Beuth, "Continuous delamination of sprayed deposits via applied curvature," *International Journal of Mechanical Sciences*, vol. 40, no. 1, pp. 1–13, 1998.

[34] N. W. Klingbeil and J. L. Beuth, "On the design of debond-resistant bimaterials part I: Free-edge singularity approach," *Engineering Fracture Mechanics*, vol. 66, pp. 93–110, 2000.

[35] N. W. Klingbeil and J. L. Beuth, "On the design of debond-resistant bimaterials part II: A comparison of free-edge and interface crack approaches," *Engineering Fracture Mechanics*, vol. 66, pp. 111–128, 2000.

[36] D. Yao, Z. Zhang, and J. K. Shang, "An experimental technique for studying mixed-mode fatigue crack growth in solder joints," *Transactions of the ASME: Journal of Electronic Packaging*, vol. 118, pp. 45–48, 1996.

[37] D. Yao and J. K. Shang, "Effect of load-mix on fatigue crack growth in 63Sn – 37Pb solder joints," *Transactions of the ASME: Journal of Electronic Packaging*, vol. 119, pp. 114–118, 1997.

[38] X. X. Xu, A. D. Crocombe, and P. A. Smith, "Mixed mode fatigue and fracture behaviour of joints bonded with either filled or filled and toughened adhesive," *International Journal of Fatigue*, vol. 17, no. 4, pp. 279–286, 1995.

[39] H. Nayeb-Hashami and P. Yang, "Mixed mode I/II fracture and fatigue crack growth along 63Sn – 37Pb solder/brass interface," *International Journal of Fatigue*, vol. 23, pp. S325–S335, 2001.

[40] M. Okazaki, M. Okamoto, and Y. Harada, "Interfacial fatigue crack propagation in Ni-based superalloy protective coatings," *Fatigue and Fracture of Engineering Materials and Structures*, vol. 24, pp. 885–865, 2001.

[41] M. M. A. Wahab, I. A. Ashcroft, A. D. Crocombe, and P. A. Smith, "Numerical prediction of fatigue crack growth propagation lifetime in adhesively bonded structures," *International Journal of Fatigue*, vol. 24, pp. 705–709, 2002.

[42] F. Ellyin and J. Wu, "Elastic-plastic analysis of a stationary crack under cyclic loading and effect of overload," *International Journal of Fracture*, vol. 56, pp. 189–208, 1992.

[43] Z. Suo and J. W. Hutchinson, "Interface crack between two elastic layers," *International Journal of Fracture*, vol. 43, pp. 1–18, 1990.

[44] J. W. Hutchinson and Z. Suo, "Mixed mode cracking in layered materials," *Advances in Applied Mechanics*, vol. 29, pp. 63–191, 1992.

[45] T. L. Anderson, *Fracture Mechanics: Fundamentals and Applications*. CRC Press LLC, Boca Raton, 1995.

[46] G. E. Dieter, *Mechanical Metallurgy*. New York: McGraw-Hill Book Company, 1986.

[47] N. E. Dowling, *Mechanical Behavior of Materials*. Upper Saddle River, NJ 07458: Prentice Hall, 1993.

[48] P. C. Paris *et al.*, "Extensive study of low fatigue growth rates in A533 and A508 steels," *Stress Analysis and Growth of Cracks, Proceedings of the 1971 National Symposium on Fracture Machanics, Part I, ASTM STP 513,*, pp. 141–176, 1972.

[49] P. G. Charalambides, J. Lund, A. G. Evans, and R. M. McMeeking, "A test specimen for determining the fracture resistance of bimaterial interfaces," *Journal of Applied Mechanics*, vol. 56, pp. 77–82, 1989.

[50] J. Dundurs, "Discussion of 'edge-bonded dissimilar orthogonal wedges under normal and shear loading'," *Journal of Applied Mechanics*, vol. 36, pp. 650–652, 1969.

[51] Hibbitt, Karlsson, Sorensen, and Inc., *ABAQUS Theory Manual*, 1999.