

**NUMERICAL PREDICTIONS OF EVOLVING CRACK FRONT
GEOMETRY AND FATIGUE LIFE FROM COUNTERSUNK HOLES IN
THIN PLATES**

**PRÉDICTIONS NUMÉRIQUES DE L'ÉVOLUTION DE LA GÉOMÉTRIE
DES FISSURES ET LA DURÉE DE VIE EN FATIGUE DÉBUTANT DE
TROUS CHANFREINÉS DANS LES PLAQUES MINCES**

A Thesis Submitted

to the Division of Graduate Studies of the Royal Military College of Canada

By:

Lucas Alousis, B.Eng,

In Partial Fulfillment of the Requirements for the Degree of
Master of Applied Science in Aeronautical Engineering

October 2014

© This thesis may be used within the Department of National
Defence but copyright for open publication remains the property of the author.

ACKNOWLEDGEMENTS

First and foremost, I would like to extend my gratitude to my supervisor Dr. Diane Wowk for her invaluable support and guidance during the last two years. I am grateful to the Directorate of Technical Airworthiness and Engineering Support (DTAES) of the Department of National Defence (DND) for funding this work and to the wonderful people and the National Research Council Canada (NRC) who were always available to answer my questions and provide valuable information. My thanks are also extended to Dr. David DuQuesnay for providing me with advanced knowledge of materials and fatigue. I would also like to thank my fellow graduate students who played a large part in making the past two years an amazing experience. Of course, I would not have been able to accomplish this thesis without the support of my family and my girlfriend Courtney, who were always there for me whenever it was needed.

ABSTRACT

Alousis, Lucas, M.A.Sc. (Aeronautical Engineering). Royal Military College of Canada, October 2014.
Numerical Predictions of Evolving Crack Front Geometry and Fatigue Life from Countersunk Holes in Thin Plates. Thesis Supervisor: Dr. Diane Wowk.

The study of fatigue crack growth for damage tolerance analysis is of interest to the aerospace industry to enable accurate and economical estimates of a component's life before replacement or repairs are required. Many numerical analyses used for damage tolerance employ a series of assumptions that enable results to be generated quickly at the cost of accuracy. With advances in computational power and numerical analyses some of these assumptions, such as crack shape, are not necessary and not including the assumptions in the analysis result in more accurate fatigue predictions. This thesis presents the development and application of an automated crack growth program, called the ACGP, to predict crack growth and fatigue life in non-standard geometries, where the shape of the crack front is determined iteratively.

The ACGP was developed using a simulated growth technique and employed finite element analysis and Paris growth equations to predict fatigue crack growth of cracks in thin plates. The ACGP was validated for predictions of fatigue life and crack growth that were found to agree with experimental and accepted numerical results. The predictions were also compared to results generated using the fatigue crack growth program AFGROW and differences and similarities are presented. Models were also created to predict crack growth emanating from countersunk fastener holes in thin plates. Assessment and comparison of the results are presented, identifying the growth pattern, critical initial location and the effect of geometric properties on the fatigue life.

Keywords: Finite element analysis, fatigue, crack growth, countersunk fastener holes, spline

RÉSUMÉ

Alousis, Lucas, M. Sc. A. (Génie aéronautique). Collège militaire royal du Canada, Oct 2014. Prédiction numérique de l'Évolution de la Géométrie des Fissures et la Durée de vie en Fatigue Débutant de Trous Chanfreinés dans les Plaques Mince. Directeur de thèse: Dr. Diane Wowk

L'étude de la résistance, la durabilité et la tolérance aux dommages des matériaux est d'intérêt à la communauté aérospatiale pour améliorer la sécurité et estimer de la durée de vie de composantes en avance d'un remplacement ou d'une réparation. Plusieurs des méthodes numériques employées pour l'analyse de la tolérance aux dommages utilisent des hypothèses simplificatrices permettant un calcul rapide au dépend de la précision. Des progrès en informatique et en analyse numérique ont rendu certaines de ces simplifications redondantes, tel que la géométrie des fissures, et, en conséquence, les prédictions de la vie en fatigue sont devenues plus précises. Cette thèse discutera du développement et l'application d'un programme informatique capable de prédire la propagation des fissures et la résistance en fatigue pour les cas de géométries hors norme, où la géométrie de la fissure est déterminée de façon itérative. Le programme de prédiction automatisé de propagation de fissure, appelé ACGP, emploie une technique de simulation de propagation, une méthode des éléments finis et les équations de croissance de Paris pour prédire la propagation des fissures dans un plaque mince. Les prédictions de l'ACGP de la vie en fatigue et la propagation des fissures se comparent bien avec des résultats expérimentaux et numériques. Les résultats de l'ACGP ont été aussi validés contre ceux du programme informatique en analyse de fatigue AFGROW, et les similarités et différences sont présentées. De plus, la modélisation par éléments finis est employée pour prédire la propagation de fissures débutant des trous chanfreinés dans les plaques minces. L'évaluation et comparaison des résultats sont présentées, résultant en une identification du modèle de croissance, une détermination du point initial critique de la fissure et l'effet des propriétés géométriques sur la durée de vie de fatigue.

Mots-Clés: Analyse par éléments finis, fatigue, croissance des fissures, trous de fixations fraisées, spline

Table of Contents

List of Figures	vii
List of Tables	x
List of Symbols and Acronyms.....	xi
Chapter 1 Introduction.....	1
Chapter 2 Literature Review.....	3
2.1 Background.....	3
2.2 Numerical Predictions of Crack Shape Evolution and Fatigue Life.....	9
2.2.1 Computational Modelling of the Stress Field.....	10
2.2.2 Methods Used to Predict SIFs along the Crack Front.....	11
2.2.3 Growth Using the Paris Law.....	13
2.2.4 The Simulated Growth Technique.....	16
2.2.5 The Geometry of the Crack Front.....	19
2.2.6 Validation of the Numerical Method.....	23
Chapter 3 Description of the ACGP.....	25
3.1 Pre-Processing.....	28
3.2 Processing.....	28
3.2.1 Creation of the FEM Model.....	28
3.2.2 Extraction of the Stress Intensity Factors.....	37
3.2.3 Growth Model.....	39
3.3 Post-Processing.....	45
Chapter 4 Validation of the ACGP Program.....	47
4.1 Accuracy of the Stress Intensity Factor Predictions.....	47
4.1.1 Two Symmetric Corner Cracks Emanating from a Hole in a Plate.....	47
4.1.2 A Single Corner Crack Emanating from a Hole in a Plate.....	52
4.2 Accuracy of the Crack Growth Model.....	57
4.3 ACGP Comparison with AFGROW Predictions.....	61
4.4 ACGP Comparison with Experimental Results.....	70
4.5 Summary.....	80
Chapter 5 Crack Growth in Countersunk Fastener Holes.....	82
5.1 Comparison between the ACGP and AFGROW Predictions.....	85
5.2 Influence of Countersunk Geometry on Fatigue Life.....	93
5.2.1 Discretization Error.....	96

5.3	Effect of Initial Crack Location	98
5.3.1	Initial Corner Crack (Simulation 2)	98
5.3.2	Initial Knee Crack (Simulation 7)	101
5.3.3	Initial Upper Crack (Simulation 12)	103
5.3.4	Fatigue Life	105
5.4	Effects of Countersunk Geometry	107
5.4.1	The Countersunk Hole Radius	107
5.4.2	Plate Thickness	109
5.4.3	Countersink Angle	110
5.5	Discussion of the Countersunk Simulations	112
Chapter 6	Conclusions and Recommendations	114
6.1	Summary and Conclusions	114
6.2	Recommendations	116
References	118
Appendices	122
Appendix A	Initial Crack Front Geometries	123
Appendix B	– Visual Basic Code for the ACGP (Version 3)	130

List of Figures

Figure 2.1: Illustration of the stages of crack growth due to cyclic loading. E: applied loading leads to crack extension of increment da F: Loading leads to plastic deformation that blunts the crack tip G: upon unloading the crack the crack tip becomes sharp [2].	3
Figure 2.2: Crack formation a) mode 1 opening b) mode 2 in-plane shear c) mode 3 out of plane shear or tearing mode.	4
Figure 2.3: Cylindrical position coordinates for the stress field at a crack tip [3].	5
Figure 2.4: Three regions of fatigue growth 1) threshold 2) Paris 3) fracture [4].	6
Figure 2.5: Superposition method for a crack growing symmetrically from a loaded fastener hole [8].	7
Figure 2.6: Mesh configuration of a crack tip using the 1/4-point displacement method [14].	11
Figure 2.7: Meshing of a crack front using two rows of concentric elements around the crack tip.	12
Figure 2.8: Integration path with radius ρ used for the contour integral method.	13
Figure 2.9: Paris growth technique: a) initial crack front b) calculation of ΔK along the crack front c) new point projected normal to the tangent of the crack front d) definition of the new crack front e) new crack front [19].	14
Figure 2.10: Fatigue growth calculation models: a) two points along the crack front assuming an elliptical shape, b) several points along the crack front using a spline to represent the shape [14].	17
Figure 2.11: Ellipse fit (blue) compared to a spline fit for eight points along a crack front.	17
Figure 2.12: Mesh along the crack front for a corner crack with a) the crack front fit to a spline with many elements b) the crack front fit to an ellipse with fewer elements.	18
Figure 2.13: Beta factor distribution along the crack front of a corner crack, showing a drop in magnitude near the free surfaces.	19
Figure 2.14: Cracks of different initial shape developing into a similar shape a) high aspect ratio b) aspect ratio >1 c) aspect ratio=1 [28].	20
Figure 2.15: Crack with an irregular initial shape quickly becoming "smooth" and developing into a preferred shape [28].	20
Figure 2.16: Geometry and loading found in three-row countersunk riveted lap joints [33].	21
Figure 2.17: Location of a corner crack and a knee crack at a countersunk hole.	22
Figure 2.18: Growth pattern for cracks emanating from countersunk holes for a) a corner crack b) a knee crack. The surface locations of fastest and slowest growth are shown in red and blue respectively.	23
Figure 3.1: Possible plate geometries with applied loading and initial crack locations that can be modeled a) corner crack in a plate with a hole b) corner crack in a plate with a notch c) corner, knee and upper crack in a plate with a countersunk fastener hole.	26
Figure 3.2: Steps of the automated crack growth program (ACGP), pre-processing shown in yellow, processing shown in blue and post-processing shown in red. i refers to the loops of the system.	27
Figure 3.3: Version 3, Excel user interface.	28
Figure 3.4: Screenshot of a plate with a hole in StressCheck showing the mesh, loads (arrows) and constraints (anchors).	29
Figure 3.5: Co-ordinate system used to define the crack front a) if the feature is located in the middle of the plate b) if the feature is located at the edge of a plate.	30
Figure 3.6: Example of x and y co-ordinate input for a corner crack front emanating from a hole with a radius of 0.3695mm.	31
Figure 3.7: Order of points to be input during pre-processing for a) a corner crack b) a knee crack c) an upper crack.	31

Figure 3.8: A spline fit to the crack front enables complex crack shapes to be modeled. Only the mesh on the crack face is shown.	32
Figure 3.9: Possible <i>int</i> values for a) a corner crack with the shortest crack length along the straight shank b) a corner crack with the shortest crack length along the lower surface c) a knee crack d) an upper crack.	33
Figure 3.10: Typical crack tip mesh.....	34
Figure 3.11: Effect of changing the meshing parameters on the crack front and the body.....	36
Figure 3.12: Version 2-5 extraction along the crack front a) entire crack front illustrating more points in the first and last 10% of the crack front b) close up at the last 10% of the crack front.	38
Figure 3.13: Location of the radius of integration at the crack tip.....	39
Figure 3.14: The crack growth process a) location of the highest ΔK value b) projected points along the crack front c) new crack front fit with a spline.	40
Figure 3.15: Radius of integration at point 2 falling outside of the model domain.	40
Figure 3.16: An extraction path (red lines) at the free surface falling outside the model domain.	41
Figure 3.17: Projection of a new point normal to the tangent of a point along the crack front.	43
Figure 3.18: Stages of crack growth.	43
Figure 3.19: Linear extrapolation of the last point at the intersection of the crack and the feature.....	44
Figure 3.20: Version 1 post-processing interface.	45
Figure 3.21: <i>a</i> and <i>c</i> values gathered for a) a corner crack b) a knee crack and c) a upper crack.....	46
Figure 4.1: Geometry, material and applied stress used to generate SIF results from Bakuckas [13].....	48
Figure 4.2: Model, geometry and meshing parameters for two symmetric corner cracks emanating from a hole in a plate. The second geometry is shown in brackets.	50
Figure 4.3: Parametric angle of a corner crack for two different aspect ratios.	51
Figure 4.4: Geometric factors vs. normalized angle along the crack face for two symmetric corner cracks emanating from a hole in a plate.....	52
Figure 4.5: Geometry, material and applied stress used to generate SIF results from Fawaz and Andersson [13].....	53
Figure 4.6: Mesh of the shallow corner crack at the intersection of the lower surface.....	54
Figure 4.7: Model, geometry and meshing parameters for a corner crack emanating from a hole in a plate.	55
Figure 4.8: Beta factor results for a corner crack emanating from a hole in a plate.	56
Figure 4.9: Model and geometry parameters for a corner crack emanating from a notch in a plate used in AFGROW.	58
Figure 4.10: Process used to generate growth results from the ACGP.	59
Figure 4.11: Crack length <i>a</i> vs. crack length <i>c</i> predicted from the ACGP and AFGROW.....	60
Figure 4.12: Crack growth vs. number of cycles comparing the ACGP to AFGROW.	61
Figure 4.13: Model, geometry and meshing parameters for a corner crack emanating from a notch in a plate.....	63
Figure 4.14: <i>a</i> vs. <i>c</i> comparing the ACGP to AFGROW.....	64
Figure 4.15: Aspect ratio vs. <i>a</i> comparing the ACGP to AFGROW.	64
Figure 4.16: Crack growth vs. number of cycles comparing the ACGP and AFGROW.....	65
Figure 4.17: Max ΔK vs. length along the surface comparing the thesis program to AFGROW.	66
Figure 4.18: Crack growth vs. number of cycles comparing the ACGP (ΔK increased by 5%) and AFGROW.	66

Figure 4.19: Geometry comparison between a spline and ellipse fit	67
Figure 4.20: ΔK vs. crack front angle for four different crack lengths. ΔK values using an ellipse fit are shown in red and using a spline fit are shown in blue	68
Figure 4.21: Marker band profile from experimental results [43].	71
Figure 4.22: Model, geometry and meshing parameters for a corner crack emanating from a notch in a plate used for the ACGP simulation.	72
Figure 4.23: Crack shape at different points of propagation.....	73
Figure 4.24: Growth along the surface vs. number of cycles comparing the thesis program to experimental results.	74
Figure 4.25: da/dN vs. ΔK used to generate Paris parameters.	76
Figure 4.26: Aspect ratio vs. length along the notch for Paris tests.....	77
Figure 4.27: Length along the notch vs. length along the surface for Paris tests.....	77
Figure 4.28: Length along the surface vs. number of cycles comparing results generated for a full test and during post-processing	78
Figure 4.29: Length along the surface vs. number of cycles comparing the Paris tests to experimental results.	79
Figure 5.1: Geometry of a countersunk fastener used to hold two sheets together.....	82
Figure 5.2: Three initial crack locations: corner, knee and upper.....	83
Figure 5.3: Three stages of crack growth in countersunk holes defined as type 1, type 2 and type 3.	84
Figure 5.4: Direction of the normalized angles calculated for a corner, knee and upper crack in a countersunk hole.	84
Figure 5.5: AFGROW countersunk geometry and interface.	85
Figure 5.6: Model, geometric and meshing parameters for the ACGP comparison with AFGROW.....	87
Figure 5.7: Aspect ratio vs. growth along the lower surface comparing the ACGP to AFGROW for a crack emanating from a countersunk hole.	88
Figure 5.8: Crack shape comparison for different stages of growth from the ACGP and AFGROW predictions.....	89
Figure 5.9: Mesh of a crack before transition a) crack front viewed from inside the plate b) crack tip near the intersection of the straight shank and countersink.	90
Figure 5.10: a and c growth vs. number of cycles comparing the ACGP and AFGROW for a crack emanating from a countersunk hole.....	91
Figure 5.11: ΔK vs. normalized angle for shape 2 spline and ellipse.....	92
Figure 5.12: ΔK vs. normalized angle for shape 3 spline and ellipse.....	92
Figure 5.13: Baseline model, geometric and meshing parameters used for countersunk tests.	94
Figure 5.14: Baseline mesh for the countersunk tests for an initial a) corner crack b) knee crack c) upper crack.....	95
Figure 5.15: P-level test comparing aspect ratio vs. c growth.	97
Figure 5.16: P-level test comparing c growth vs. fatigue life.....	97
Figure 5.17: Six predicted crack shapes of a corner crack in a countersunk hole.....	99
Figure 5.18: Beta vs. normalized angle for six crack shapes with initial corner crack.....	100
Figure 5.19: Six predicted crack shapes for a knee crack in a countersunk hole.....	101
Figure 5.20: Beta vs. normalized angle for six crack shapes with initial knee crack.	102
Figure 5.21: Six predicted crack shapes for an upper crack in a countersunk hole.	103
Figure 5.22: Beta vs. normalized angle for six crack shapes with initial upper crack.	104

Figure 5.23: Illustrating the location of c for three initial crack locations.....	105
Figure 5.24: c growth vs. the number of cycles for three different initial crack locations. The simulated cracks were located in plates made of 2024-T3 Alclad aluminum undergoing cyclic loading of 100MPa with a stress ratio of 0.	106
Figure 5.25: Relative Von Mises stress levels for simulation 2 where red shows higher stresses and blue shows lower stresses.	107
Figure 5.26: Three different countersunk radius geometries.	108
Figure 5.27: c growth vs. the number of cycles for three different countersunk radii.....	108
Figure 5.28: Three different countersunk plate thicknesses.....	109
Figure 5.29: c growth vs. the number of cycles for three different plate thicknesses.....	110
Figure 5.30: Three different countersink angles.	111
Figure 5.31: c growth vs. the number of cycles for three different countersink angles.	111
Figure A.1 Location where the initial crack geometry is input into the ACGP.....	123
Figure A.2: Order of points along the crack front to be input during pre-processing for a) a corner crack b) a knee crack c) an upper crack.....	123

List of Tables

Table 4.1: Comparison of maximum ΔK values for four sets of geometry.....	69
Table 4.2: Paris parameters for three tests.	76
Table 5.1: Simulations for cracks emanating from countersunk holes in a plate.....	96
Table A.1: Initial geometry used for the simulation from Section 4.1.1 Two symmetric Corner Cracks Emanating from a Hole in a Plate	124
Table A.2: Initial geometry used for the simulation from Section 4.1.2 A single Corner Crack Emanating from a Hole in a Plate	124
Table A.3: Initial geometry used for the simulation from Sections 4.2.....	125
Table A.4: Initial geometry used for the simulation from Section 4.4 Comparison with Experimental ..	126
Table A.5: Initial geometry used for the simulations from Chapter 5 for simulations 1, 2 and 3.....	126
Table A.6: Initial geometry used for the simulation from Chapter 5 for simulation 4	126
Table A.7: Initial geometry used for the simulations from Chapter 5 for simulations 5, 7 and 9.....	127
Table A.8: Initial geometry used for the simulation from Chapter 5 for simulation 6	127
Table A.9: Initial geometry used for the simulation from Chapter 5 for simulation 8	127
Table A.10: Initial geometry used for the simulation from Chapter 5 for simulation 10	128
Table A.11: Initial geometry used for the simulation from Chapter 5 for simulation 11	128
Table A.12: Initial geometry used for the simulation from Chapter 5 for simulation 12	129
Table A.13: Initial geometry used for the simulation from Chapter 5 for simulation 13	129

List of Symbols and Acronyms

List of Acronyms

ACGP	Automated Crack Growth Program
AFGROW	Air Force Growth (Software)
BEM	Boundary Element Method
CIM	Contour Integral Method
EPFM	Elastic Plastic Fracture Mechanics
FADD3D	Fracture Analysis by Distributed Dislocations in 3Dimensions
FEAM	Finite Element Alternating Method
FEM	Finite Element Method
FRANC3D	Fracture Analysis Code in 3-Dimensions
HLT	High Load Transfer
LEFM	Linear Elastic Fracture Mechanics
LLT	Low Load Transfer
NRC	National Research Council Canada
SIF	Stress Intensity Factor
WFM	Weight Function Method

List of Symbols

a	Crack length
C	Paris constant
C_s	Countersunk depth
da	Change in crack length
dN	Change in the number of load cycles
E	Modulus of elasticity
h	Length
IR	1 st row of elements
K	Stress intensity factor
m	Paris exponent
OR	2 nd row of elements
Q	Elliptical shape factor
q	Scaling factor
R	Radius
R_s	Stress ratio
r	Bakuckas radius
S_t	Bakuckas applied stress
t	Depth
w	Width
x	Horizontal coordinate
y	Vertical coordinate
$Y(a)$	Geometry factor

List of Symbols (Greek)

α	Crack closure stress ratio
β	Beta factor
γ	Countersink angle

θ	Angle
ν	Poisson ratio
ρ	Radius of integration
σ_{ij}	Magnitude of the stress field
σ	Applied stress

List of Subscripts

I	Mode 1
i	Location number
max	Maximum
min	Minimum
op	Opening
L	Loading
IPR	Radius of integration
R	Residual stress

Chapter 1 Introduction

The study of failure due to fatigue can be analysed using fracture mechanics, which examines the influence of loading, crack shape and structural geometry on the fracture resistance of materials [1]. One situation that enforces the importance of studying failure due to fatigue in the aerospace industry was the catastrophic failure of the de Havilland Comet. The de Havilland Comet flew to altitudes higher than its competition and the expansion and contraction of the fuselage during this more severe flight cycle combined with the high stress concentrations at the corners of the square windows caused failure to occur due to fatigue. The initiation sites for fatigue in structures usually occur in areas where stress concentrations exist. In a complex structure, such as an aircraft, stress concentrations are present at geometric design details such as fillets, rounds, bolts and rivet holes. There are typically many initiation sites, and cracks propagate at different rates depending on geometry, loading and material. Crack propagation in a variety of complex configurations must be able to be predicted to ensure safety of the structure.

Nowadays the damage tolerance philosophy is used in the aircraft industry to prevent structural failure. This philosophy implies that cracks exist and structural failure will not occur before the cracks can be detected. Damage tolerance utilizes the principles of fracture mechanics to predict how quickly cracks will grow, the largest crack size the structure can withstand before failure, and the detection and repair/replace intervals of the structures. Damage tolerance requires that the tools used to predict the rate of crack growth and the residual strength of a component are reliable, and can be adopted for complex crack configurations. When predicting the rate of crack growth in a component with complex geometry or loading, the path and shape of the crack are often unknown. It is typically assumed that the crack front takes on a simplified shape such as an ellipse, but this can result in inaccurate predictions of fatigue life.

A program was developed for this thesis that uses numerical analysis to predict fatigue life, but does not assume crack shape or path and that can be employed for damage tolerance analysis. The

automated crack growth program (ACGP) was developed from an existing preliminary code developed at the Royal Military College of Canada to predicted growth with a crack front fit to an ellipse. The code was altered so that incremental growth of a crack in a thin plate could be automatically predicted and the crack front could be represented by a spline. This allowed the crack shape to grow according to the local stresses predicted along the crack front. From information gathered from appropriate literature, the program was optimised to automatically calculate stress intensity factors along crack fronts using finite element analysis. Excel and Visual Basic were used to project new crack fronts without assuming a set crack shape or growth pattern. The goal of this thesis was to validate the fatigue life and crack growth predictions of the ACGP using experimental results, validate the SIF predictions using numerical results and to provide insight into the growth of mode 1 cracks from countersunk fastener holes in a plate.

This thesis is divided into seven chapters outlined below.

- Chapter 1 explains the motivation and goals for this research.
- Chapter 2 provides background applicable to this thesis, noting relevant studies from the literature.
- Chapter 3 details how the ACGP works and its unique features.
- Chapter 4 examines five case studies used to validate the accuracy of the ACGP.
- Chapter 5 examines the predicted growth of cracks emanating from countersunk holes in a plate.
- Chapter 6 provides a summary and concluding remarks on the significance of the work completed in this thesis, as well as recommendations to expand the accuracy and capabilities of the ACGP.

Chapter 2 Literature Review

2.1 Background

Cracks due to fatigue are present in many metallic components, propagating at different rates depending on the geometry of the structure, loading and material. In areas where cracks are found, the applied load typically causes tension in the structure. This stretching exploits the weakest path in the material, usually a fault or impurity, resulting in crack formation or growth. During loading the crack grows by a length da and during unloading the crack tip is blunted and growth stops. The crack growth rate is defined as the change in crack length per number of incremental load cycles (da/dN). Fatigue crack growth due to cyclic loading is shown in Figure 2.1.

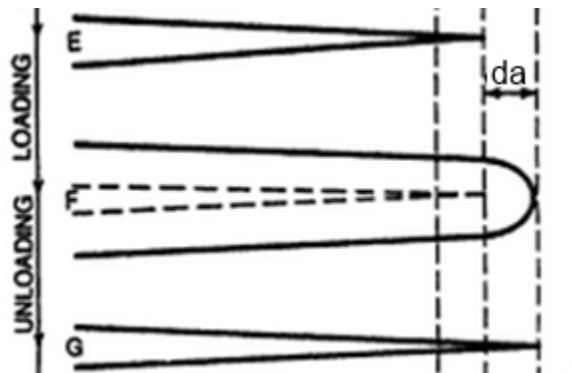


Figure 2.1: Illustration of the stages of crack growth due to cyclic loading. E: applied loading leads to crack extension of increment da F: Loading leads to plastic deformation that blunts the crack tip G: upon unloading the crack the crack tip becomes sharp [2].

Cracks are formed in one or a combination of three possible modes depending on the overall type of loading and deformation mode. Mode 1 is the opening mode and is due to tensile loading, mode 2 is the in-plane shear mode resulting from in-plane shear loading and mode 3 is the out-of-plane shear mode or tearing mode which results from an out of plane shear load. The three modes are illustrated in Figure 2.2. This thesis will focus on mode 1 cracks in plates subject to uniform tensile loading.

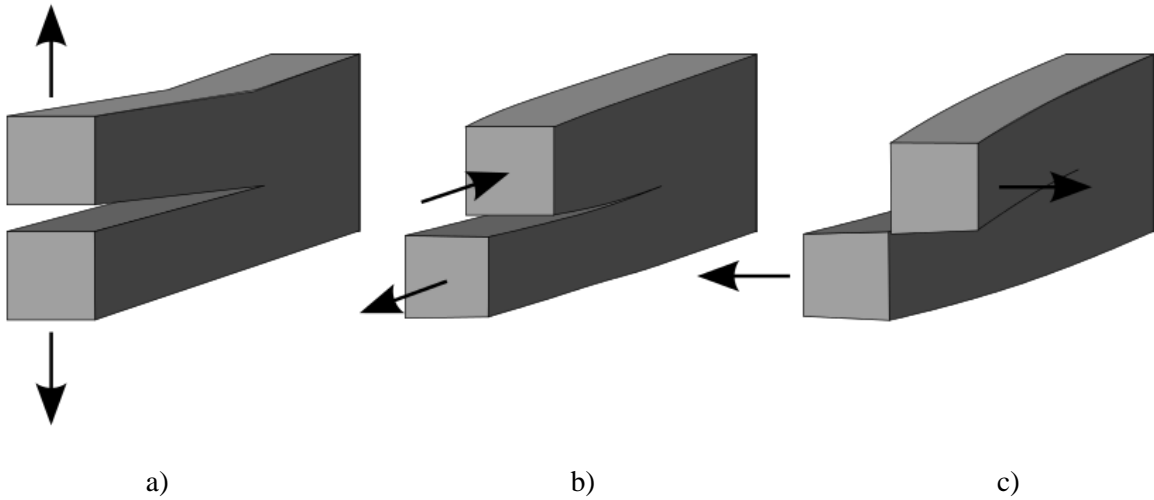


Figure 2.2: Crack formation a) mode 1 opening b) mode 2 in-plane shear c) mode 3 out of plane shear or tearing mode.

Fracture mechanics is based on the analysis of the stress distribution near the tip of a crack. Crack growth in different materials can be analysed using two different regimes of fracture mechanics: linear elastic fracture mechanics (LEFM) or elastic plastic fracture mechanics (EPFM). LEFM assumes the far field strain is predominantly elastic and the region of plasticity around the crack tip is small. EPFM is used where far field strain is not elastic. The present study will use LEFM as it requires much simpler finite element models than EPFM.

In LEFM the magnitude of the stress field (σ_{ij}) at a given location is a function of a parameter called K , the stress intensity factor (SIF), which is proportional to the applied stress, given by Equation (1).

$$\sigma_{ij} = \frac{K}{\sqrt{2\pi r}} * f_{ij}(\theta) \quad (1)$$

where r and θ are cylindrical position coordinates as seen in Figure 2.3 and $f_{ij}(\theta)$ is a function to determine the location of the principle or shearing stresses.

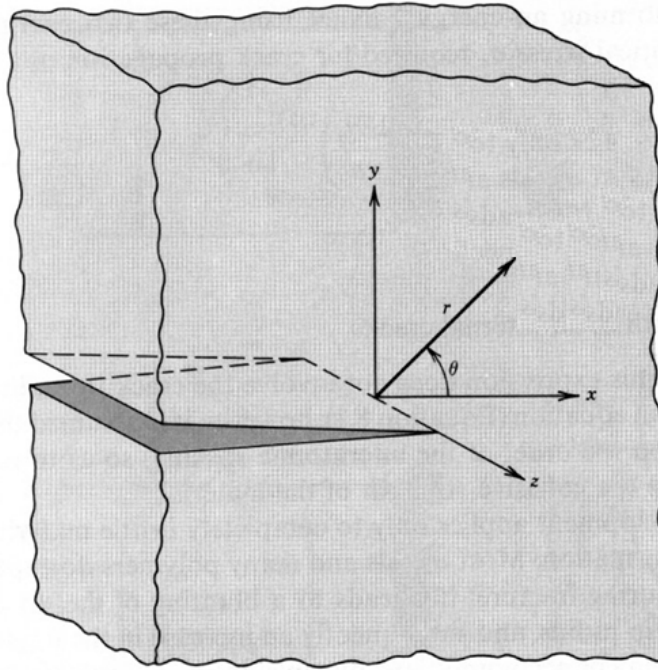


Figure 2.3: Cylindrical position coordinates for the stress field at a crack tip [3].

The stress intensity factor, K , is a function of the applied stress (σ), crack length (a) and a geometry factor ($Y(a)$):

$$K = Y(a)\sigma\sqrt{\pi a} \quad (2)$$

The geometry factor may also be called a beta factor and represented by β . Beta factors are traditionally used when performing fatigue calculations in aircraft.

Stress intensity factors can be related to the crack growth rate (da/dN) through the Paris equation using the change in SIF (ΔK) from the minimum and maximum loading,

$$\frac{da}{dN} = C\Delta K^m \quad (3)$$

where C is the Paris constant, m is the Paris exponent and ΔK is the stress intensity factor range observed during a loading cycle. From the Paris equation it can be inferred that the higher the ΔK the faster a crack will grow.

There are three regions characteristic of a typical fatigue growth curve: the threshold region where the crack first initiates and begins to grow, the linear region of the $\log da/dN$ versus $\log \Delta K$ plot also known as the Paris region where the crack growth is stable, and the unstable region where crack growth is accelerated and fracture occurs. The three regions can be seen in Figure 2.4 on a log-log scale of da/dN versus ΔK . The C and m values used in Equation (3) are determined by fitting a curve to the stable linear region to determine the slope (value of m) and intersection (value of C).

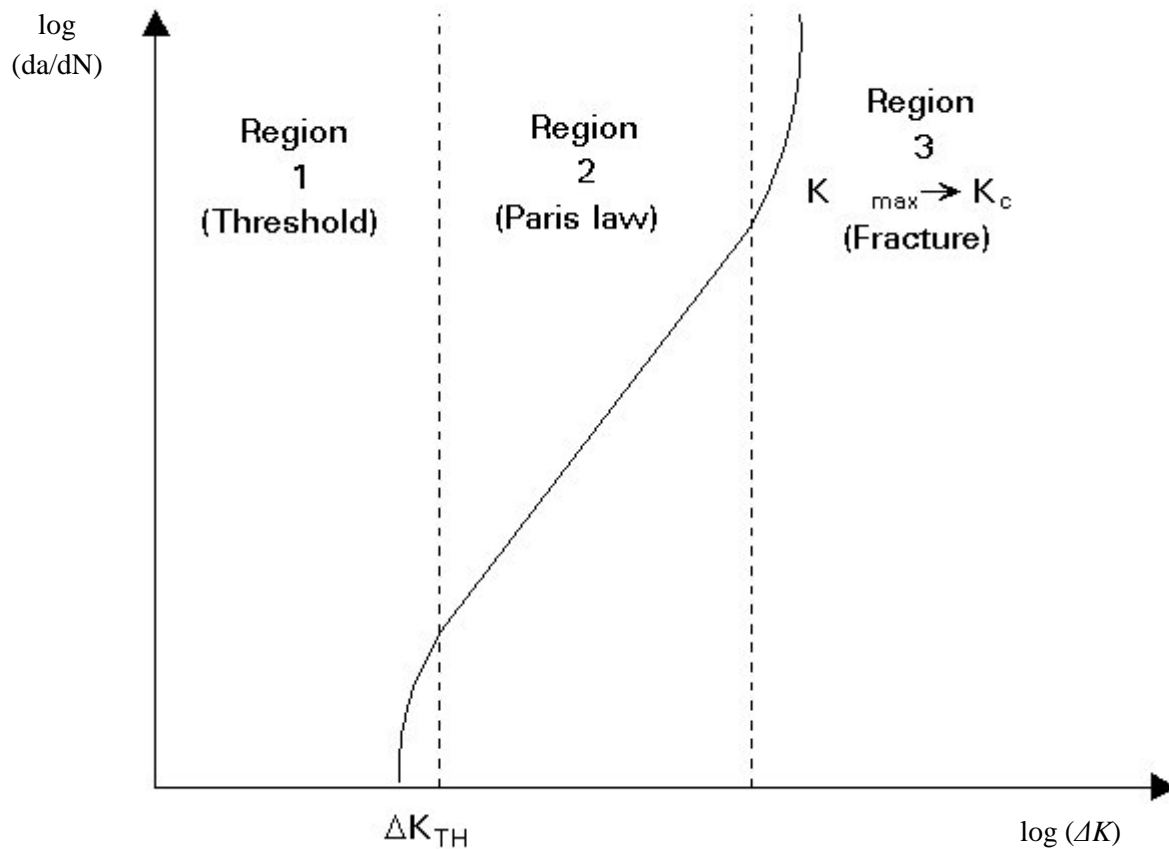


Figure 2.4: Three regions of fatigue growth 1) threshold 2) Paris 3) fracture [4].

Nowadays the damage tolerance philosophy is used in the aircraft industry. The damage tolerance design philosophy for metallic structures is characterized by the following [5]:

- The initial flaws in the structure are assumed to be cracks inherent in the material or induced during the manufacturing process (e.g. tool drop). For a given structure the cracks are assumed to

be in the most critical location. The initial size is determined by inspection or by specifications based on historical data as documented in damage tolerance handbooks.

- Crack growth is stable.
- The assumed initial flaw does not affect static strength considerably.
- The goal is to demonstrate that the predicted half life of a crack is equal to the usage life, where an inspection interval is one usage life. If not, maintenance is performed.

Damage tolerance analysis utilizes SIFs and Paris growth data to determine the expected remaining life in a component and the crack growth rate so that inspections and repairs of the component can be scheduled safely and economically. The handbook method is one method used to determine SIFs using equations for simple configurations and can only be applied for LEFM. The equations can be found in compendiums such as those from Rooke and Cartwright [6] and Murakami [7] who determined the SIFs through experimental testing or computational analysis. For more complex geometry or loading, the SIFs found for the simple configurations can be superimposed or compounded together to generate the SIF for the complex geometry. These SIF's can then be used in LEFM analyses. An example of the superposition method is shown in Figure 2.5 for a crack growing symmetrically from a loaded fastener hole.

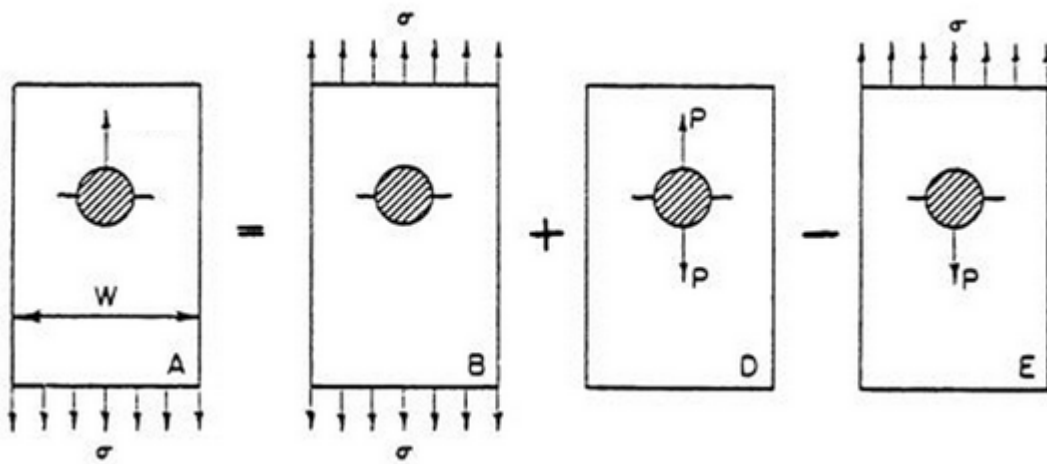


Figure 2.5: Superposition method for a crack growing symmetrically from a loaded fastener hole [8].

The SIF for the configuration can be found by summing the SIFs for each individual configuration using

$$K_{IA} = K_{IB} + K_{ID} - K_{IE} \quad (4)$$

Equation (4) can also be written in terms of beta factors, and since $\beta_{IA} = \beta_{IE}$,

$$\beta_{IA} = \frac{\beta_{IB} + \beta_{ID}}{2} \quad (5)$$

An edge crack approaching a fastener hole that cuts across a stringer is an example where the compounding method is applied. The beta factor for this complex geometry can be determined using

$$\beta_{total} = q\beta_a\beta_b\beta_c \quad (6)$$

where q is a scaling factor used to match experimental or computational results, β_a is the beta factor for an edge crack, β_b is the beta factor for a crack approaching a hole and β_c is the beta factor for a crack across a stringer.

Computational methods, such as the finite element method (FEM) can also be used to determine SIFs. These methods enable SIFs to be extracted through analysis of a modeled component without the need for superposition or compounding. With these methods, a complex 3D geometry can be analyzed to provide SIFs at multiple locations along the crack front. The handbook methods provide a fast and easy way to generate SIF solutions but the downfall of this method is that the solutions are limited to standard configurations where crack shape and path are known. In components with more complex geometries, neither the crack shape nor the path are known and these methods cannot be used with confidence. Computational analysis can provide crack growth and fatigue life solutions for more complex cases with a higher degree of accuracy but generating the results are more time consuming and require a high degree of knowledge from the user. There are a variety of computational methods that have been successfully employed to predict fatigue life and crack growth, which are presented in the following section.

2.2 Numerical Predictions of Crack Shape Evolution and Fatigue Life

One method that has been successfully utilized for predicting crack shape evolution and fatigue life in complex crack configurations is a numerical approach that generates successive crack fronts through an iterative process as first presented by Lin and Smith [9-11]. This method can be broken down into three stages:

1. A computational model is created that considers geometry, initial crack shape, material properties and loading.
2. The computational model is solved to predict stress intensity factors (SIFs) along the crack front.
3. New crack geometry is predicted using the Paris growth law to advance points along the crack front.

The following sections will present the different approaches to this method, broken down into six sections:

2.2.1 Computational modelling of the stress field

2.2.2 Methods used to predict SIFs along the crack front

2.2.3 Growth using the Paris Law: magnitude of crack growth increment at points along the crack front

2.2.4 The Simulated Growth Technique: projecting the crack increment to create a new crack front

2.2.5 The Geometry of the Crack Front: what patterns of growth have been previously observed

2.2.6 Validation of numerical predictions

2.2.1 Computational Modelling of the Stress Field

Many different computational methods can be utilized to determine the stress field from which stress intensity factors along the crack front are derived. These methods typically require the user to input the geometry of the model, the shape, size and location of the crack, the loads involved and the material properties. The finite element method (FEM) is a commonly used method for predicting SIFs. The FEM represents the geometry of interest with a set of elements whose behaviour is defined by polynomial functions known as shape functions. The shape functions are used to produce an approximate solution of the stress distribution in the component and SIFs can be generated from this solution. This method models the entire geometry including the crack itself and can be very computationally demanding depending on the density of the required mesh. The FEM has been used to generate very accurate predictions of SIFs along the crack front [9, 10] for cracks emanating from holes in a plate. One adaptation of the FEM is the finite element alternating method (FEAM) that was utilized by Nishioka and Atluri [11] to estimate SIFs. This method alternates between a finite element analysis of an uncracked body and an analytical solution of a crack under traction forces in an infinite body. FEAM is a less computationally demanding technique because only the uncracked body is modelled to calculate the stress field in the body. The stress field in the vicinity of the crack is then used in an analytic solution to calculate SIFs along the crack front.

The weight function method (WFM) is another method that has been employed to generate predictions of SIFs [12]. This method converts a 3D body into small slices through the width and thickness of a plate and generates SIF solutions from the forces acting on the crack surface due to loading and the restraint effect due to the uncracked regions of the body. The WFM creates a simplistic model that is not as computationally demanding as FEM. Another method that can be utilized is the 3D boundary element method (BEM). This method only models the surface of the body and can therefore be built quickly. BEM has been implemented in two programs: Fracture Analysis Code in 3-Dimensions (FRANC3D) and Fracture Analysis by Distributed Dislocations in

3Dimensions (FADD3D). This method has been utilized in a variety of fracture problems to provide accurate SIF solutions for relatively coarse meshes [13]. Each computational method has the ability to provide accurate estimations of SIFs as shown by Bakuckas [13]. Bakuckas performed an extensive study to compare the SIF solutions estimated from each of the numerical methods mentioned above. The study compiled results for corner cracks emanating from a hole in a plate and the results for each method were found to be within $\pm 3\%$ of each other. The FEM is utilized in this thesis because of the accessibility to commercial codes that can predict SIFs. Using common codes allows those involved in aircraft maintenance to perform analyses in house.

2.2.2 Methods Used to Predict SIFs along the Crack Front

Various methods can be employed to predict SIFs along the crack front. These methods utilize the mesh modeled around the crack tip to derive SIF results from the forces and displacements experienced along the crack front. One method that is commonly used is the $\frac{1}{4}$ -point displacement method [14] where element nodes are positioned at the $\frac{1}{4}$ -point away from the crack tip, as shown in Figure 2.6, and the displacements at these points are used to estimate the SIFs.

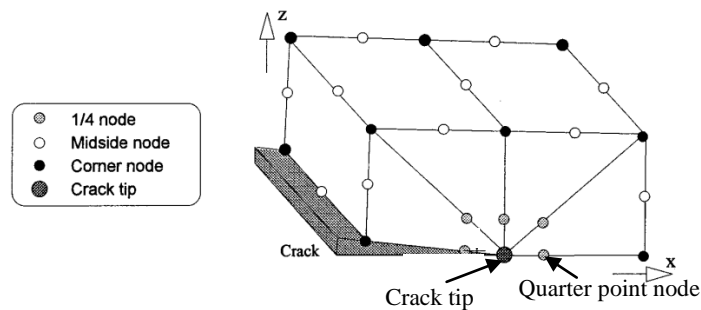


Figure 2.6: Mesh configuration of a crack tip using the 1/4-point displacement method [14].

Another method that can be utilized to calculate SIFs is the contour integral method (CIM) which allows the SIF extractions to occur anywhere along the crack front, at locations independent of the nodal location. For this method a series of concentric elements around the crack tip are generated with the diameter of the elements decreasing as they become closer to the crack tip [10, 15], shown in Figure 2.7.

The advantage of using a concentric series of elements is that the elements help to distribute the numerical errors associated with the crack tip singularity [16].

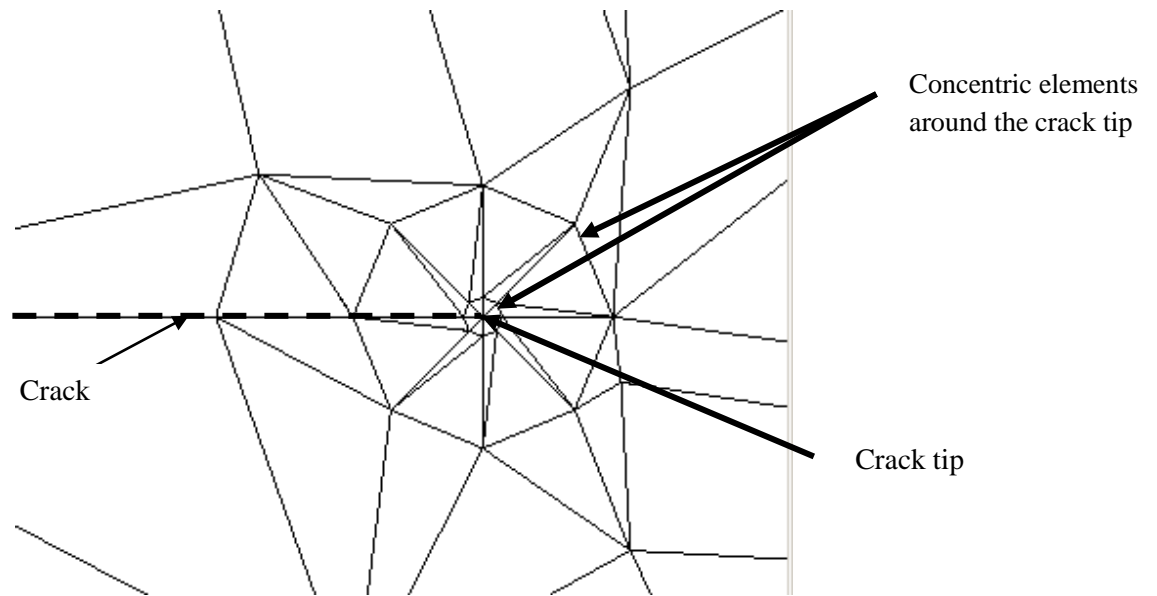


Figure 2.7: Meshing of a crack front using two rows of concentric elements around the crack tip.

With a proper mesh around the crack front the SIFs are estimated using the stresses and displacements calculated from integration along a circular path of radius ρ centred on the crack tip, shown in Figure 2.8.

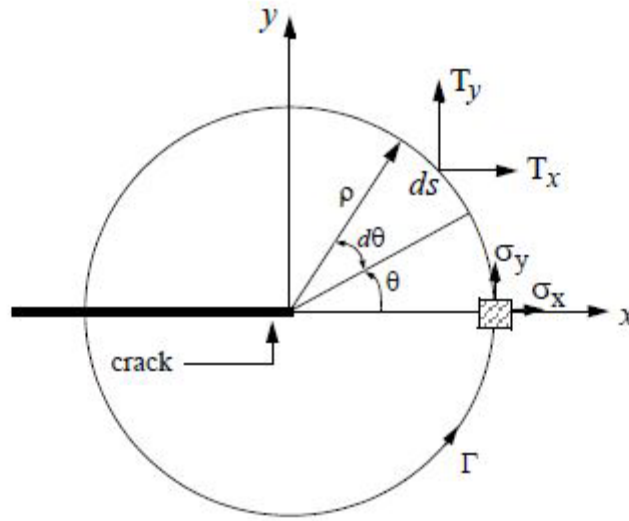


Figure 2.8: Integration path with radius ρ used for the contour integral method.

The mode 1 stress intensity factors (K_I) are then estimated using

$$K_I = \sqrt{2\pi}A_1 \quad (7)$$

where A_1 is the solution generated from the integration around the crack tip. The contour integral method is utilized in the commercial software StressCheck, employed in this thesis. When utilizing the StressCheck CIM, the key to accurately determining SIFs is to extract the stress results from the second row of elements around the crack tip because the first row of elements contains the crack tip where a stress singularity occurs [17]. It is recommended that radius of the second row of elements be 0.15 times the length of the smallest crack length [17].

2.2.3 Growth Using the Paris Law

Once stress intensity factors have been predicted along the crack front, growth can be simulated by advancing the crack at discrete points along the crack front using the Paris law. The advanced points are used to create a new crack front geometry. This method has been commonly employed in the past for advancement at two points along the crack front [18] and for advancement at seven to nine points along the crack front [9- 11, 18, 21, 22].

To determine the growth increment Δa at a point i along the crack front a derived equation from the Paris law is used [14]

$$\Delta a_i = \left(\frac{\Delta K_i}{\Delta K_{max}} \right)^m \Delta a_{max} \quad (8)$$

where ΔK_{max} is the largest ΔK value calculated along the crack front. At the point along the crack front where ΔK_{max} occurs, the growth increment Δa will be equal to the maximum growth increment Δa_{max} .

The ΔK values at points along the crack front (ΔK_i) are calculated using Equation (9).

$$\Delta K_i = K_i(1 - Rs) \quad (9)$$

where Rs is the ratio of the minimum applied stress and the maximum applied stress. This ratio is called the stress ratio and can be calculated using Equation (10).

$$Rs = \frac{\sigma_{min}}{\sigma_{max}} \quad (10)$$

If the stress ratio is 0 then ΔK_i is equal to K_i .

Using these equations the growth increment at every point can be calculated where a SIF was extracted. The growth increment at a point is projected normal to the tangent of the crack front at that point and the advanced points are used to define the new crack front as seen in the process illustrated in Figure 2.9.

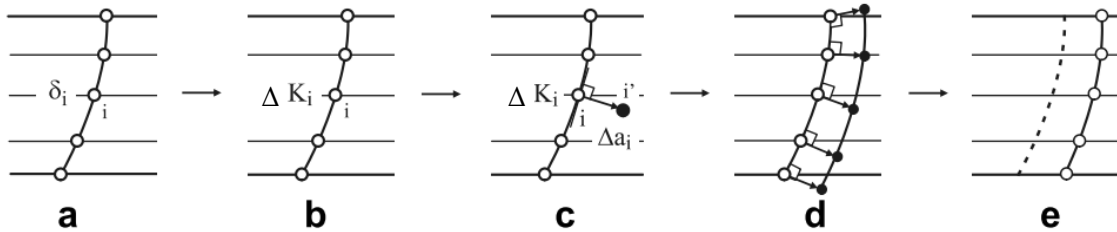


Figure 2.9: Paris growth technique: a) initial crack front b) calculation of ΔK along the crack front c) new point projected normal to the tangent of the crack front d) definition of the new crack front e) new crack front [19].

To account for other phenomena such as crack closure or residual stresses, both of which have been shown to affect fatigue predictions, refinements can be made to the model or to the calculated SIFs. The crack closure phenomenon was discovered by Elber [20] and is observed when the crack tip does not separate when a load is applied. An opening load is required before the crack tip separates. This has an effect on crack growth because the load before opening does not contribute to crack extension. Park et al. [21] accounted for crack closure by altering the ΔK values to account for the opening load using

$$\text{if } K_{op} < K_{min}, \Delta K = K_{max} - K_{min} = K_{max}(1 - Rs) \quad (11)$$

$$\text{if } K_{op} > K_{min}, \Delta K = K_{max} - K_{op} = K_{max}(1 - \alpha) \quad (12)$$

$$\alpha = \frac{K_{op}}{K_{max}} \quad (13)$$

where K_{op} is the maximum SIF calculated along the crack front when the opening load is applied and α is the opening stress ratio. The K_{op} value was determined by Ray [22] using the same material, loading and stress ratio employed by Park et al. [21]. According to a study done by Forth et al. [23] crack closure will not occur during fatigue tests for coupons made from 7075-T73 if the stress ratio is greater than 0.5.

To account for residual stresses, LaRue and Daniewicz [24] suggested a superposition method where the SIFs due to residual stresses (K_R) were calculated and superimposed upon the SIFs calculated from the loading, $(K_{max})_L$ and $(K_{min})_L$, without residual stresses using

$$K_{max} = (K_{max})_L + K_R \quad (14)$$

$$K_{min} = (K_{min})_L + K_R \quad (15)$$

and calculating the stress ratio Rs using

$$Rs = \frac{K_{min}}{K_{max}} \quad (16)$$

2.2.4 The Simulated Growth Technique

The simulated growth technique combines the calculation of SIFs and the Paris growth technique and incrementally repeats the processes to simulate growth of the crack until it has reached a desired length. The main difference between different methods is how the new crack geometry is represented and whether the shape is predefined or arbitrary.

Newman and Raju [10] pioneered the simulated growth technique, calculating two stress intensity factors along the crack front, one at the free surface and the other at the deepest penetration point along the crack. They then used the Paris law to estimate growth increments at the two points and fit a semi elliptical crack which intersected the new points. This simplistic approach assumes an elliptical crack shape is maintained during the entire propagation. One disadvantage with this simplistic method is that crack propagation exhibits two main stages [19]. The first stage can be seen as a transient stage where the shape change is different for every crack and strongly dependant in the initial crack shape. In this stage the crack is trying to grow to a preferred shape so that the ΔK values and subsequent growth is constant over the entire crack front. The second stage is reached when the crack has achieved its preferred shape, at this stage the SIF values over the crack front are equal and this is referred to as an iso-K profile [25]. The transient stage of crack propagation cannot be represented with a simplistic model, therefore methods that use arbitrary crack front geometry were developed. One such method was created by Lin and Smith [9-11]. This method used the Paris law to estimate growth increments at seven to nine points along the crack front to enable significant shape variations to be modelled. Figure 2.10 a) shows the method used from Newman and Raju, simulating growth for two points and assuming the growth of the crack to be semi elliptical in shape. Figure 2.10 b) shows the method created by Lin and Smith using several points and fitting the new crack front with a spline. The advanced method gives a more accurate representation for growth of a crack that is non-elliptical in shape.

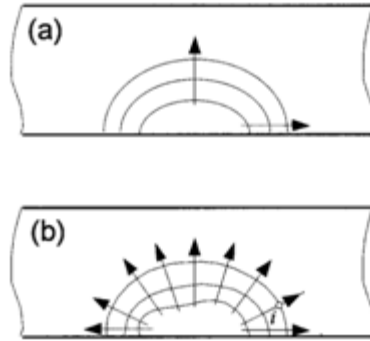


Figure 2.10: Fatigue growth calculation models: a) two points along the crack front assuming an elliptical shape, b) several points along the crack front using a spline to represent the shape [14].

There are advantages and disadvantages to representing the crack front with a spline or an ellipse. When fitting the projected points by an ellipse, a very smooth crack front is created, which provides advantages when creating and meshing a finite element model but accuracy is lost because not every projected point will fall on the new crack front, as seen with the blue ellipse fit to the crack front in Figure 2.11. By fitting the projected points with a spline, every projected point will be represented in the new crack front as seen with the dotted line in Figure 2.11. This allows for a more accurate representation of the crack shape but makes modelling and meshing more difficult.

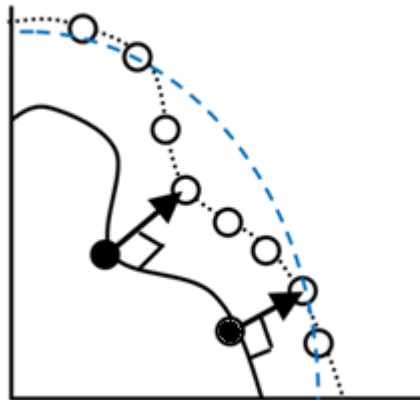


Figure 2.11: Ellipse fit (blue) compared to a spline fit for eight points along a crack front.

The differences in meshing crack fronts represented by splines and ellipses can be seen in Figure 2.12. With an ellipse fit, fewer elements are required over the crack front as seen in Figure 2.12 b). When the crack is fit with a spline, many more elements are required along the crack front as seen in Figure 2.12 a).

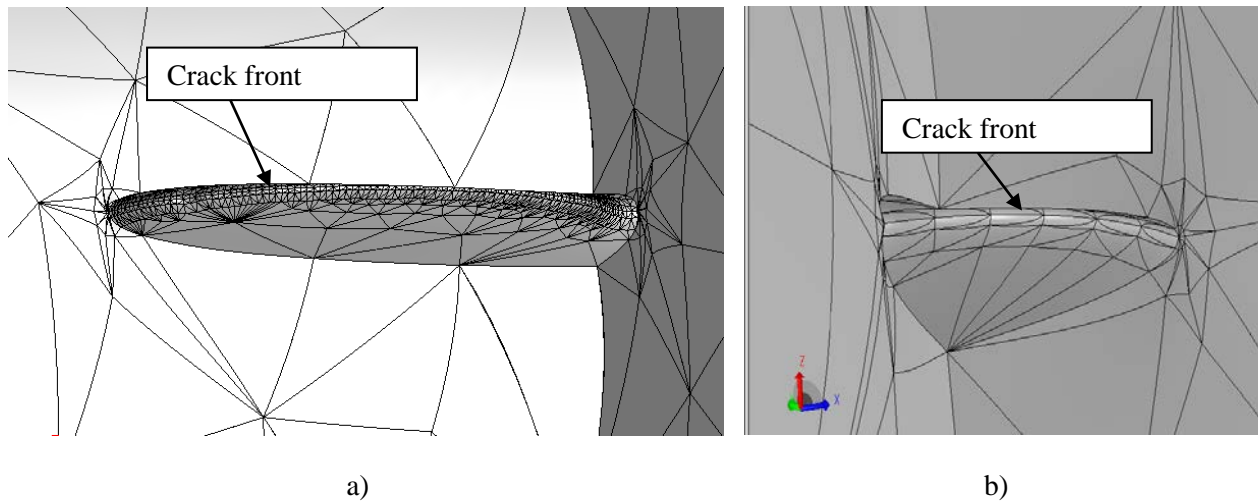


Figure 2.12: Mesh along the crack front for a corner crack with a) the crack front fit to a spline with many elements b) the crack front fit to an ellipse with fewer elements.

There is a limitation to the spline method created by Lin and Smith. This limitation is due to the fact that not enough points are extracted where the crack front intersects the free surfaces. The maximum SIF values are typically found close to the free surface but at the free surface there is usually a sharp drop in the SIF values. This is illustrated for the beta factor distribution for a corner crack in Figure 2.13 where the angle is normalized by dividing by 90° . Some authors have attributed the large drop at the free surfaces to singularities at the free surface or to errors in the computational program [26] and ignored them for incremental growth. Sevcik et al. [27] identified however, that the drop in stress intensity factors near the free surface is the mathematically accurate prediction of the SIF values for mode 1 loading. Knowing that SIFs can be accurately estimated at the edges, FEM with a large number of extraction points should be employed in these regions to predict crack shape as accurately as possible. To the

author's knowledge using more than nine extraction points for multiple iterations of crack growth, as performed in this thesis, has not been previously implemented.

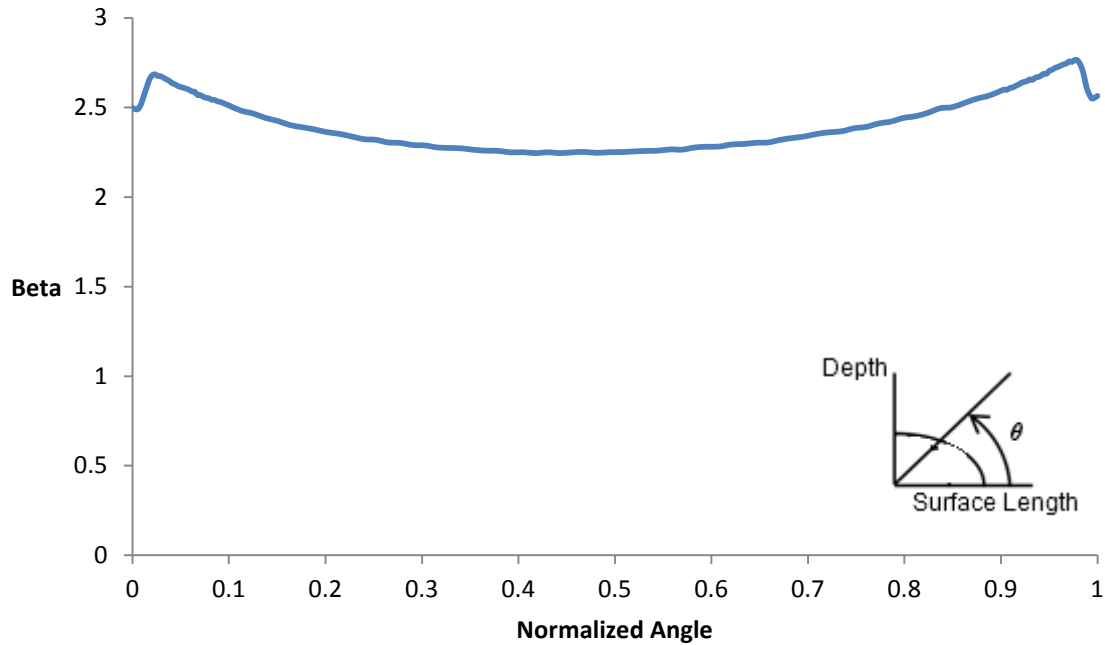


Figure 2.13: Beta factor distribution along the crack front of a corner crack, showing a drop in magnitude near the free surfaces.

2.2.5 The Geometry of the Crack Front

Computational analysis has been successful in utilizing fracture mechanics to discover how cracks grow and the factors that affect both growth and fatigue life. This information can assist in aircraft design for damage tolerance but can also be used to create better modeling techniques that generate more accurate solutions. The following two sections will overview some of the results that have been observed for general crack growth and growth in countersunk holes specific to aircraft and how the results can influence modeling techniques.

General Crack Front Geometry

Cracks of different shapes, but at similar locations will achieve a similar shape as the crack propagates through the material if the loading conditions, material and geometry are the same. Lin and Smith [28] observed this when studying three, quarter elliptical corner cracks with different initial

shapes. They found that though the initial growth shape was different for each crack, as they propagated they assumed a similar shape as seen in Figure 2.14. This has also been observed for cracks with an initial crack that has an irregular shape [28]. The initially irregular crack will very rapidly grow to a “smooth” crack shape and subsequently develop to a preferred pattern as illustrated in Figure 2.15. The shape is defined by the aspect ratio which is equal to the length of the crack along the hole divided by the length of the crack along the lower surface.

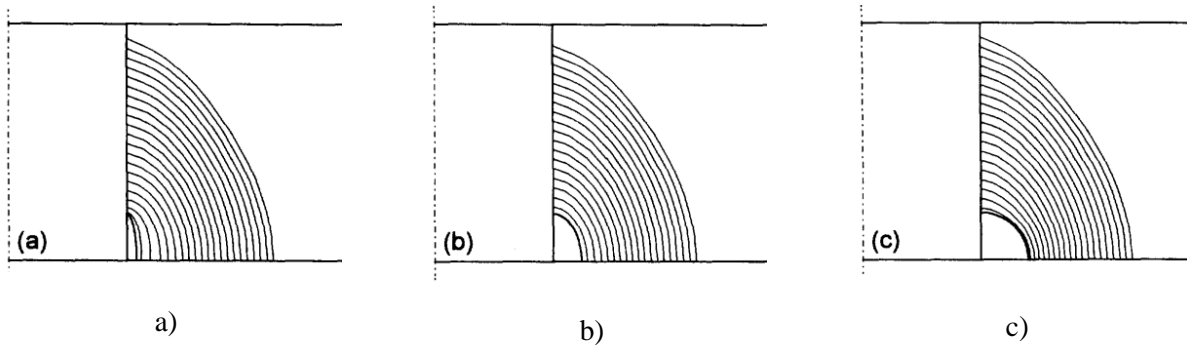


Figure 2.14: Cracks of different initial shape developing into a similar shape a) high aspect ratio b) aspect ratio >1 c) aspect ratio $=1$ [28].

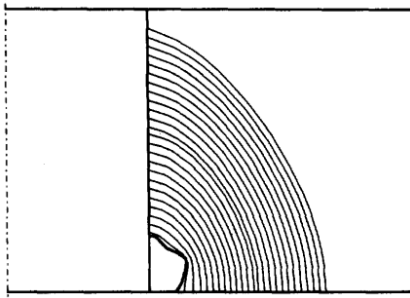


Figure 2.15: Crack with an irregular initial shape quickly becoming "smooth" and developing into a preferred shape [28].

The distribution of SIFs along the crack front dictates how a crack will propagate. If the SIFs are higher over one section of the crack, the crack will grow faster in that section. If a section of a crack is concave the SIFs in the concave section will typically be higher so that it grows to a shape with a uniform curvature [25]. Gozin observed a tunneling effect where the crack grows slower near free surfaces compared to the interior of a component [29]. The consequence of this in regards to accurately modelling growth means that non-uniform growth over the crack front should be utilized.

As the crack grows, it is trying to propagate to a “preferred” shape where the growth along the entire crack front is constant. This is achieved when the SIFs along the crack front are equal, also referred to as an *iso-K* state [30, 25]. This “preferred” shape is generally elliptical [31] with the shape of the ellipse dependant on loading, material properties and geometry [32]. This is why many studies have represented crack fronts with an ellipse.

Crack Emanating from Countersunk Holes

Cracks developing at countersunk fastener holes are an area of concern for the aerospace industry and the study of these cracks is important for damage tolerance analysis. Countersunk fasteners are used to join fuselage skins because they reduce the overall drag in an aircraft and they are typically found in lap joints using a row of rivets [33], as seen in Figure 2.16. When performing fatigue analysis on cracks in countersunk holes the configuration is typically assumed to be for a single countersunk hole in a plate. The behaviours observed for cracks emanating from a single countersink in a plate are representative of cracks in countersunk holes in lap-splice joints [34]. The crack interaction effects are not found until the later stages of the fatigue crack growth [35].

Two load conditions are usually found in fuselage lap joints: tension due to the hoop stresses introduced by the ground air ground cycle and bending due to the joint designs [34]. These loads can be seen in Figure 2.16.

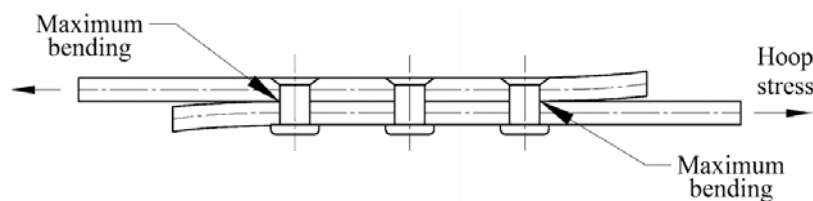


Figure 2.16: Geometry and loading found in three-row countersunk riveted lap joints [33].

Cracks in countersunk holes typically initiate near the sharp intersection of the countersink and the straight shank as knee cracks or where the straight shank and faying surface meet as corner cracks shown in Figure 2.17. Park et al. [36] determined that the location of initiation is dependent on the type of loading that the part experiences. Parts that undergo low load transfer (LLT) (representative of a upper wing skin to stiffener joint) initiate cracks near the sharp intersection of the countersink and the straight shank hole and parts that undergo high load transfer (HLT) (representative of wing double-butt joint) developed as a corner crack where the straight shank and faying surface meet.

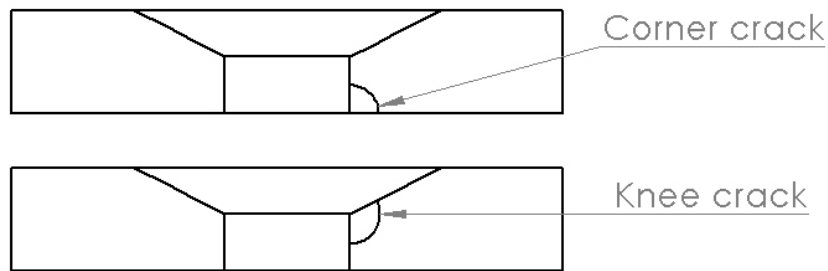


Figure 2.17: Location of a corner crack and a knee crack at a countersunk hole.

Grandt Jr. [37] concluded that a corner crack will grow faster along the shank and the countersink than it will along the lower surface. When the corner crack becomes a through crack the growth at the upper surface will be uniform until there is constant growth along the crack front as illustrated in Figure 2.18 a) with the locations where growth is faster shown in red and the location where growth is slower shown in blue. A knee crack will grow faster along the shank until it breaks through the bottom surface, after breakthrough it will grow faster along the bottom surface until it becomes a through crack where the growth increases along the top surface as shown in Figure 2.18 b).

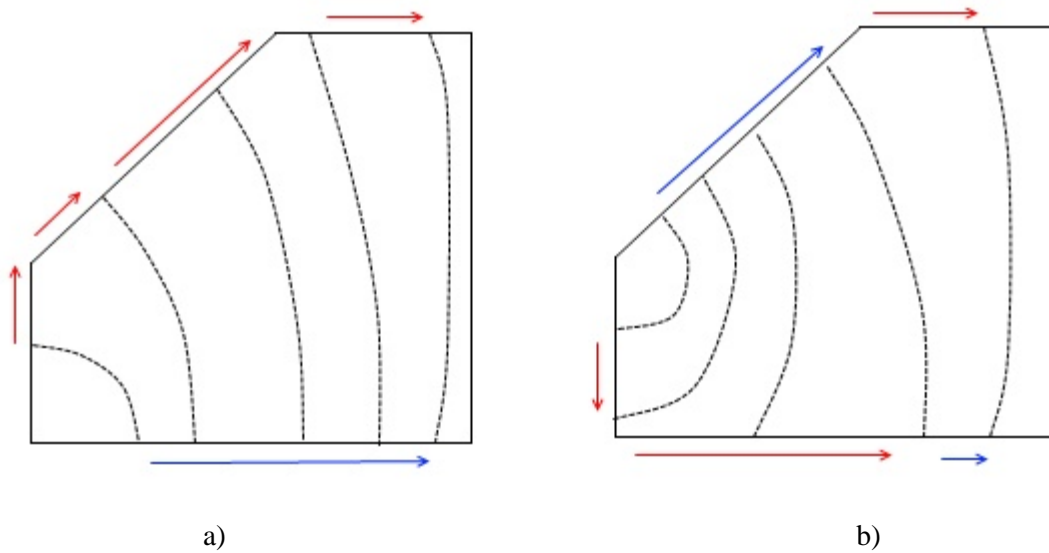


Figure 2.18: Growth pattern for cracks emanating from countersunk holes for a) a corner crack b) a knee crack. The surface locations of fastest and slowest growth are shown in red and blue respectively.

There are limited numerical modelling results of fatigue life and crack growth for cracks emanating from countersunk holes to be found in the literature. Many of the results to be found deal with the accurate predictions of SIFs over the crack front, with the crack front modelled as an ellipse or quarter circle [38, 34, 36, 37, 39]. These studies did not predict the actual shape of a crack front from a countersunk hole which is one of the capabilities of the ACGP.

2.2.6 Validation of the Numerical Method

Modern computers allow the mathematical formulas to be solved faster than before but it is important to note that the numerical analysis predicts behavior based on the formulation of mathematical models. When performing a numerical analysis it is essential that the results are as close to the exact solution as possible so that the only errors result from the mathematical models and not the numerical analysis [40]. When employing the finite element method the discretization error associated with element sizing is deemed to be acceptable when the solutions converge following mesh refinement. This means that any further refinement of the mesh will result in no change to the results. Convergence of a finite element analysis can be obtained using three different FEM methods. The *h-method*, utilized in the

commercial software ANSYS, uses 1st and 2nd order elements and achieves convergence by increasing the number of elements in the model. When applied for crack analysis the number of elements around the crack tip must be very high to achieve an accurate solution. The *p-method*, utilized in the commercial software StressCheck, enables users to increase the order of the elements up to 8. To achieve convergence, rather than refining the size of the mesh, the elemental order is increased. The mesh is extremely important but is not required to be as fine compared to the *h-method*. For crack analysis, fewer elements need to be generated around the crack tip and the higher order elements are more capable of capturing the singularity at the crack tip [16]. This thesis uses StressCheck along with an embedded automeshing to create well defined meshes. The *hp-method* is a combination of the *h* and *p-methods* where convergence is obtained by both the refinement of the mesh and an increase in polynomial order. If the *p-method* is used on a properly designed mesh it will produce the accuracy of the *hp-method* [40].

To determine the accuracy of a numerical method, it is important to validate the predicted results with experimental results, widely accepted methods or other software. One technique compares the predicted SIF values with experimental SIF values calculated using Anderson's back calculation method [41]. This method requires fatigue striation measurements to be conducted on fatigue specimens and the SIF values are determined using the spacing and number of cycles. There is typically large error between the analytical and experimental results when comparing SIFs because of the non-uniform, stepwise crack extension process seen in experimental results [42]. Fawaz and Hill concluded that the best method for validating numerical results is to use fatigue life, crack history and crack rate rather than comparing the SIF distribution using striation measurements [42].

Chapter 3 Description of the ACGP

The ACGP uses the commercial FEM software StressCheck and the commercial software Microsoft Excel to predict crack shape and fatigue life for cracks in thin plates undergoing mode 1 loading. The program utilizes an automated growth technique to incrementally grow a crack through the material until a desired crack length is reached. This program was developed to enable more accurate predictions of propagating crack shapes in three-dimensional bodies with the future goal of better predicting fatigue life in bodies with a complex geometry under a variety of loading conditions. Its advantages over some commercial programs, such as AFGROW, are that it enables a crack to propagate in a way that is more representative of the stresses along the crack front due to the fact that its geometry is represented by a spline. Since the crack front is not predefined to be an ellipse, as is typically assumed, it can take on any shape throughout its growth. The ACGP is based on LEFM but does not incorporate the effects of crack closure or residual stresses in the model. The term fatigue life, used in the ACGP and this thesis, refers to the number of cycles used to grow the crack to a certain length not the number of cycles before failure. The computer code for the ACGP can be found in Appendix B.

The current program allows for the propagation of mode 1 cracks by applying a traction load to one end of a plate and constraining the opposite end in all directions with a crack propagating parallel to the loaded and constrained faces, as shown in Figure 3.1. Currently, there are five versions of the program, each with their own Excel interface and separate program file. Each version models a crack in a plate originating from a different design feature. Version 1 models a corner crack emanating from a hole in a plate as seen in Figure 3.1 a), version 2 models a corner crack emanating from a notch in a plate as seen in Figure 3.1 b), version 3 models a corner crack emanating from a countersunk hole in a plate as seen in Figure 3.1 c), version 4 models a knee crack emanating from a countersunk hole in a plate as seen in Figure 3.1 c) and version 5 models a upper crack emanating from a countersunk hole in a plate as seen in Figure 3.1 c). In all cases, the initial corner crack is located at the intersection of the bottom surface of the plate and the feature (notch or hole). For the countersunk geometry initial locations at the intersection

between the hole and countersink (knee) and the upper surface and the countersink (upper) can also be modelled.

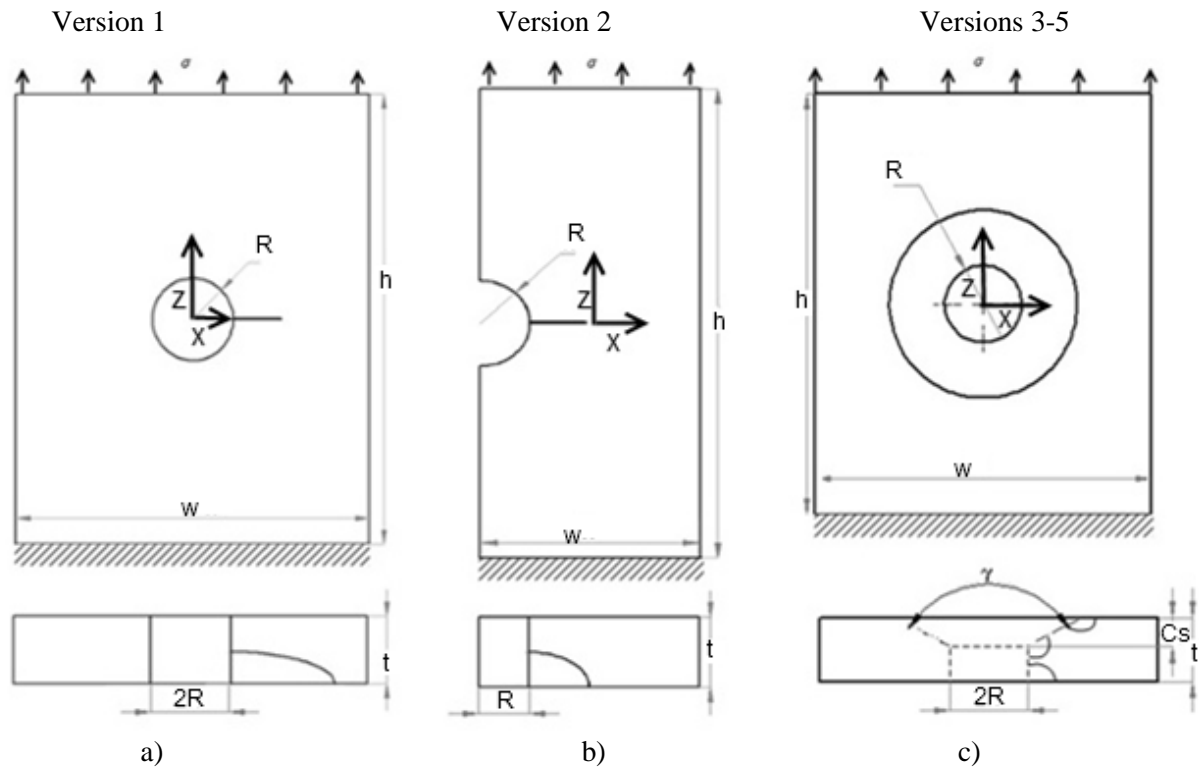


Figure 3.1: Possible plate geometries with applied loading and initial crack locations that can be modeled a) corner crack in a plate with a hole b) corner crack in a plate with a notch c) corner, knee and upper crack in a plate with a countersunk fastener hole.

Version 1 and version 2 were used to validate the techniques employed in the program and are discussed in Chapter 4. Versions 3-5 were used to predict how cracks behave in countersunk holes and are discussed in Chapter 5.

The techniques used to automate the crack growth prediction processes are based on the crack growth model created by Lin and Smith [9-11] to simulate growth at a crack front. In this thesis the process has been modified, based on observations in the literature, to improve the accuracy of the crack growth and fatigue life predictions.

The ACGP can be broken down into three main parts as illustrated in Figure 3.2: pre-processing (yellow), processing (blue) and post-processing (pink). During pre-processing, the geometry of the model and initial crack are input into the Excel interface along with the loading, meshing parameters, material properties and growth parameters. The processing stage can be broken down into three successive steps that are repeated until the desired number of iterations have been reached: creation of the FEM model, extraction of the SIFs and applying the crack growth model. The finite element model is created in StressCheck and the SIFs are automatically extracted using an embedded function within StressCheck. The calculations for the crack growth model are performed in Excel. The post-processing stage consists of analysing the results to generate the growth pattern and calculate the fatigue life within Excel.

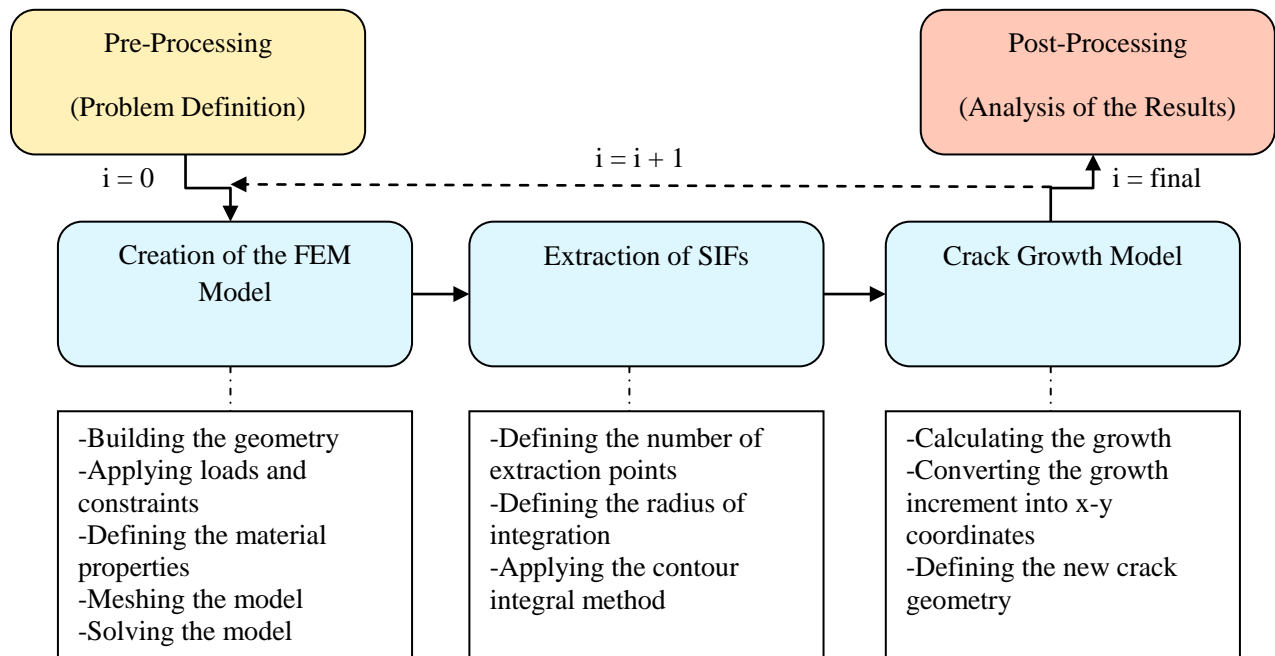


Figure 3.2: Steps of the automated crack growth program (ACGP), pre-processing shown in yellow, processing shown in blue and post-processing shown in red. i refers to the loops of the system.

The following sections describe the different steps in the program. Differences between the five versions of the program are noted.

3.1 Pre-Processing

During pre-processing the problem is defined by inputting values for the geometry, material properties, loading, mesh and growth into an Excel sheet where the program is launched. The version 3 user interface is shown in Figure 3.3. Cells requiring input are shown in orange. These values are used by StressCheck to automatically create and extract results from the finite element model and also by the crack growth and post-processing programs within Excel.

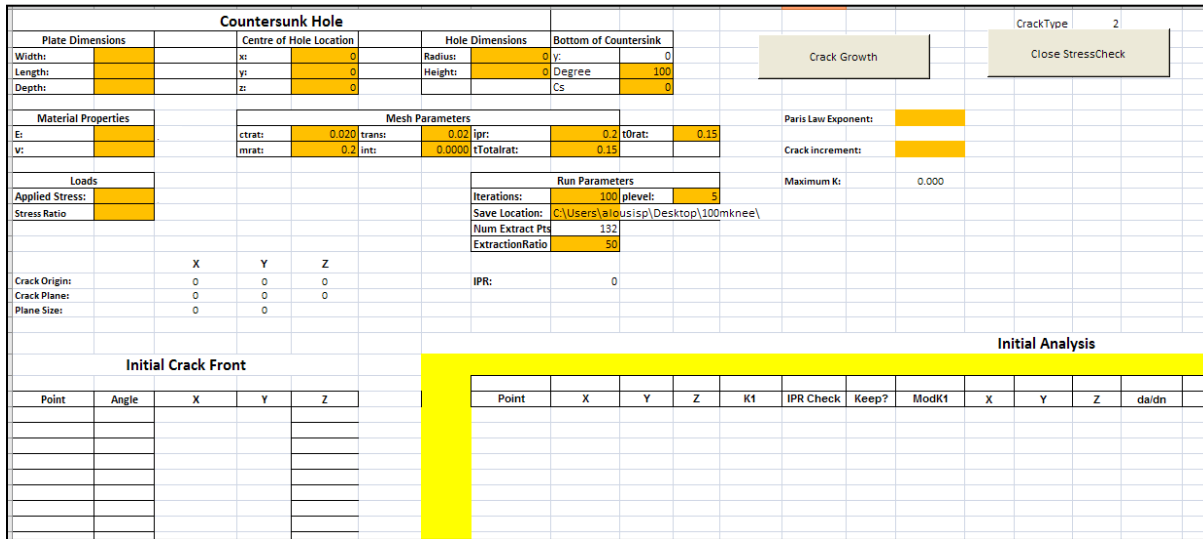


Figure 3.3: Version 3, Excel user interface.

Section 3.2 will detail the input values that are required and their purpose within the automated growth program. The values that are to be input will be bolded for identification purposes, and correspond to the orange cells in the spreadsheet.

3.2 Processing

3.2.1 Creation of the FEM Model

The first step in processing, is the creation of the FEM model within StressCheck. The model is automatically created using the geometry, material, loading and meshing parameters defined within the

pre-processing stage. Figure 3.4 shows a screenshot of the StressCheck model for a hole in a plate with the mesh of tetrahedral elements, loads and constraints applied.

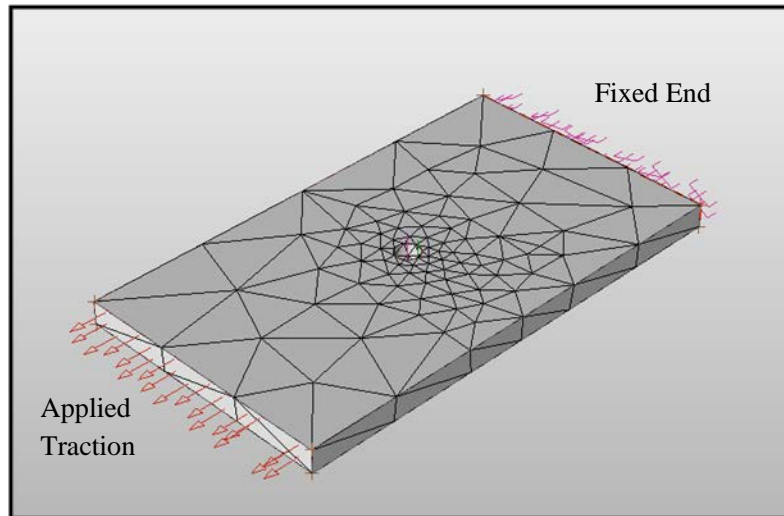


Figure 3.4: Screenshot of a plate with a hole in StressCheck showing the mesh, loads (arrows) and constraints (anchors).

Geometry

To create the model geometry the program requires the **length** (h), **width** (w) and **depth** (t) of the plate to be defined along with the location and dimensions of the feature to be input. Refer to Figure 3.1 for illustrations of these parameters.

- For a hole located in the centre of a plate, the **radius** (R) and **depth** (t) of the hole is defined. Its location with respect to the plate is set to (0,0,0) corresponding to its x, y and z coordinates located in the centre of the plate.
- For a notch in a plate, the **radius** (R) and **depth** (t) of the notch is defined. Its location with respect to the plate is set to (-w/2,0,0) indicating that it is located at the edge of the plate.
- For a countersunk fastener hole in a plate the **radius** of the straight shank (R) is defined along with the **countersink angle** (γ) and the **countersunk depth** (Cs). Its location with respect to the plate is set to (0,0,0) corresponding to its x, y and z coordinates located in the centre of the plate.

The program requires the initial geometry of the crack to be defined in terms of x and y coordinates as a series of points along the crack front. The coordinate system used to define x and y for a feature in the middle of a plate (hole or countersunk fastener hole) is located at the bottom surface at the centre of the feature as illustrated in Figure 3.5 a). The coordinate system used to define x and y for a feature located at the edge of a plate (notch) is located at the bottom surface at the edge of the plate as illustrated in Figure 3.5 b).

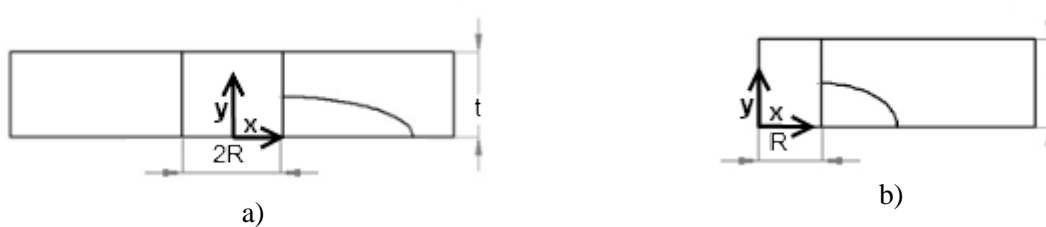


Figure 3.5: Co-ordinate system used to define the crack front a) if the feature is located in the middle of the plate b) if the feature is located at the edge of a plate.

Figure 3.6 shows where the coordinates are input for a corner crack in the Excel spreadsheet. The point with the smallest y coordinate (at the plate surface) is input first followed by successive points along the crack front as illustrated in Figure 3.7 a). For a knee crack emanating from a countersunk hole in a plate the co-ordinates with the smallest y-value are input first followed by the next successive coordinates until the crack front has been represented as illustrated in Figure 3.7 b). For an upper crack the co-ordinates with the largest y value are input first as shown in Figure 3.7 c).

Crack Front		
le	x (mm)	y (mm)
	0.47630043	0
	0.476204275	0.0059879
	0.466765374	0.0558143
	0.463886826	0.0615053
	0.460724704	0.0670441
	0.45726088	0.072407
	0.453490113	0.0775673
	0.411773126	0.1170045
	0.406470804	0.1207052
	0.3695	0.1377573

Plate w Hole

Figure 3.6: Example of x and y co-ordinate input for a corner crack front emanating from a hole with a radius of 0.3695mm.

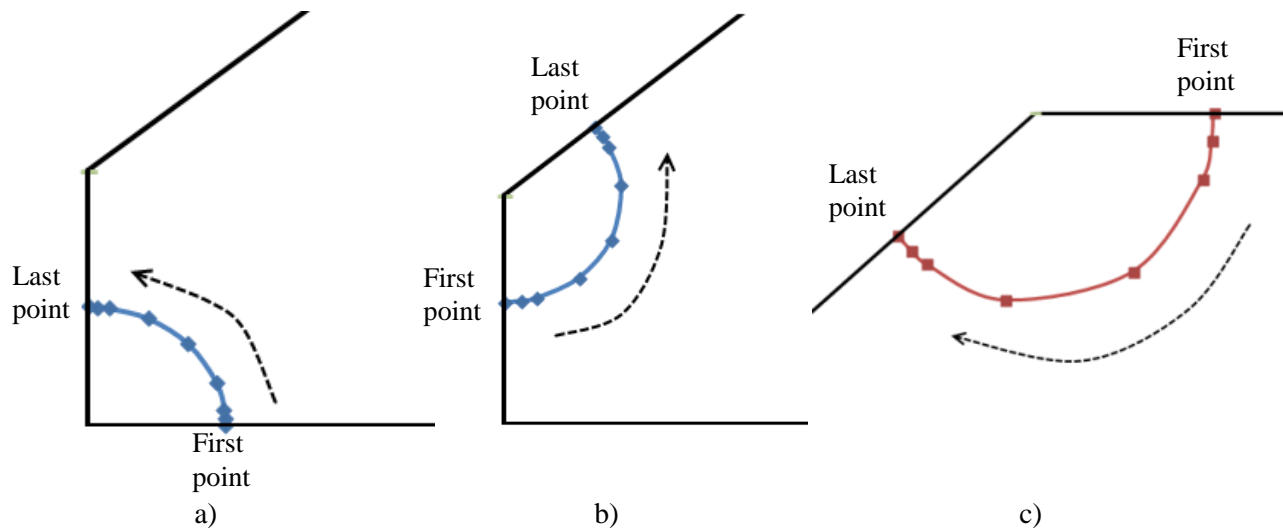


Figure 3.7: Order of points to be input during pre-processing for a) a corner crack b) a knee crack c) an upper crack.

A spline is fit through all of the points defined along the crack front. This enables complex crack shapes such as the example shown in Figure 3.8 to be modelled.

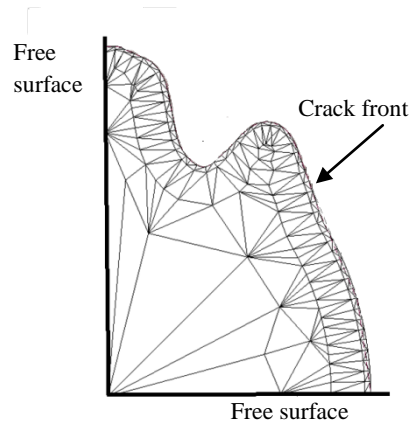


Figure 3.8: A spline fit to the crack front enables complex crack shapes to be modeled. Only the mesh on the crack face is shown.

Material Properties

The program requires the linear elastic material properties of **modulus of elasticity** (E) and **Poisson's ratio** (ν) to be input during the pre-processing stage. The values are used by StressCheck to solve for the stress distributions surrounding the crack tip.

Mesh Parameters

Because of the ability of StressCheck to automatically generate free meshes, meshing parameters are entered at the pre-processing stage in terms of ratios as defined below. The recommended numerical values provided are from Wowk et al. [16] who used using the commercial software StressCheck to determine accurate SIFs over an elliptical crack front emanating from a hole in a thin plate.

ctrat: The ctrat value defines the number of elements along the crack front. This value is calculated as 1 divided by the number of elements desired. A value of 0.08 or lower is recommended [16].

mrat: This value defines the ratio between the largest and smallest elements in the mesh of the entire model. A value of 0.8 or lower is recommended [16].

trans: This value defines the rate of change of the element size. If this value is high a large element is permitted to be beside a small element. The smaller this value, the more gradual the change is from large to small elements. A value of 0.08 or lower is recommended [16].

tTotalrat: This value defines the size of the 2nd row of elements along the crack front, as shown in Figure 3.10, compared to the smallest crack dimension (*int*), as shown in Figure 3.9. The value recommended by StressCheck is 0.15 [17] making the diameter of the 2nd row of elements 0.15 times the smallest crack dimension.

t0rat: A ratio that defines the size of the 1st row of elements along the crack face compared to the 2nd row of elements as shown in Figure 3.10. StressCheck recommends value ranges from 0.1-0.15 [17]. A value of 0.1 will make the 1st row of elements 0.1 times the size of the second row of elements.

int: The length of the smallest crack dimension, typically along a free surface. For corner cracks this value is either the length along the feature, as seen in Figure 3.9 a), or the length along the plate surface as in Figure 3.9 b). For knee cracks this value is the largest horizontal distance between the straight shank and the crack front, as illustrated in Figure 3.9 c). For upper cracks this value is the length of the crack along the upper surface, as seen in Figure 3.9 d). The *int* value automatically changes for each iteration.

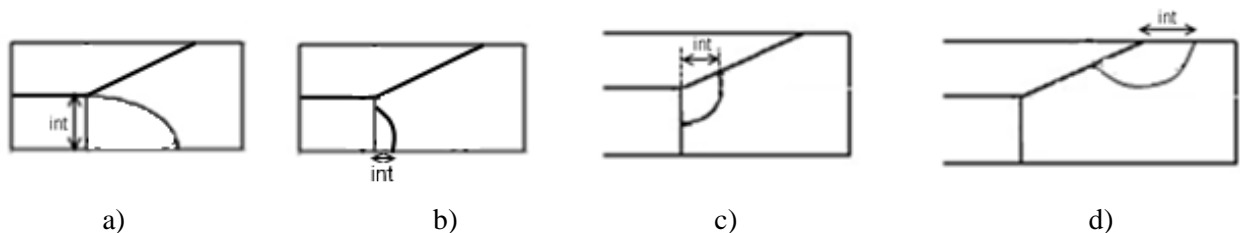


Figure 3.9: Possible *int* values for a) a corner crack with the shortest crack length along the straight shank b) a corner crack with the shortest crack length along the lower surface c) a knee crack d) an upper crack.

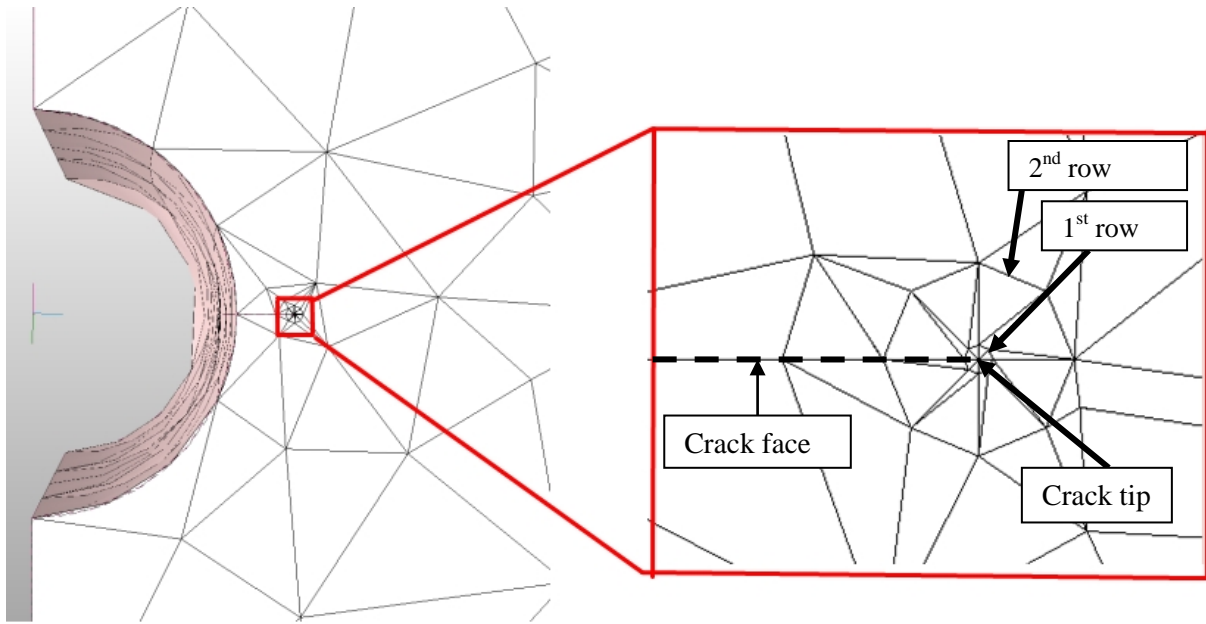


Figure 3.10: Typical crack tip mesh.

Figure 3.10 illustrates the typical mesh surrounding the crack tip. The radius of the 2nd row of elements (OR) is calculated using

$$OR = tTotalrat * int \quad (17)$$

The radius of the 1st row (IR) of elements is calculated using

$$IR = OR * t0rat \quad (18)$$

Figure 3.11 shows a series of meshes which highlight how changing the mesh parameters affect the mesh of the crack front and the mesh of the body.

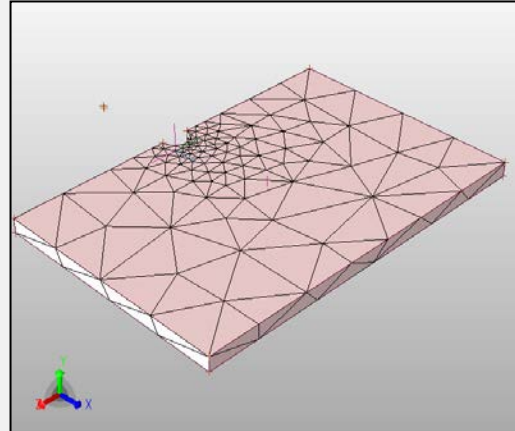
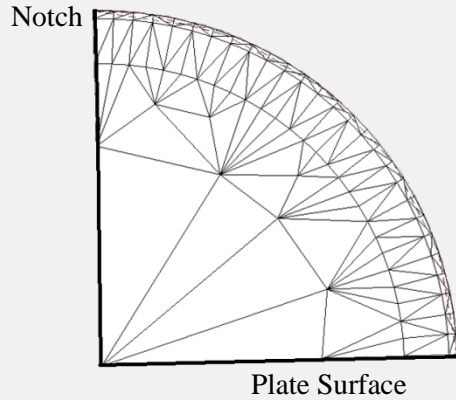
Mesh Parameters

Crack Front

Body

Baseline

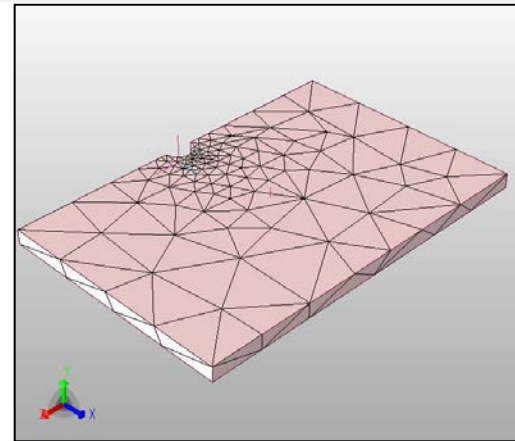
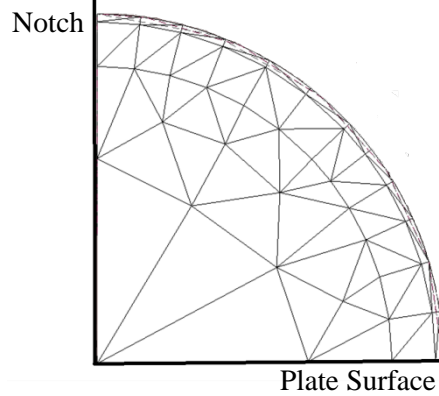
ctrat=0.04
mrat=0.2
trans=0.02
tTotalrat=0.15
tOrat=0.15



a) The mesh of the crack and body using the baseline parameters.

Reducing the number of elements along the crack front

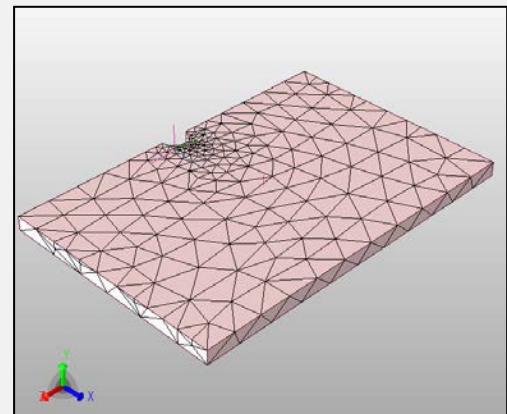
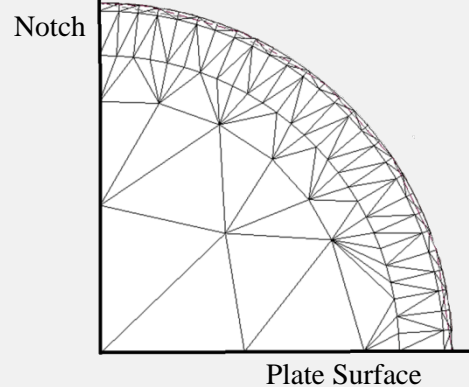
ctrat=0.04->
ctrat=0.1



b) Reducing the value of ctrat reduces the number of elements along the crack front.

Reducing the size ratio between the largest and smallest mesh elements

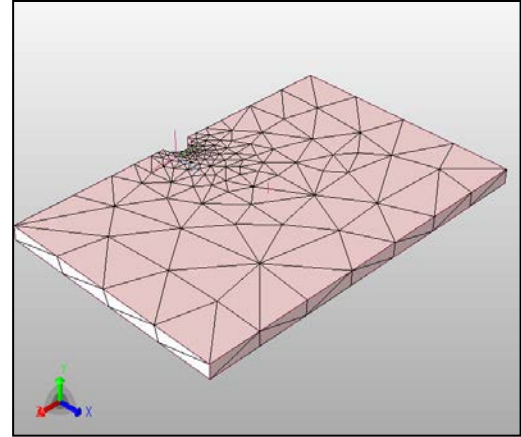
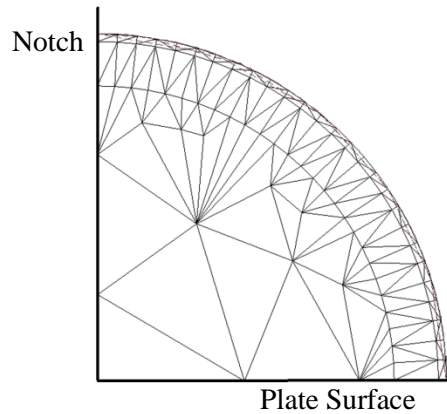
mrat=0.2->
mrat=0.1



c) Reducing the value of mrat results in elements in the body having a more uniform size.

Increasing the rate of change in element size

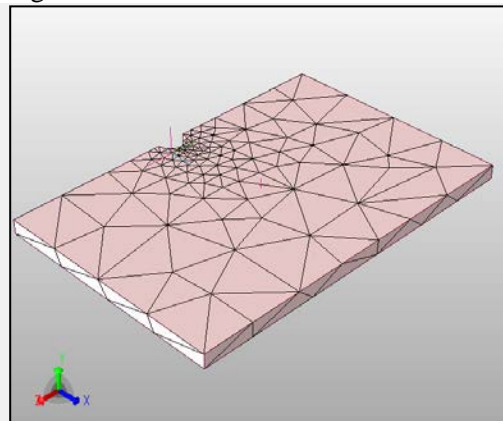
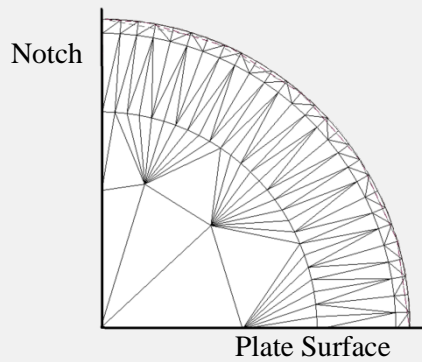
trans=0.02 ->
trans=0.05



d) Increasing the value of trans permits larger elements to be beside smaller elements.

Increasing the radius of the 2nd row of elements

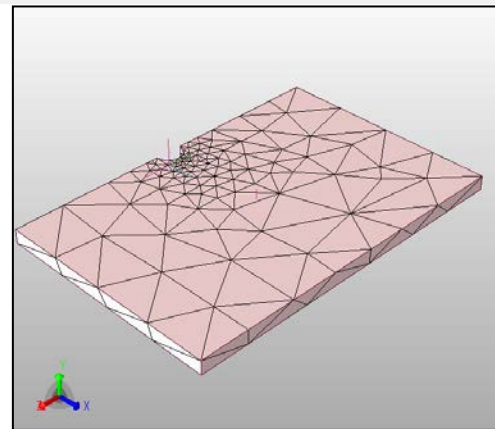
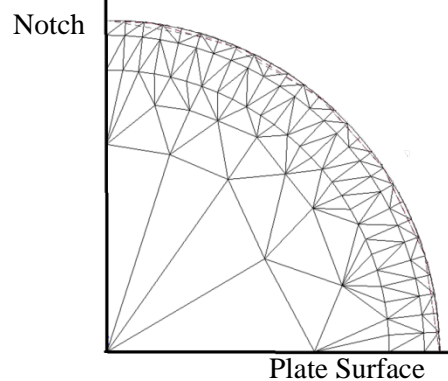
tTotalrat=
0.15 ->
tTotalrat=0.3



e) Increasing the value of tTotalrat causes the radius of the 2nd row of elements to be increased which causes the radius of the 1st row of elements to be increased.

Increasing the diameter of the first row of elements

t0rat= 0.15 ->
t0rat=0.3



f) Increasing the value of t0rat causes the radius of the 1st row of elements to be increased.

Figure 3.11: Effect of changing the meshing parameters on the crack front and the body.

Loading and Boundary Conditions

The program requires the **applied stress** and **stress ratio** values to be input during the pre-processing stage. The applied stress value is the traction load applied to the plate in StressCheck and the stress ratio (Rs) is used in the growth model in Excel. The fixed end displacement is automatically applied to the StressCheck body and cannot be altered.

Solution

The program requires the **p-level** to be input during the pre-processing stage. This value defines the order of the element shape function used in StressCheck and can range from 1-8. The finite element model is then solved as a linear analysis to determine the stress distribution at the crack tip. Once the model is solved, SIFs are extracted at discrete points along the crack front. The recommended p-level from Wowk et al. is 5 [16].

3.2.2 Extraction of the Stress Intensity Factors

StressCheck determines the SIFs at the crack tip values using the contour integral method at a pre-defined number of points specified during pre-processing. In version 1 of the program the exact number of extraction points is specified in the **Num Extract Pts** cell. In versions 2-5 the number of extracted SIFs along the crack front is determined using an **ExtractionRatio** defined during pre-processing (values are constant and typically range between 5 and 50). The number of extraction points at the first and last 10% of the crack front is determined using Equation (19). For the remaining 80% of the crack front the number of extraction points is calculated using Equation (20) and are evenly spaced. This results in more extraction points in the first and last 10% of the crack front as shown in Figure 3.12.

$$\text{First and last 10\%:} \quad \textit{Extraction Number} = \textit{int} * \textit{ExRatio} + 13 \quad (19)$$

$$\text{Remaining 80\%:} \quad \textit{Extraction Number} = \textit{int} * \textit{ExRatio} * 2.5 + 13 \quad (20)$$

The extraction ratio equations were chosen to provide a high number of extraction points near the free surfaces but have not been optimized for efficiency. This should be an area of focus for future versions of the program.

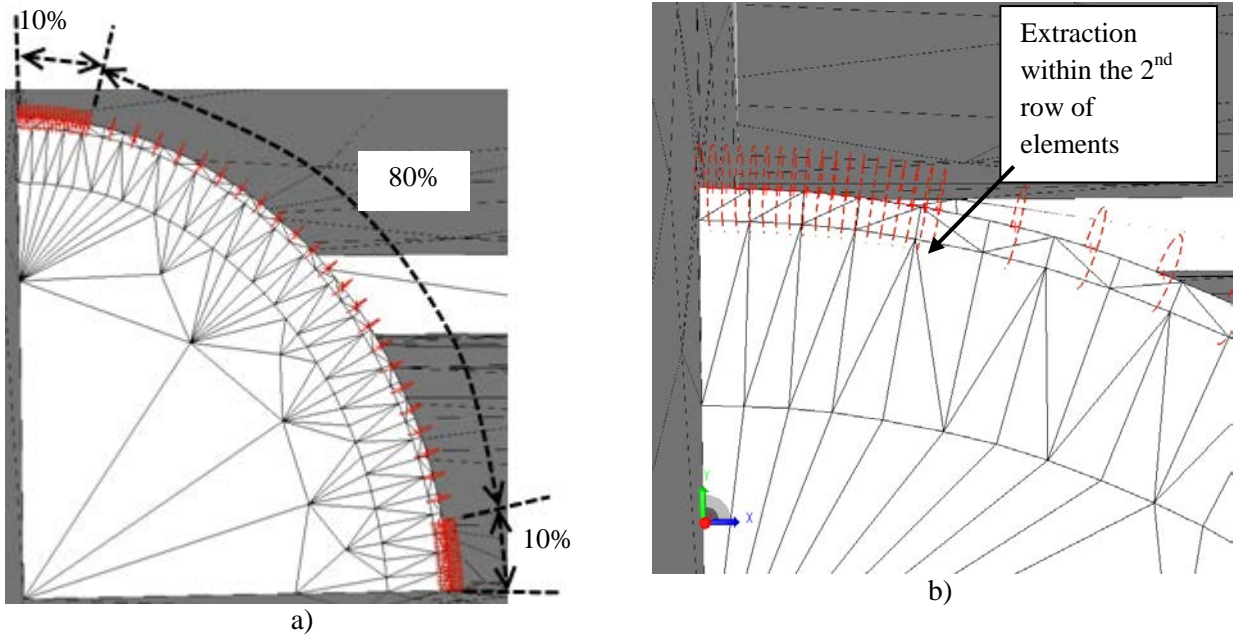


Figure 3.12: Version 2-5 extraction along the crack front a) entire crack front illustrating more points in the first and last 10% of the crack front b) close up at the last 10% of the crack front.

StressCheck uses the CIM to calculate SIFs for the defined extraction points. The accuracy of the CIM is dependent on the radius of integration which is calculated for each iteration using Equation (21). The radius of integration must pass through the second row of elements because, as stated in Chapter 2, the first row of elements contains the stress singularity [17].

$$\text{Radius of integration} = \text{ipr} * \text{tTotalrat} * \text{int} \quad (21)$$

where the **ipr** value is defined during preprocessing and determines the location within the 2nd row of elements through where the integration path radius passes. This value must range from 0.15 to 1 so that the radius of integration falls within the 2nd row of elements around the crack front, as seen in Figure 3.12 b). The default value provided by StressCheck is 0.2 [17]. Figure 3.13 illustrates where the radius of

integration occurs within the 2nd row of elements. At each extraction point, the x and y coordinates and the stress intensity factor are extracted for use in the growth model.

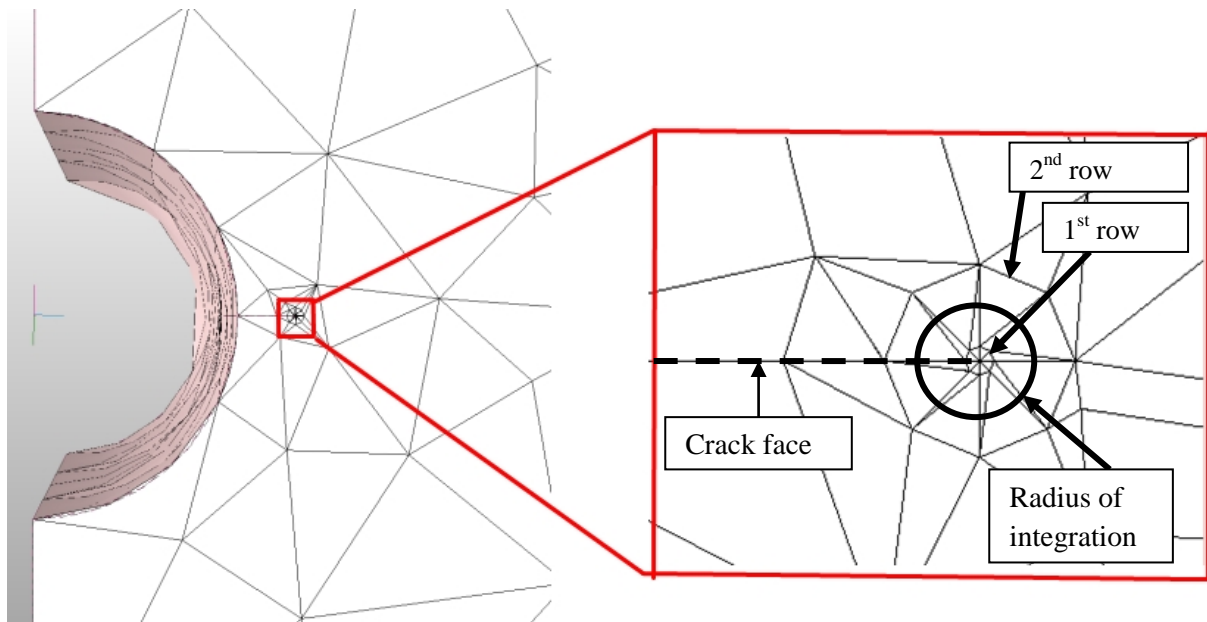


Figure 3.13: Location of the radius of integration at the crack tip.

3.2.3 Growth Model

The crack growth model uses equations programmed into Excel to predict the new locations of points along the crack front using the same principles employed by Lin and Smith [14]. The new locations are based on the distribution of SIFs along the crack front and are projected perpendicular to the tangent of the crack front as shown in Figure 3.14 b). The largest growth occurs at the location where ΔK is maximum, shown as ΔK_{max} in Figure 3.14 a) and a spline is fit to the new points, illustrated in Figure 3.14 c) to produce the newly formed crack front.

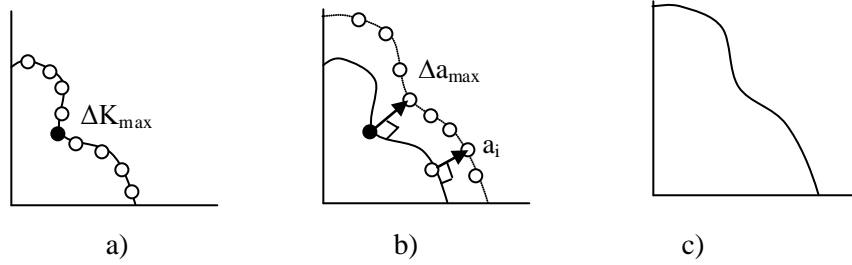


Figure 3.14: The crack growth process a) location of the highest ΔK value b) projected points along the crack front c) new crack front fit with a spline.

The first step in the growth model is to determine whether the circular extraction path defined by the radius of integration falls within the domain of the finite element model. This is done by analyzing the position and location of the extracted points and using Equation (22).

$$y_{IPR} = \sin\left(\tan^{-1}\left(\frac{x_2 - x_1}{y_2 - y_1}\right)\right) * IPR \quad (22)$$

where (x_1, y_1) and (x_2, y_2) are coordinates defined in Figure 3.15 and IPR is the radius of integration. y_{IPR} is the y component of the radius of integration. If y_{IPR} is larger than y_2 , the extraction path has fallen outside the plate geometry and will produce erroneous results. Using this method the first and last points are always excluded (unless there is a 90° intersection between the crack front and the edge). A similar equation is used to determine the horizontal distance of the extraction path used for the portion of the crack that intersects a vertical free surface.

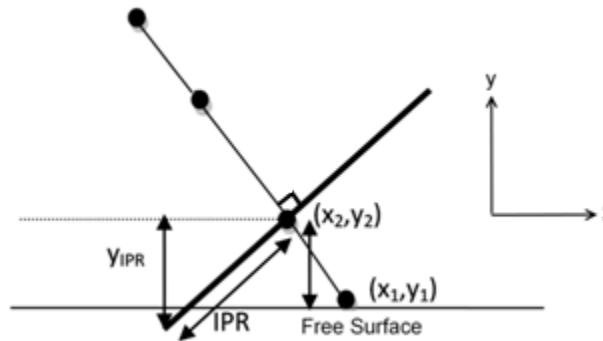


Figure 3.15: Radius of integration at point 2 falling outside of the model domain.

If the extraction path falls outside of the model domain, the SIF results are excluded from the growth calculations and no new coordinates are predicted for the excluded points. This usually occurs at the first and last extraction points because they are typically extracted at the free surface and don't intersect the crack front at 90°. Figure 3.16 shows an extraction point at the free surface falling outside of the StressCheck model domain.

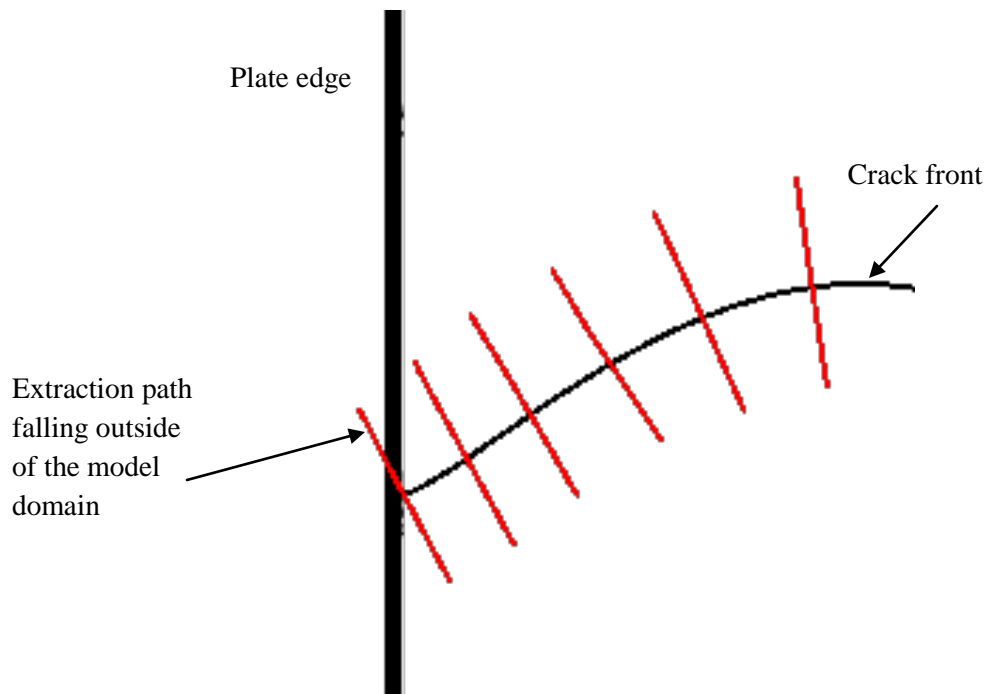


Figure 3.16: An extraction path (red lines) at the free surface falling outside the model domain.

The next step in the growth model is to use the predicted SIFs to calculate ΔK_i s using the stress ratio (R_s) defined during pre-processing and Equation (23).

$$\Delta K_i = K_i(1 - R_s) \quad (23)$$

where K_i represents the SIFs values predicted at specific points. The growth increment Δa is then calculated for every point along the crack front using Equation (24).

$$\Delta a_i = \left(\frac{\Delta K_i}{\Delta K_{max}} \right)^m \Delta a_{max} \quad (24)$$

The maximum crack increment Δa_{max} and the Paris exponent (m) are defined during pre-processing. The maximum crack increment is typically set to 0.1-1% of the thickness of the plate. If the increment is set too high, errors will occur within the model because the concave regions of crack front could overlap when projected. If the increment is set too low, the ACGP will be inefficient and require an excessive number of iterations to reach the desired crack length. The crack increment is projected normal to the tangent of the crack front at each extraction point. To create the new crack geometry each Δa must be converted into x and y coordinates. This is done by first calculating the slope of the tangent of the crack front at each extraction point using

$$Slope_i = -\frac{x_{i-1} - x_{i+1}}{y_{i-1} - y_{i+1}} \quad (25)$$

The absolute values of the x and y components of growth are then calculated using Equations (26) and (27).

$$|x \text{ growth component}_i| = \Delta a_i * \frac{1}{\sqrt{Slope^2 + 1}} \quad (26)$$

$$|y \text{ growth component}_i| = \Delta a_i * \frac{Slope}{\sqrt{Slope^2 + 1}} \quad (27)$$

An illustration of the projection of a new point normal to the tangent of a point along the crack front is shown in Figure 3.17.

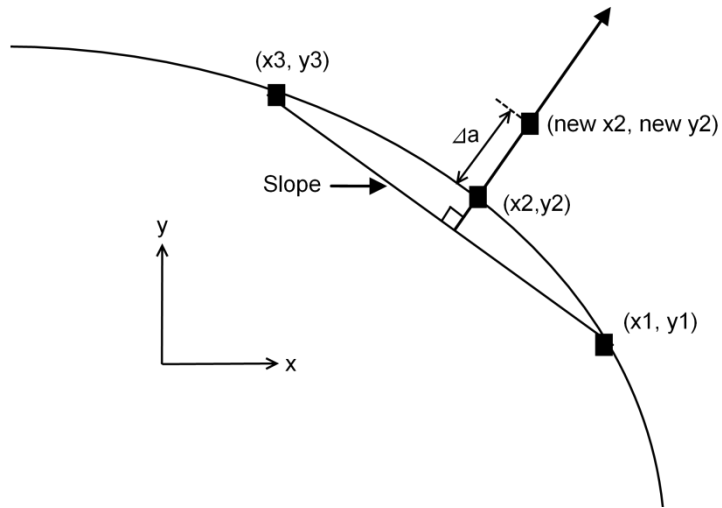


Figure 3.17: Projection of a new point normal to the tangent of a point along the crack front.

To enable outward growth at every point along the crack front the sign of the growth components at certain stages along the crack front is altered. These three stages are shown in Figure 3.18 for a corner crack.

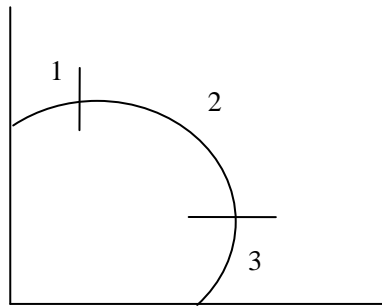


Figure 3.18: Stages of crack growth.

In stage 1 the x growth component is negative and the y growth component is positive, in stage 2 the x growth component is positive and the y growth component is positive, in stage 3 the x growth component is positive and the y growth component is negative. The x and y growth components are then added to the original coordinates of each point to generate the position of the new crack front. The first and last 4

points of the new geometry are linearly extrapolated to determine the coordinates of the points at the intersection of the free surfaces. Figure 3.19 illustrates the extrapolation using four points. Preliminary results had shown that using a linear extrapolation is more robust than using a polynomial extrapolation and did not alter the final fatigue or crack shape predictions.

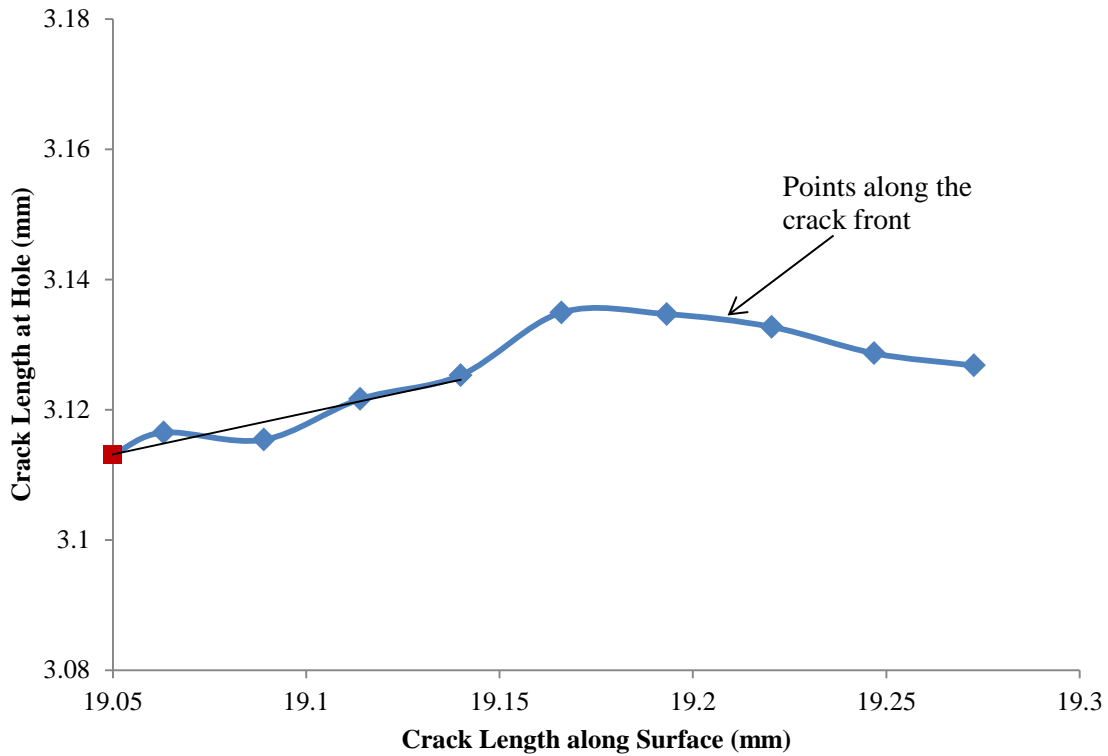


Figure 3.19: Linear extrapolation of the last point at the intersection of the crack and the feature.

For versions 2-5, every 3rd point is used to define the crack geometry within the first and last 10% of the crack front. This value was chosen after a series of preliminary tests and to prevent extraction errors from occurring due the creation of an unnecessarily fine mesh created if every point was fit.

Each processing stage is repeated until the program has completed the number of iterations defined in the pre-processing stage. If the crack has not reached the desired crack length during the processing stage the program can be continued.

3.3 Post-Processing

Post-processing takes place in a separate spreadsheet within Excel, shown in Figure 3.20. The Paris constant (C) is input, along with the initial number of cycles corresponding to the initial crack length.

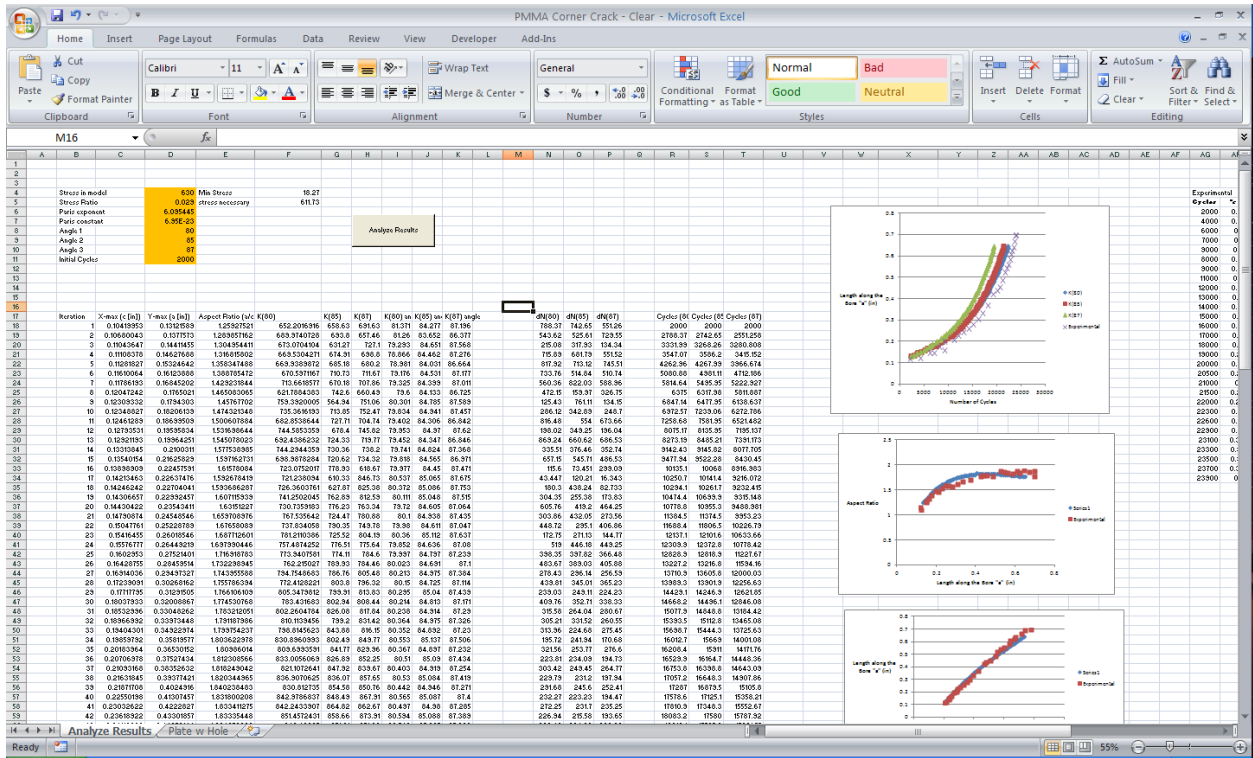


Figure 3.20: Version 1 post-processing interface.

Post-processing begins by gathering the required data generated from each processing iteration. For each iteration the a and c crack dimensions are gathered, as defined in Figure 3.21. The ΔK and Δa values at a point along the crack front (typically at an angle of 25° along the crack front) are also gathered to determine the number of cycles during that iteration using Equation (28).

$$dN = \frac{\Delta a}{C \Delta K^m} \quad (28)$$

The location along the crack front where the ΔK and Δa values are taken does not matter because the ratio of $\Delta a/\Delta K^m$ is the same for each point when using Equation (24). Using the ΔK and Δa values at any point will produce the same dN .

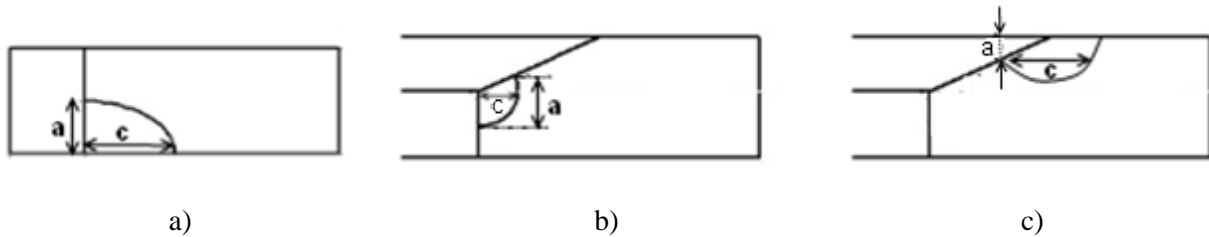


Figure 3.21: a and c values gathered for a) a corner crack b) a knee crack and c) a upper crack.

The aspect ratio for each processing iteration is calculated using Equation (29), which provides a way to quantitatively compare the crack shape.

$$\text{Aspect ratio} = \frac{a}{c} \quad (29)$$

The total number of cycles is calculated by adding the number of cycles for every iteration and a plot of the crack length versus the number of cycles is automatically created.

Chapter 4 Validation of the ACGP Program

This chapter presents and discusses the studies performed to validate the accuracy of the ACGP program for predicting crack growth and fatigue life of corner cracks in thin plates under mode I loading. Section 4.1 covers the validation of the StressCheck finite element model used to predict SIFs, section 4.2 covers the validation of the growth model used to predict fatigue life, section 4.3 compares results from complete runs of the ACGP and AFGROW predictions and section 4.4 compares the results from the ACGP to experimental results provided by the National Research Council Canada (NRC) [43].

4.1 Accuracy of the Stress Intensity Factor Predictions

The accuracy of the SIF predictions automatically generated from the StressCheck model within the ACGP was evaluated using two different geometries: two symmetric corner cracks emanating from a hole in a plate and a single corner crack emanating from a hole in a plate.

4.1.1 Two Symmetric Corner Cracks Emanating from a Hole in a Plate

Problem Description

The ability of the ACGP to predict accurate SIFs was first evaluated through comparison with published numerical SIF predictions for two symmetric elliptical corner cracks emanating from a hole in a plate as seen in Figure 4.1. Numerical predictions are used to validate the SIF prediction accuracy because, as stated by Fawaz and Hill [42], it is difficult to measure SIFs directly from experimental results due to the stepwise growth along the crack front that is observed in reality.

The published results from Bakuckas [13] and Fawaz and Andersson [9] were used to validate the ACGP SIF predictions. Bakuckas, as mentioned in Chapter 2, found a $\pm 3\%$ range in SIF predictions along an elliptical crack front emanating from a hole in a plate with two symmetric corner cracks using seven different numerical methods. Fawaz and Andersson generated results using an *hp*-version finite element analysis employing a strongly graded mesh with seven elements along the crack front and extracting SIFs using the contour integral method. Fawaz and Andersson state that the error in their solutions is less than

1% [9] and their solutions are currently implemented in the commercial program AFGROW [44]. Two sets of dimensions were used for validation; the first accounts for finite width and the second eliminates the finite width effects by modelling a larger plate geometry. The first set of dimensions was used by Bakuckas and is shown in Figure 4.1 along with the loading and material properties. The geometry of the plate and hole are represented as ratios with respect to plate width and thickness and the aspect ratio of the crack was specified to be 0.8. The modulus of elasticity (E) was 0.0069MPa (1psi), and Poisson's ratio (ν) was 0.3. These values were not meant to represent a real material but rather a "unit" case and the modulus of elasticity does not affect the value of the predicted SIF result. The applied stress, indicated as S_t , is 0.0069MPa (1psi) and the crack front was fit to a quarter ellipse with a horizontal radius (major radius) c and a vertical radius (minor radius) a .

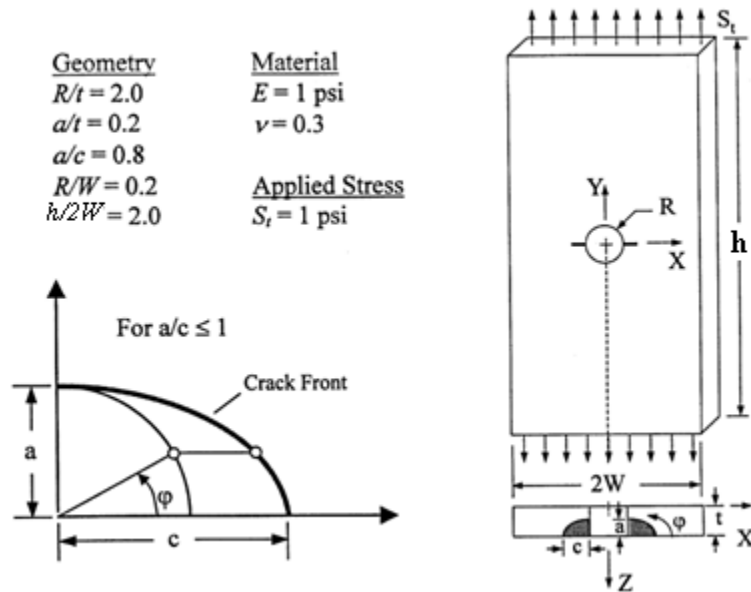
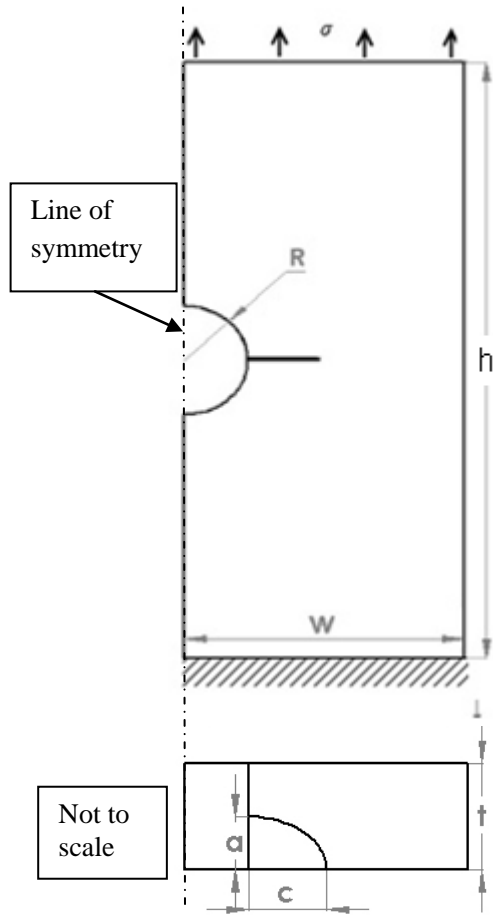


Figure 4.1: Geometry, material and applied stress used to generate SIF results from Bakuckas [13].

To eliminate width effects, Fawaz and Andersson altered the $h/2W$ and R/W ratios to $h/2W=1$ and $R/W=0.01$ to represent a wider plate using the same hole radius.

To recreate the tests from the published results, version 2 of the ACGP was employed which simulates an elliptical corner crack emanating from a notch in a plate. Symmetry was used along the face

containing the notch to simulate two symmetric corner cracks in a hole. One iteration of the ACGP was run for the specified geometry to generate SIF results. It was not necessary for the growth model within the ACGP to be used. The geometry ratios, material and loading specified in the published results were input. A radius of 25.4mm was chosen to represent the radius of the hole and the remaining geometric values were determined using the specified ratios. The crack front was fit to a quarter ellipse with a horizontal radius of 3.18mm and a vertical radius of 2.54mm. The *ctrat*, *mrat* and *trans mesh* values were set to 0.04, 0.1 and 0.02 respectively to produce a sufficient number of elements along the crack front and for the global mesh. The geometric, modelling and meshing parameters used in the ACGP are shown in Figure 4.2 with the values for the geometry and mesh used for the wider plate shown in brackets.



Model Parameters

Stress	0.0069MPa
Stress Ratio	0
E	0.0069MPa
v	0.3

Geometric Parameters

h	508mm (5080mm)
W	127mm (1270mm)
a	2.54mm
c	3.175mm
R	25.4mm
t	12.7mm

Meshing Parameters

ctrat	0.04 (0.02)
mrat	0.1
trans	0.02
ipr	0.2
tTotalrat	0.15
tOrat	0.15

Figure 4.2: Model, geometry and meshing parameters for two symmetric corner cracks emanating from a hole in a plate. The second geometry is shown in brackets.

The SIF results were compared in the form of geometric factors and normalised parametric angles, to display the magnitude and their location along the crack front respectively. The geometric factors ($Y(a)$) were calculated using

$$Y(a) = \frac{\Delta K_I}{\sigma \sqrt{\frac{\pi a}{Q}}} \quad (30)$$

Q is the elliptical shape factor and is dependent on the crack shape [13]. For an ellipse, the values of Q are

$$Q = 1 + 1.464 \left(\frac{a}{c}\right)^{1.65} \quad \text{for } \frac{a}{c} \leq 1 \quad (31)$$

$$Q = 1 + 1.464 \left(\frac{c}{a}\right)^{1.65} \quad \text{for } \frac{a}{c} > 1 \quad (32)$$

The normalised parametric angle, θ , used to indicate the position along the crack front is measured with reference to a quarter circle contained within the crack. The angles are normalised by dividing the angle at a point by the maximum angle. For a corner crack in a plate the parametric angle is measured from the surface of the plate to the point, shown in Figure 4.3, and each angle is normalized by dividing by 90° .

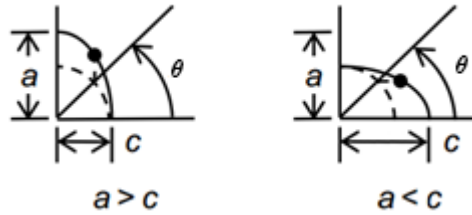


Figure 4.3: Parametric angle of a corner crack for two different aspect ratios.

Results and Discussion

The SIFs predicted from the ACGP are compared to the published results in Figure 4.4. The ACGP results, shown as smooth lines, fall within the $\pm 3\%$ range predicted by Bakuckas indicating that the ACGP is able to predict SIF results for cracks in three-dimensional solids with an accuracy comparable to a number of computational methods. When comparing the ACGP results with the Fawaz and Andersson results (indicated by points in Figure 4.4) the distribution of the geometric factors match well along the crack front though the ACGP results are consistently higher by an average of 2.9%. Both the ACGP and Fawaz and Andersson predict a sharp drop in SIFs near the free surface, with the drop in the Fawaz and Andersson results occurring closer to the free surface. All methods show SIF values that are higher near the bore of the hole than the plate surface. This is due to the concentration of stresses near the edge of the hole which decrease as the distance from the hole increases. Finite width effects can be

seen by both the ACGP and by Fawaz and Andersson as curves that are 5.9% higher than the infinite width plate.

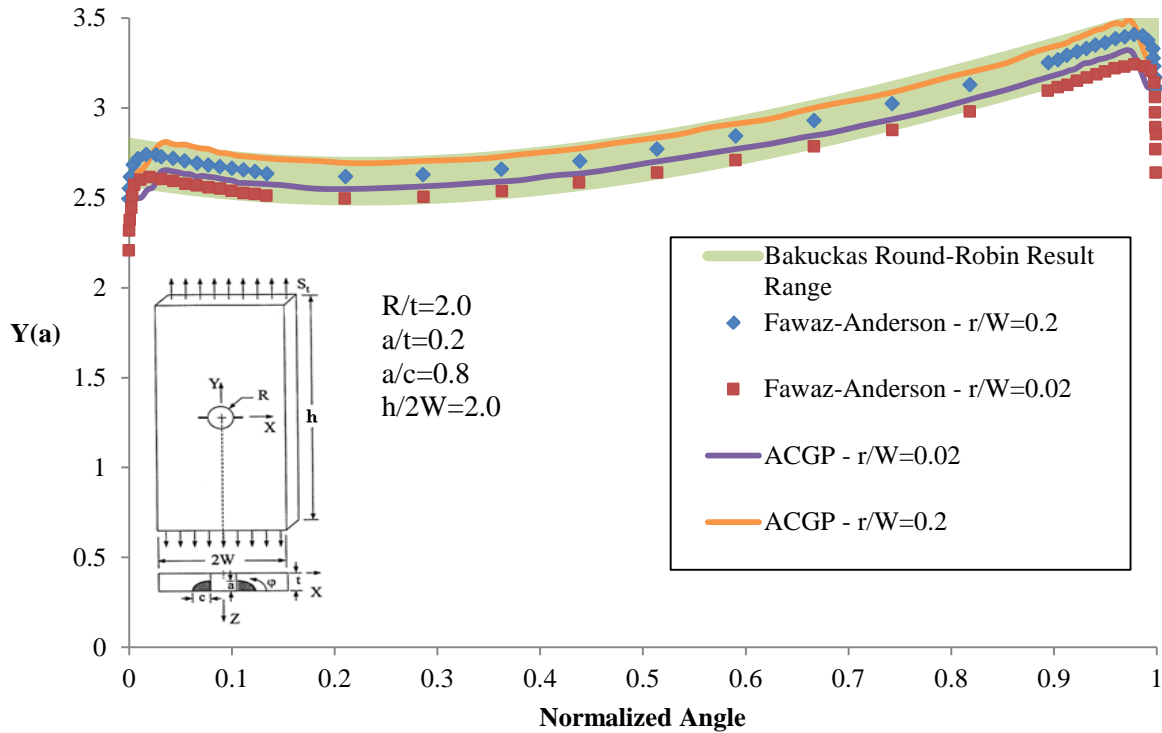


Figure 4.4: Geometric factors vs. normalized angle along the crack face for two symmetric corner cracks emanating from a hole in a plate.

A comparison of the results from the ACGP, Bakuckas and Fawaz and Andersson show that the ACGP can accurately predict SIFs for an elliptical crack emanating from a hole along the entire crack front and account for width effects.

4.1.2 A Single Corner Crack Emanating from a Hole in a Plate

Problem Description

The ability of the ACGP to predict accurate SIFs for a more complex crack shape was evaluated through comparison with published numerical SIF predictions for a shallow corner crack emanating from a hole in a plate. It is important for SIF predictions to be accurate for corner cracks that intersect free

surfaces at shallow angles to provide confidence in the ability and range of the ACGP with respect to different crack geometries. The predicted results from Fawaz and Andersson [9] were used to validate the ACGP SIF predictions. Fawaz and Andersson used the same method described in section 4.1.1 to predict SIF values along the crack front for a shallow corner crack with $a/c=0.1$ emanating from a hole in a plate and observed the same accuracy with an error less than 1% [9]. The geometry and geometric ratios Fawaz and Andersson employed are shown in Figure 4.5. A radius (r) of 2 units and a Poisson's ratio (ν) of 0.3 were used.

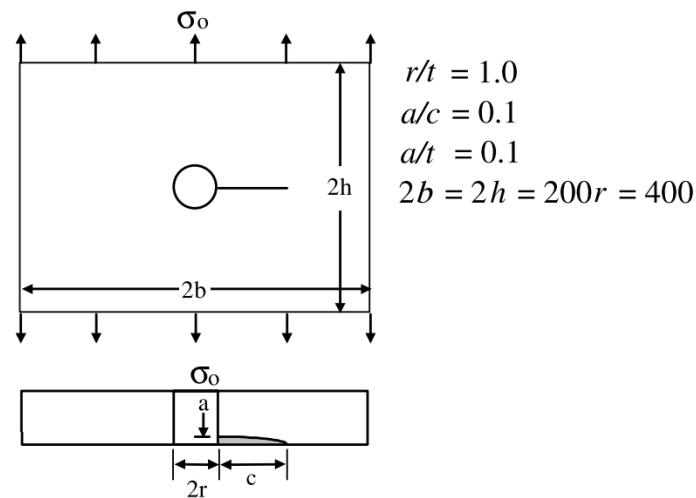


Figure 4.5: Geometry, material and applied stress used to generate SIF results from Fawaz and Andersson [13].

To recreate the tests from the published results, version 1 of the ACGP was employed which models a corner crack emanating from a hole in a plate. One iteration of the ACGP was run with the described geometry to generate SIF results and, as in the study from 4.1.1 the growth model was not utilised. The same Poisson's ratio and geometry ratios from Figure 4.5 were input. A radius of 50.8mm was chosen to represent the radius of the hole and the remaining geometric values were determined using the specified ratios. The crack front was fit with a quarter ellipse with a horizontal radius of 50.8mm and a vertical radius of 5.08mm. The modulus of elasticity and loading were not specified by Fawaz and Andersson but were both set to 0.0069MPa to create a test similar to that done in 4.1.1. The c_{tr} , m_{rat}

and trans mesh values were set to 0.018, 0.05 and 0.02 respectively to produce a sufficient number of elements along the crack front and in the global mesh. The mesh refinement was required so that the curve of the crack front at the intersection with the plate surface could be captured, as seen in Figure 4.6. The geometric, modelling and meshing parameters used in the ACGP are shown in Figure 4.7.

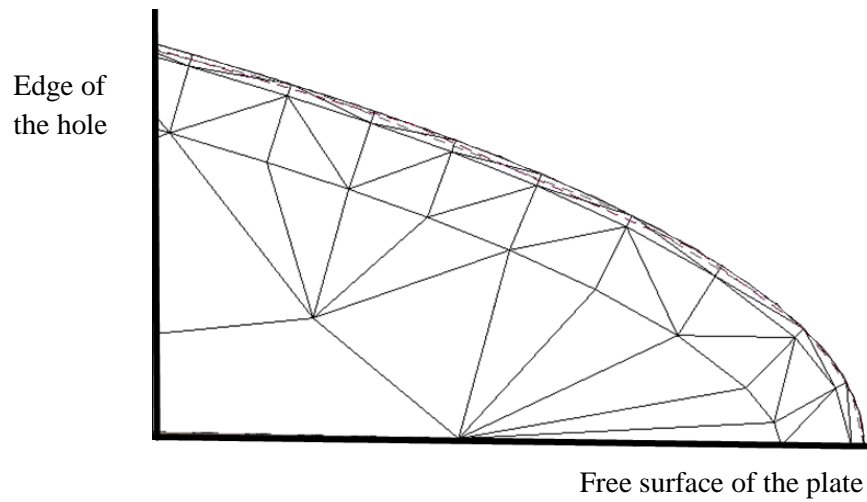
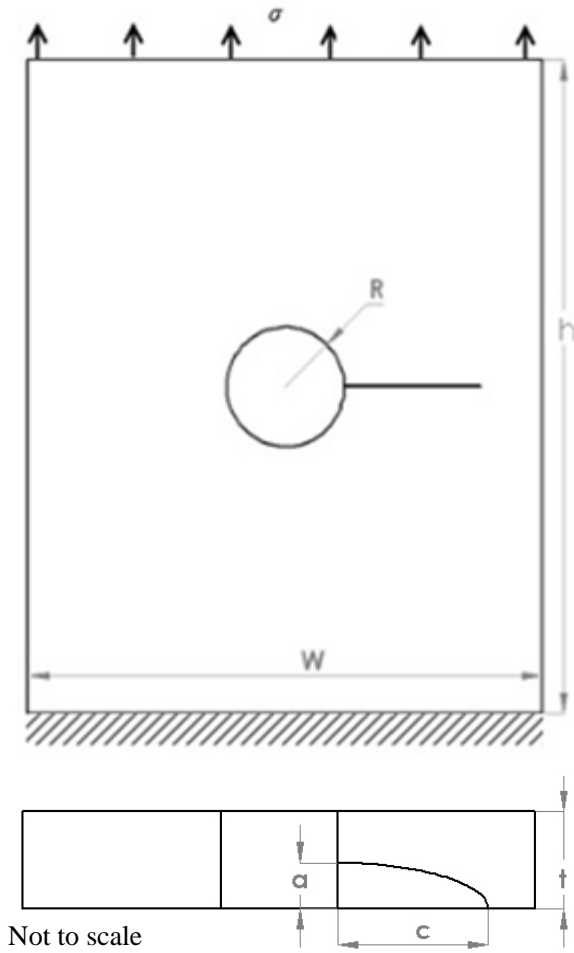


Figure 4.6: Mesh of the shallow corner crack at the intersection of the lower surface



Model Parameters

Stress	0.0069MPa
Stress Ratio	0
E	0.0069MPa
v	0.3

Geometric Parameters

h	10160mm
w	10160mm
a	5.08mm
c	50.8mm
R	50.8mm
t	50.8mm

Meshing Parameters

ctrat	0.018
mrat	0.05
trans	0.02
ipr	0.2
tTotalrat	0.15
tOrat	0.15

Figure 4.7: Model, geometry and meshing parameters for a corner crack emanating from a hole in a plate.

The SIF results were compared in the form of beta factors and normalised parametric angles to display the magnitude and their position over the crack front respectively. The beta factors, β , were calculated using

$$\beta = \frac{\Delta K_I}{\sigma\sqrt{\pi c}} \quad (33)$$

Results and Discussion

The ACGP beta factor predictions over the crack front for a shallow corner crack match the Fawaz and Andersson predictions to within 2.5% as seen in Figure 4.8. The maximum difference is seen at the intersection with the hole.

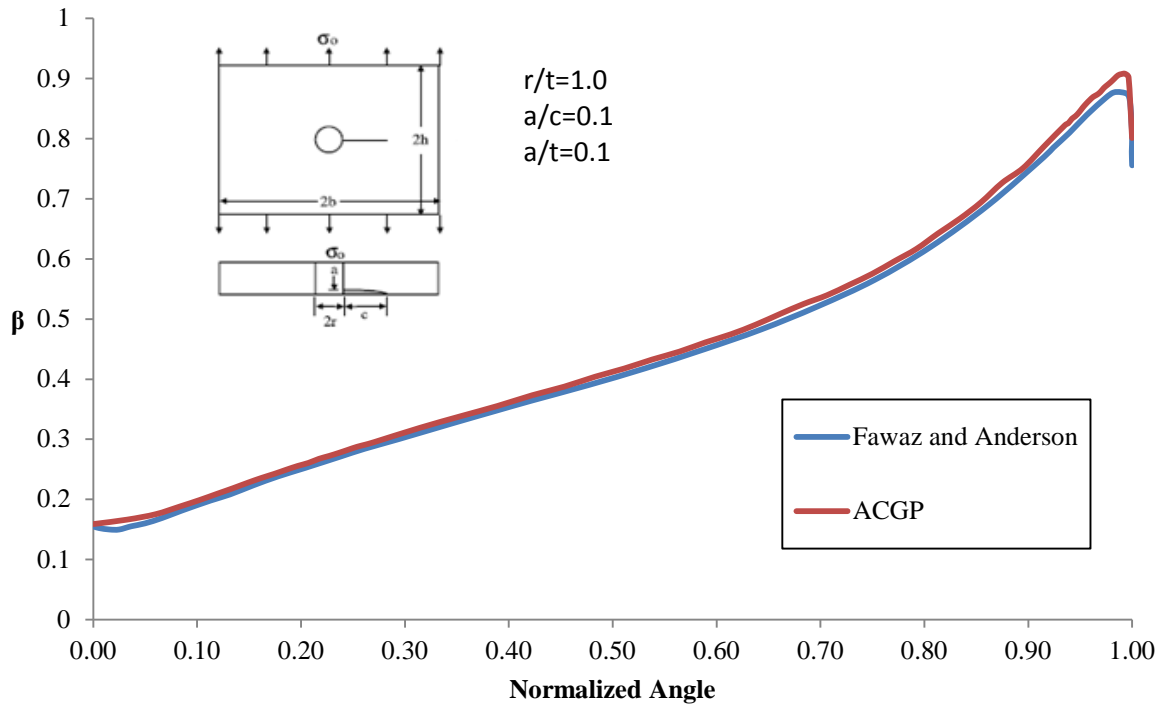


Figure 4.8: Beta factor results for a corner crack emanating from a hole in a plate.

The fact that the ACGP predictions match the published data indicates that SIF predictions over the crack front are accurate for mode 1 corner cracks in thin plates. The predictions near the free surfaces are an area of concern when predicting SIFs and Fawaz and Andersson were able to accurately predict the SIFs due to the highly graded mesh and large number of degrees of freedom in their model. The ACGP results were able to match the Fawaz and Andersson results well. The ACGP accurately predicts SIFs along the entire crack front up to the free surfaces which is important because the maximum SIF values are found near the free surfaces. Many crack growth codes, such as AFGROW assume that the maximum SIFs will occur at specific angles based on engineering judgement or experimental evidence, and only

extract SIFs at this location. Using the ACGP this assumption is no longer needed, as the entire SIF distribution along the crack front can be accurately determined.

The results from section 4.1 confirm the accuracy of predicted SIF values over an entire crack front for cracks in thin plates, calculated utilizing the finite element model employed by the ACGP.

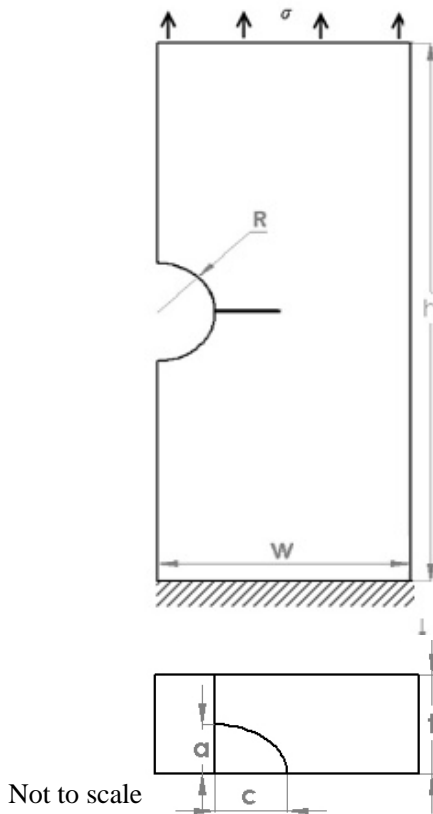
4.2 Accuracy of the Crack Growth Model

Problem Description

The accuracy and validity of the growth model employed in the ACGP was determined through a comparison with AFGROW. A technique was employed, using information output by AFGROW, to produce crack shape and fatigue life data without creating a finite element model. This technique was used in order to isolate and confirm the accuracy of only the fatigue life and growth calculations from the ACGP. AFGROW is a widely accepted tool that is used for fatigue and damage tolerance analysis in the aerospace industry. It enables fatigue life and crack growth predictions to be generated quickly using SIF solutions and growth technique that projects two points and fits the crack front to an ellipse. The SIF solutions for each crack front are calculated using handbook equations or gathered from tabulated data. AFGROW enables solutions to be generated quickly, but the geometries that can be solved are limited and the assumptions made do not fully capture the distribution of the SIFs along the crack front.

The first step in utilizing this technique was to generate crack growth results using AFGROW. The geometry was chosen based on the available solutions in AFGROW for a corner crack emanating from a notch. The width of the plate (w) was 127mm, the thickness (t) was set to 7.94mm, the radius of the notch (R) was set to 7.94mm, and the crack was fit to a quarter ellipse with a horizontal radius of 1.63mm and a vertical radius of 1.63mm. The material was chosen to represent polymethyl methacrylate (PMMA) with data provided by Grandt and Macha [45]. The modulus of elasticity (E) was set to 3.10GPa, the Poisson's ratio (ν) was set to 0.365, the Paris exponent (m) was set to 6.0954 and the Paris

constant (C) was set to 1.92×10^{-6} m/cycle, $\text{MPa}\sqrt{\text{m}}$. The loading was set to 4.76MPa with a stress ratio of 0. The geometry, material and loading parameters used the generate AFGROW results are shown in Figure 4.9.



Model Parameters

Stress	4.76MPa
Stress Ratio	0
E	3.10GPa
ν	0.365
Paris m	6.095
Paris C	1.92×10^{-6} m/cycles, $\text{MPa}\sqrt{\text{m}}$

Geometric Parameters

h	190.5mm
w	127mm
a	1.63mm
c	1.63mm
R	7.94mm
t	7.94mm

Figure 4.9: Model and geometry parameters for a corner crack emanating from a notch in a plate used in AFGROW.

With AFGROW results generated, the coordinates of the a and c crack locations, their respective ΔK values (ΔK_a and ΔK_c) and the maximum crack increment (Δa) for the first AFGROW iteration were then input into the ACGP. The Paris parameters were input into the ACGP and the growth model was run to generate new a and c coordinates. The ΔK values for the second AFGROW iteration were then input into the ACGP for the new a and c coordinates along with the maximum growth increment and the ACGP growth model was run to generate the next set of a and c coordinates. This process was repeated for 18 iterations and is described in Figure 4.10.

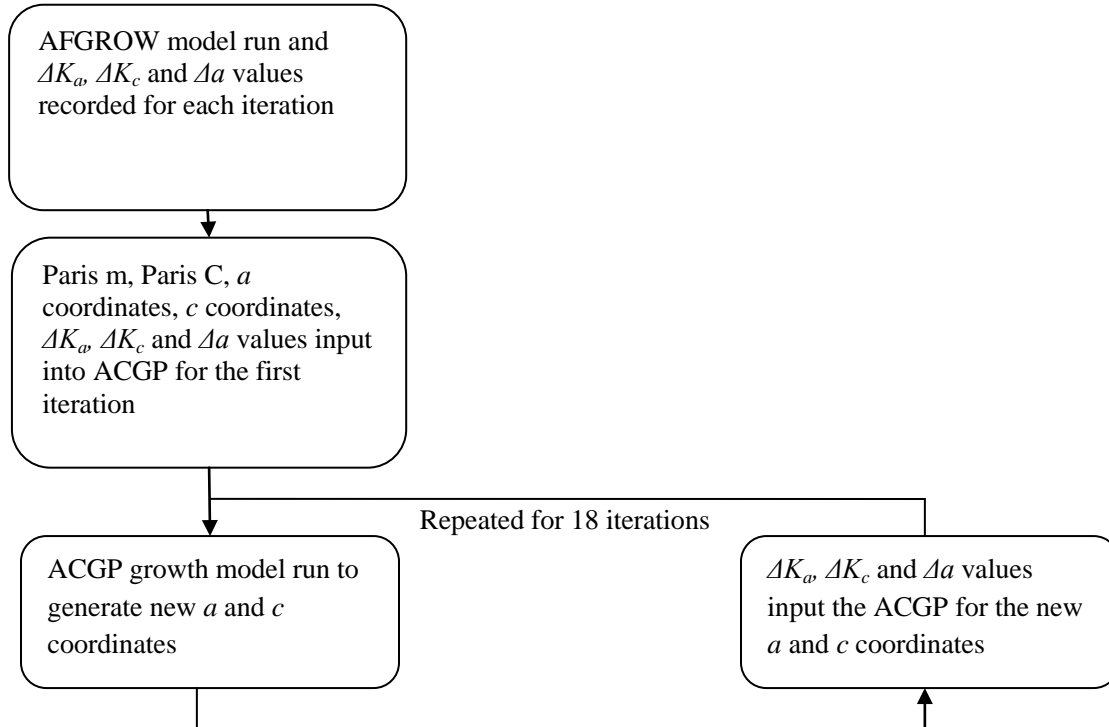


Figure 4.10: Process used to generate growth results from the ACGP.

This technique was used to determine the accuracy of the fatigue life and growth calculations used in the ACGP. If the ACGP growth calculations were incorrect, the crack shapes predicted by the ACGP and AFGROW would not match and if the ACGP fatigue calculations were incorrect, the fatigue life predictions from the ACGP and AFGROW would not match.

Results and Discussion

The accuracy of the growth calculations used in the ACGP were confirmed by analysing the a versus c results for all 18 iterations from the ACGP and AFGROW. The a versus c plot displays the relationship between the crack distances along the surface for each iteration. From the matching data seen in Figure 4.11 it can be concluded that the growth calculations used in the ACGP growth model have been correctly implemented.

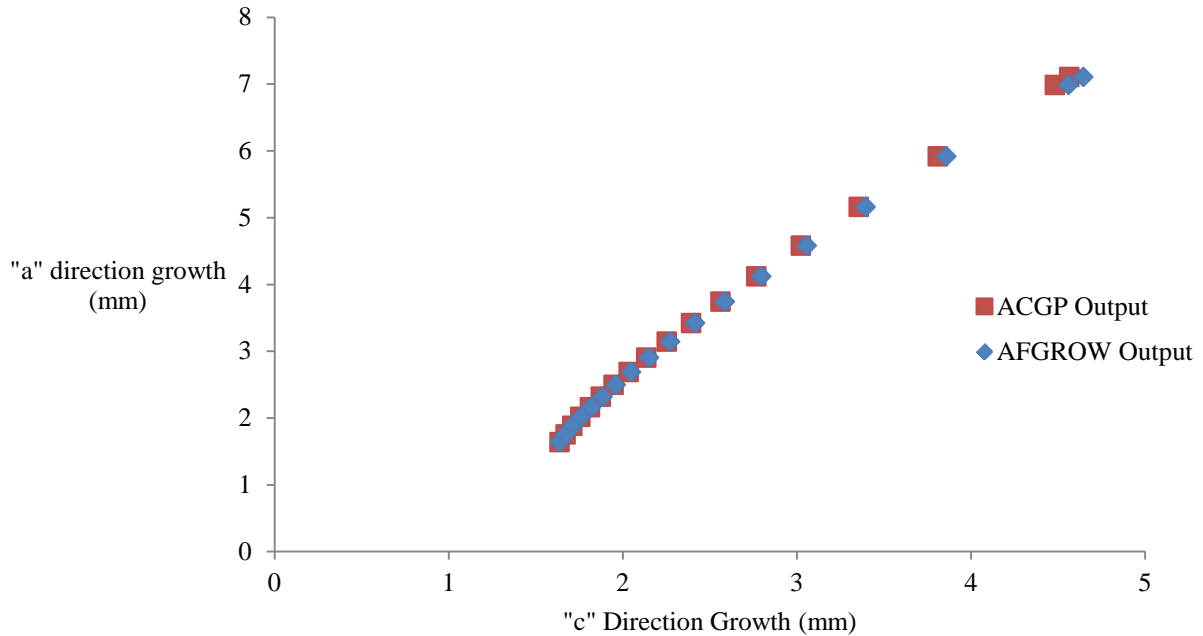


Figure 4.11: Crack length a vs. crack length c predicted from the ACGP and AFGROW.

To confirm the accuracy of the fatigue life calculations used in the ACGP, a comparison of the a and c growth versus the number of cycles was analysed. The comparison found that the number of cycles for growth of both a and c corresponded to the same number of cycles calculated by AFGROW. It is important to note that in AFGROW the number of cycles that is output is based on complete blocks of cycles and not individual cycles. For example, if the output is set to blocks of 400 cycles, AFGROW may calculate 430 cycles but it will round down to 400 cycles. The number of individual cycles that AFGROW internally calculates was found by dividing the growth increment Δa by the outputted crack growth rate provided by AFGROW for each iteration. The ACGP fatigue calculations were compared to both. The blocks of cycles (output) and the individual cycles calculated by AFGROW (internal) are shown in Figure 4.12. The ACGP results match the internal AFGROW results and confirm that the fatigue calculations used in the ACGP are accurate.

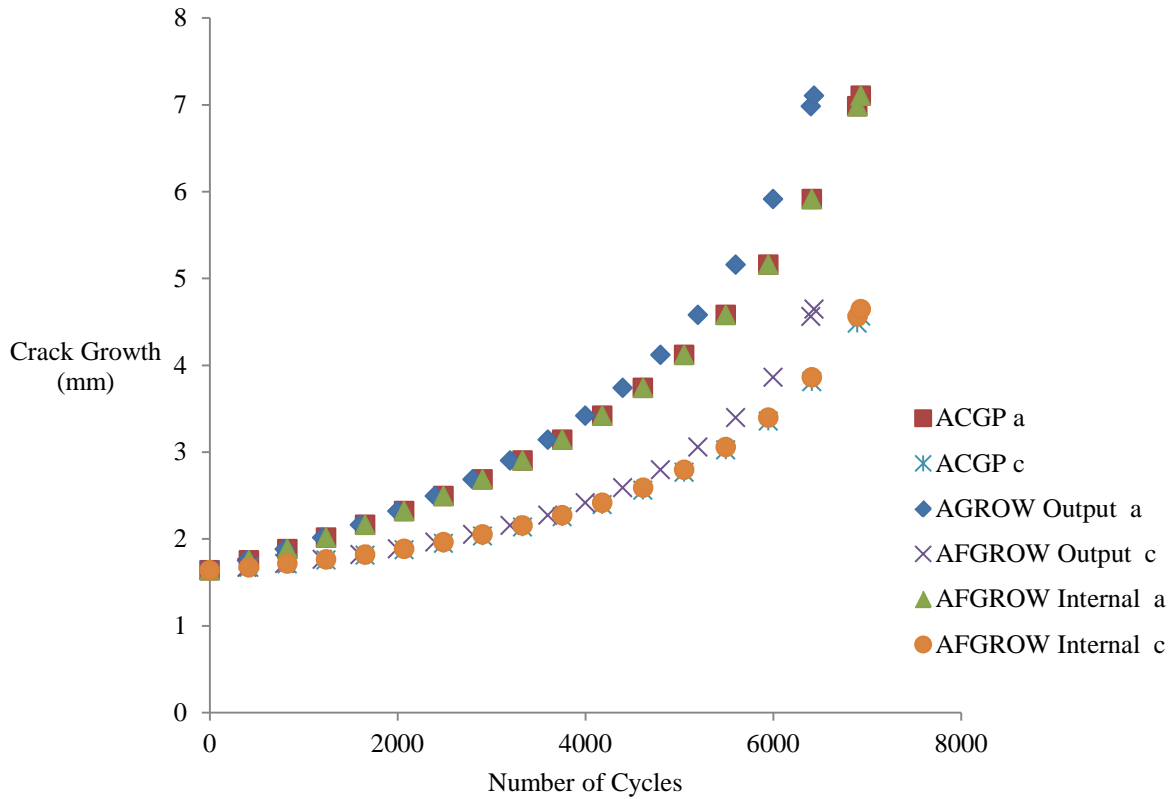


Figure 4.12: Crack growth vs. number of cycles comparing the ACGP to AFGROW.

The results from section 4.2 confirm the accuracy of the fatigue life and crack growth equations used in the ACGP.

4.3 ACGP Comparison with AFGROW Predictions

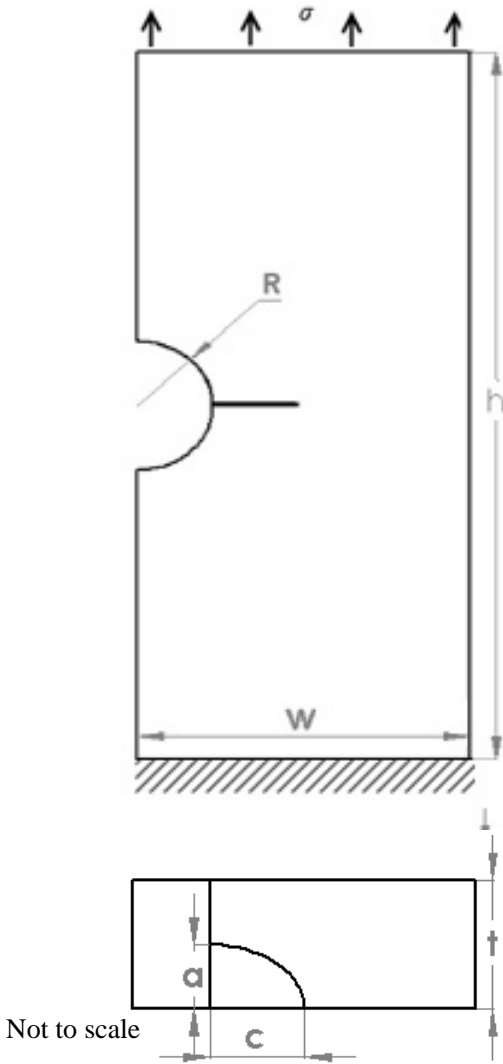
Problem Description

With the accuracy of the SIF predictions and growth model confirmed, a complete ACGP analysis was performed and compared to the AFGROW results from section 4.2. The complete ACGP analysis involved input of the initial crack shape and allowing the program to complete a full run to predict fatigue life and crack shape.

To predict crack growth AFGROW employs a similar method as Newman and Raju [10], where the growth at two points along the crack front is predicted from SIF solutions and an ellipse is fit to the

two points. For a corner crack emanating from a notch in a plate, AFGROW predicts the growth at the lower surface and at the notch surface and fits an ellipse to these two points. The SIF solutions that AFGROW uses to calculate growth comes from crack front solutions generated by Newman [46] where the SIF value used to grow the crack at the surface and at the notch is taken at 2.5° and 87° along the crack front respectively [47]. The AFGROW solutions are only valid for geometries with $r/w=1/16$, limiting the range of tests that can be performed. The AFGROW solutions are valid comparisons for the ACGP comparison test done in this section which also uses an r/w ratio of $1/16$.

Version 2 of the ACGP was employed to model a corner crack in a notch. The ACGP ran for 31 iterations with a maximum growth increment of $\Delta a_{\max}=0.76\text{mm}$ to grow the crack to a final length of 4.06mm along the surface. The same model and geometric parameters from section 4.2.1 were input, as shown in Figure 4.13. The initial crack front was fit with a quarter ellipse with a horizontal radius of 1.64mm and a vertical radius of 1.64mm . $ctrat$, $mrat$ and $trans$ mesh values were of 0.04 , 0.2 and 0.02 respectively were used to produce a sufficient number of elements along the crack front and in the global mesh.



Model Parameters

Stress	4.76MPa
Stress Ratio	0
E	3.10GPa
ν	0.365
Paris m	6.095
Paris C	1.92×10^{-6} m/cycle, MPa \sqrt{m}
Δc_{max}	0.076mm

Geometric Parameters

h	190.5mm
w	127mm
a	1.64mm
c	1.64mm
R	7.94mm
t	7.94mm

Meshing Parameters

ctrat	0.04
mrat	0.2
trans	0.02
ipr	0.2
tTotalrat	0.15
tOrat	0.15

Figure 4.13: Model, geometry and meshing parameters for a corner crack emanating from a notch in a plate.

Results and Discussion

Plots of crack length along the bore and along the surfaces (a versus c plot) can be seen in Figure 4.14, and the change in aspect ratio as the crack grows up the notch surface a (aspect ratio versus a plot), can be seen in Figure 4.15. These plots enable the overall crack shape to be quantified as well as determine whether or not the shapes predicted by each program match. The growth pattern observed in these plots is not dependant on the magnitude of ΔK along the crack front, but rather the difference between the ΔK values at the free surfaces. The growth pattern predicted by the ACGP is within 1.5% of the AFGROW predictions.

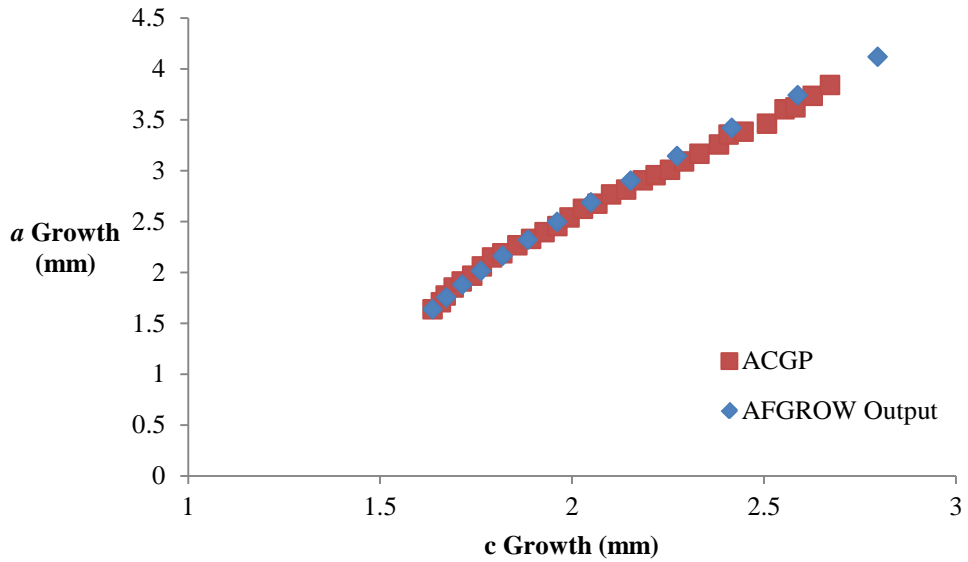


Figure 4.14: a vs. c comparing the ACGP to AFGROW.

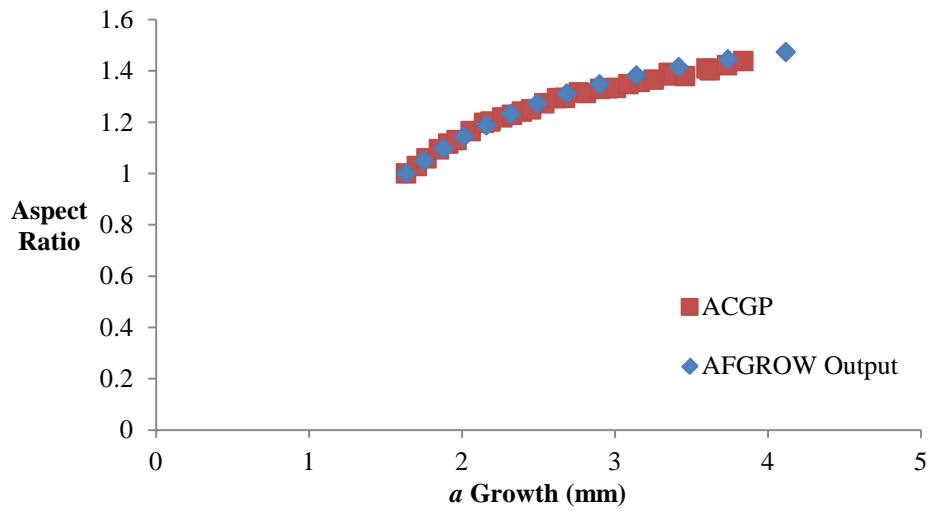


Figure 4.15: Aspect ratio vs. a comparing the ACGP to AFGROW.

A plot of crack growth versus number of cycles is shown in Figure 4.16. When comparing crack growth after approximately 6400 cycles, the AFGROW results predicted 93% further growth along the hole, a , and 80% further growth along the surface, c . From these results it was observed that the AFGROW predictions are much more conservative when compared to the ACGP results for the tested geometry.

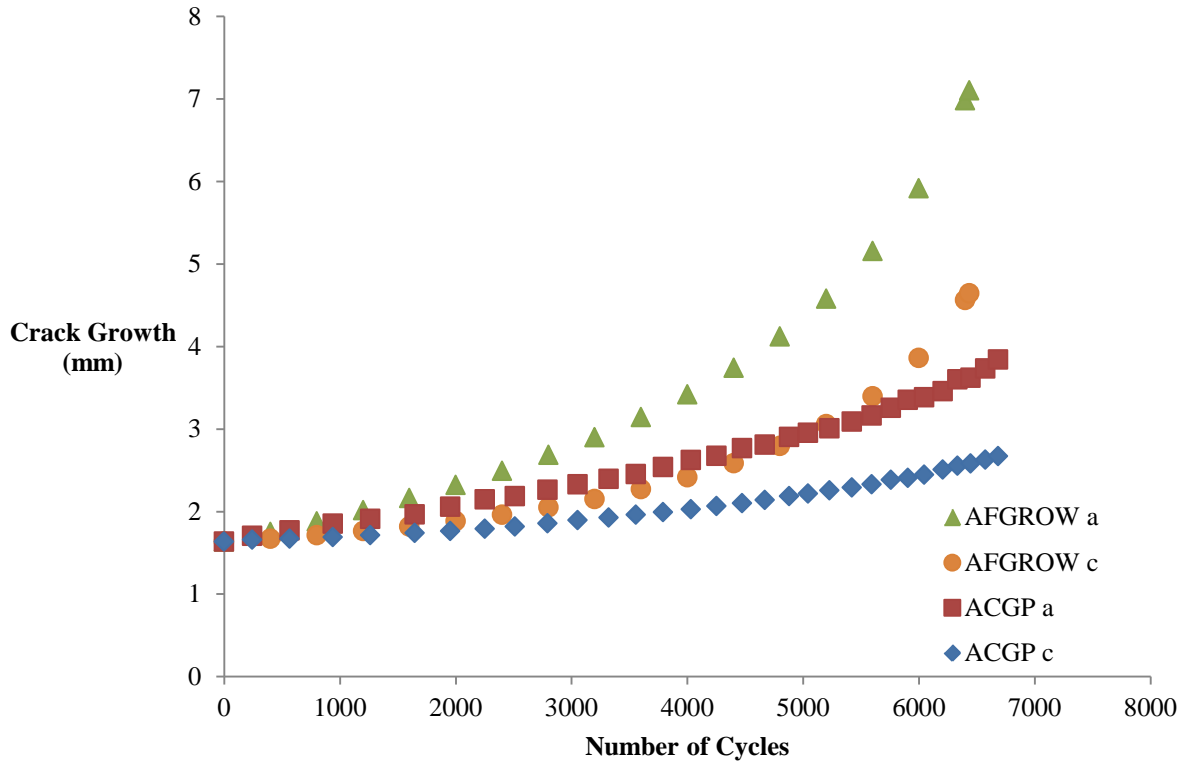


Figure 4.16: Crack growth vs. number of cycles comparing the ACGP and AFGROW.

To determine why the fatigue life predictions varied between the two programs, an analysis comparing the maximum ΔK values used by each program for each crack front was completed, as seen in Figure 4.17. In AFGROW the maximum ΔK is taken at 87° along the crack front as stated in the User's Guide [47]. For the ACGP the maximum ΔK was taken at the location along the crack front where ΔK was largest. The maximum ΔK values used in both programs were the same for the first iteration but after the first iteration the maximum ΔK values were an average of 5% higher in the AFGROW results.

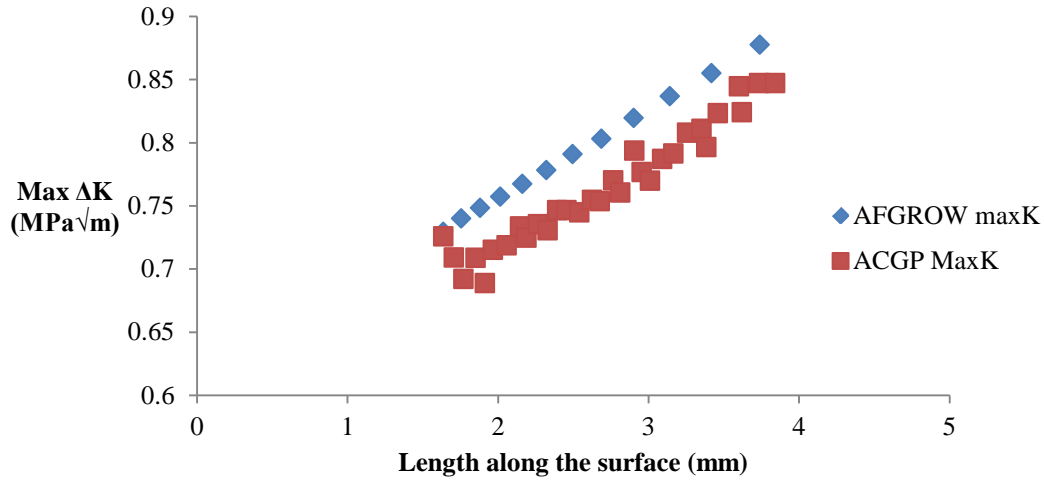


Figure 4.17: Max ΔK vs. length along the surface comparing the thesis program to AFGROW.

The difference in the ΔK used is what caused the large difference in fatigue life predictions.

Figure 4.18 shows that when the ΔK values were manually increased by 5% for a full ACGP run, the fatigue life matched the AFGROW predictions.

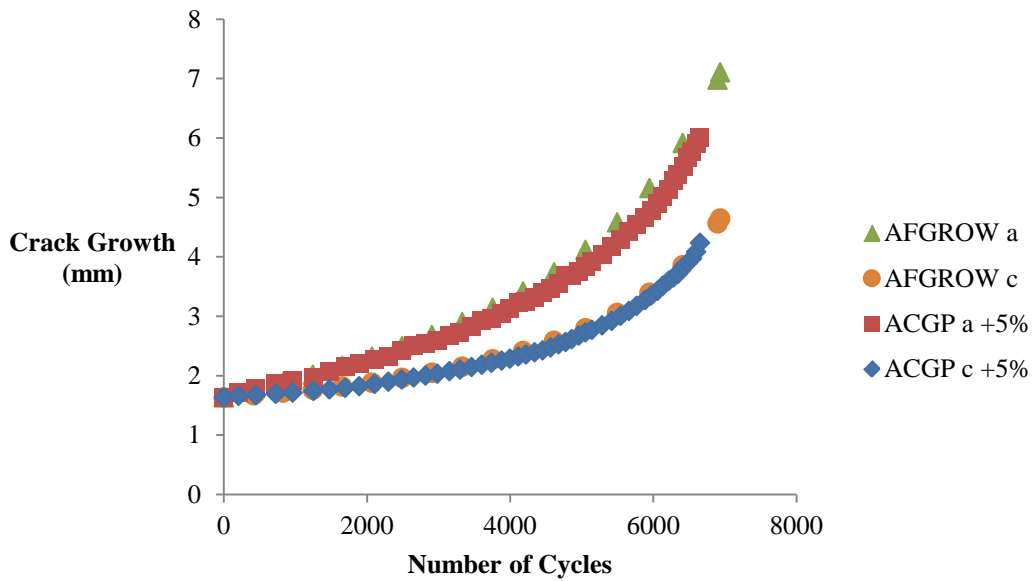


Figure 4.18: Crack growth vs. number of cycles comparing the ACGP (ΔK increased by 5%) and AFGROW.

The difference in the values of ΔK may be due to the fact that AFGROW models each crack front as an ellipse whereas the ACGP uses a spline. An analysis was conducted to determine the effects of ellipse versus spline geometry on the distribution and magnitude of SIF values over the crack front. 4 iterations of the test completed in this section were considered, as shown in Figure 4.19. For each of the iterations an ellipse was fit to the spline crack front and one iteration of the ACGP was run to predict SIF values over the elliptical crack front. The ellipse fit results in a crack geometry with more curvature than the spline and a 90° intersection at the free surfaces.

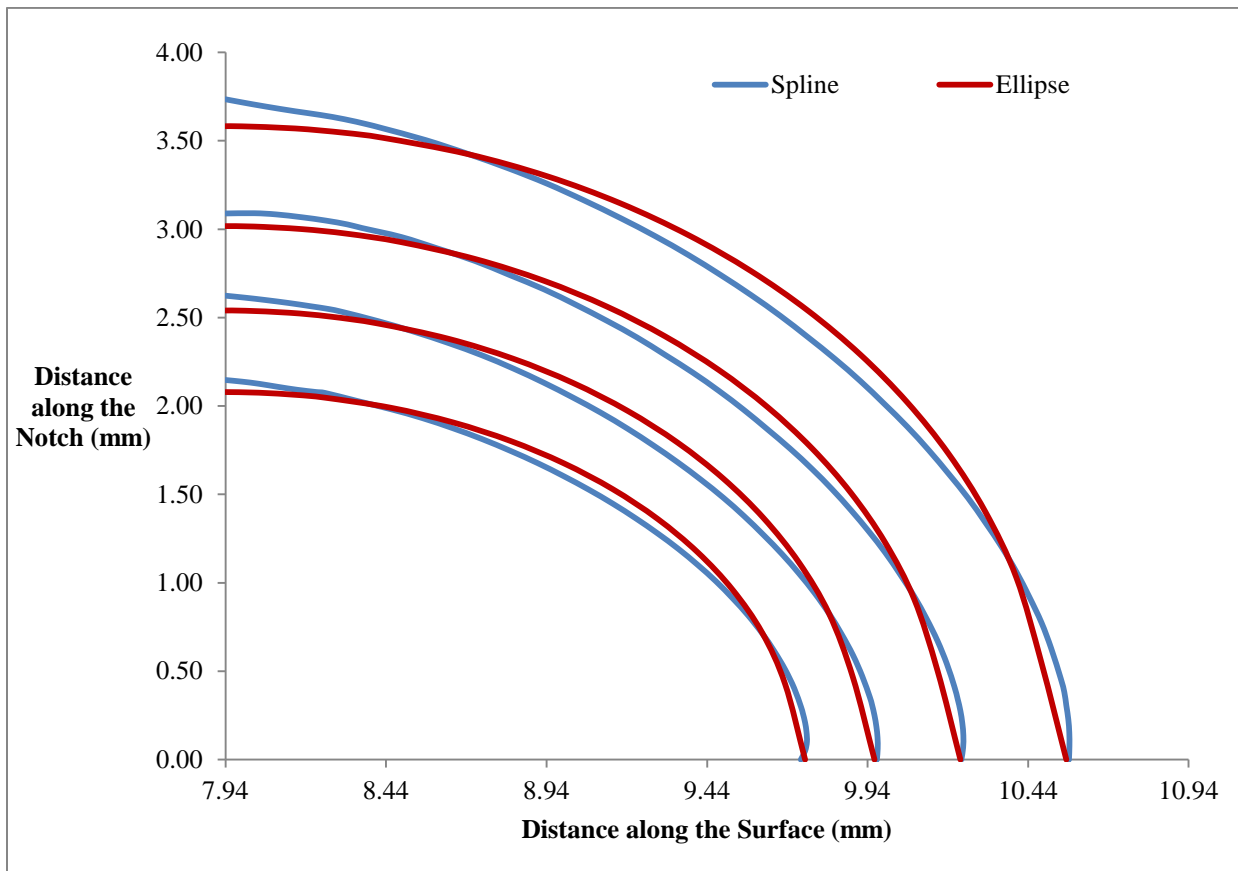


Figure 4.19: Geometry comparison between a spline and ellipse fit.

The ΔK distributions along the crack front for each crack shape were predicted and are shown in Figure 4.20 with the spline results shown in blue and the ellipse results shown in red.

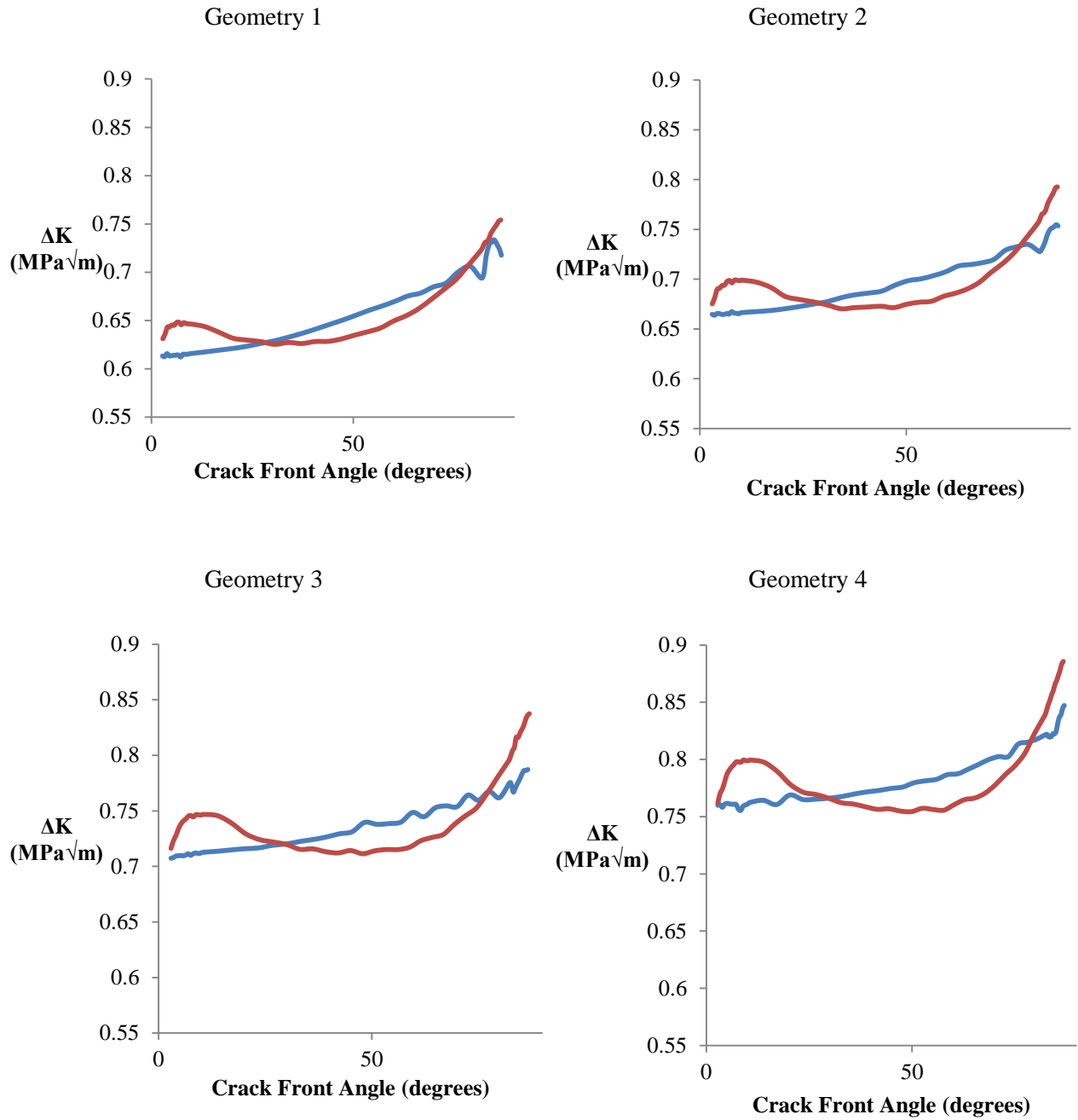


Figure 4.20: ΔK vs. crack front angle for four different crack lengths. ΔK values using an ellipse fit are shown in red and using a spline fit are shown in blue

Even though the ellipse has a relatively similar geometry to the spline as seen in Figure 4.19, Figure 4.20 shows a maximum ΔK difference over the crack front, averaged from the four geometries, of 4.7%. Table 4.1 shows the maximum ΔK values and their percent difference for each set of geometry.

Table 4.1: Comparison of maximum ΔK values for four sets of geometry

	Geometry 1	Geometry 2	Geometry 3	Geometry 4
Spline (MPa \sqrt{m})	0.734	0.755	0.787	0.847
Ellipse (MPa \sqrt{m})	0.754	0.793	0.837	0.886
Percent Difference (%)	2.72	5.03	6.35	4.60

The results observed from the spline versus ellipse geometry test indicate that the predicted fatigue life differences found between the ACGP and AFGROW may be due to the fact that the crack fronts are fit to different geometries.

The strength of AFGROW is that it enables crack growth and fatigue life predictions to be calculated very quickly. It took AFGROW under 5 seconds to generate the results in this section, while it took the ACGP 11 hours. The downfall of AFGROW is that the stresses along the entire crack front are not accurately accounted for because the crack is forced to grow as an ellipse, which is not representative of its actual shape. This leads to inaccuracies in fatigue life predictions because, as observed, if the K values differ by only 5% the final growth predictions can differ by over 90%.

4.4 ACGP Comparison with Experimental Results

Problem Description

Fawaz and Hill concluded that the best method for validating crack growth results is to compare predicted fatigue life and crack growth to experimental results [42]. This method of validation was completed by comparing results predicted by the ACGP to experimental results provided by Desnoyers and Liao [43] from NRC. Desnoyers and Liao conducted a fatigue test on a Single Edged Notch Tension (SENT) coupon machined from an aluminum 7075-T73 forging. A stress ratio of 0.8 was used with a maximum loading of 275MPa. The crack initiated as a centre crack along the notch after 649560 cycles. The crack then propagated to a corner crack then through the thickness with the crack extending through the coupon after 799169 cycles. An image of the marker band profile taken from the report is shown in Figure 4.21. The x axis represents the notch face and the y axes represent the upper and lower surfaces of the coupon. The initial crack geometry that was input into the ACGP is indicated in Figure 4.21 and the values are located in Appendix A.

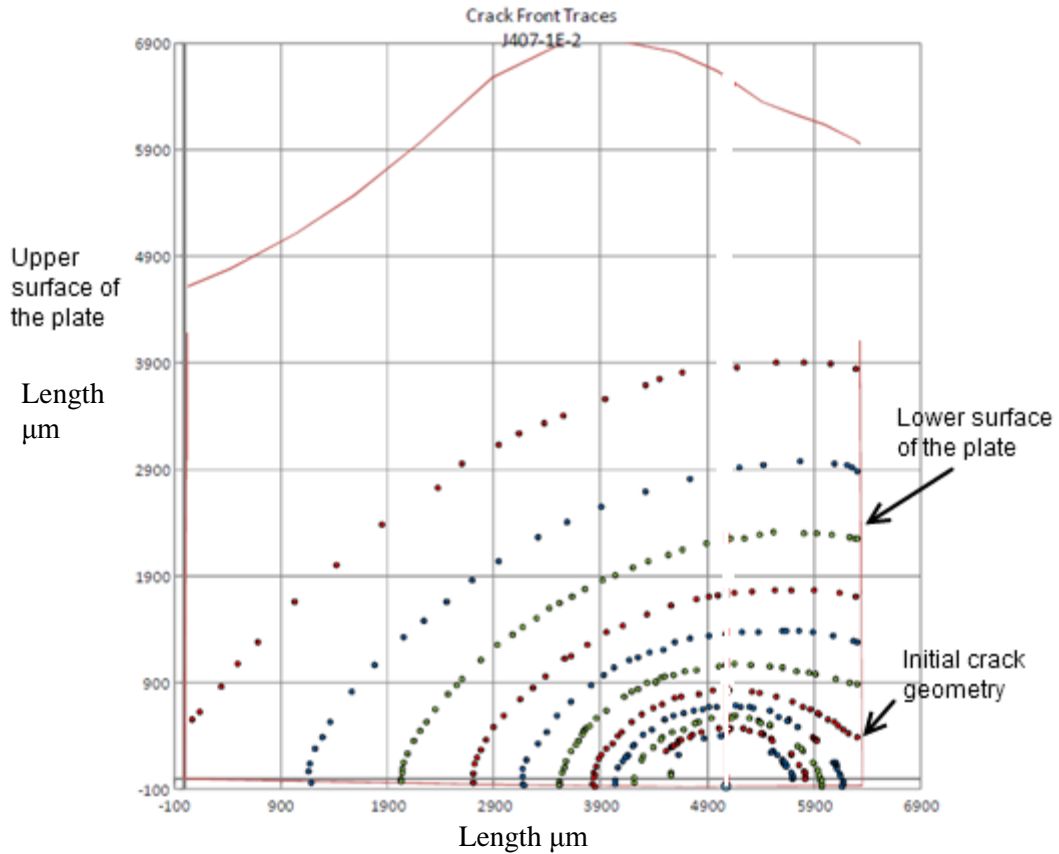
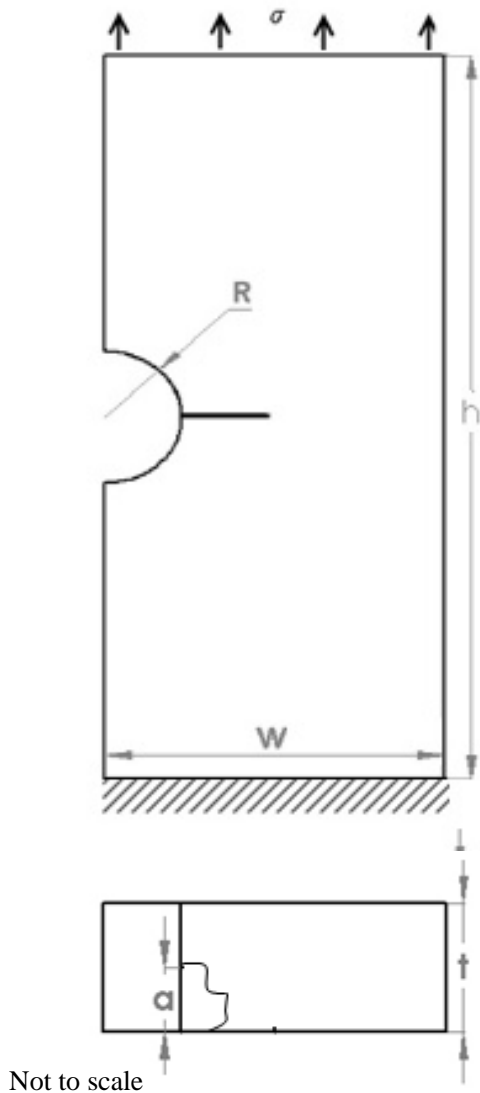


Figure 4.21: Marker band profile from experimental results [43].

Version 2 of the ACGP was employed to model a corner crack in a notch. The geometry of the plate was taken from the report by Desnoyers and Liao [43] and the initial crack geometry was digitized from Figure 4.21. The Paris values were calculated from fatigue life results provided in the report. The ACGP ran for 53 iterations with a maximum growth increment of 0.076mm to reach the position of the final marker band shown in Figure 4.21. $ctrat$, $mrat$ and $trans$ mesh values of 0.01, 0.1 and 0.02 respectively were used for the first 9 iterations to enable proper meshing of the complex crack front. After the 9th iteration, the $ctrat$ and $mrat$ values were increased to 0.03 and 0.2 respectively to allow for faster modelling. A spline fit to 28 points was used to capture the irregular crack front geometry. The geometric, modelling and meshing parameters used in the ACGP are shown in Figure 4.22.



Model Parameters

Stress	275.8MPa
Stress Ratio	0.8
E	70.3GPa
ν	0.33
Paris m	3.449
Paris C	2.74×10^{-12} m/cycle, MPa \sqrt{m}
Δc_{max}	0.076mm

Geometric Parameters

h	203.2mm
w	50.9mm
a	2.55mm
Initial surface length	0.482mm
R	1.70mm
t	6.45mm

Meshing Parameters

c _{tr} rat*	0.01
m _{tr} rat*	0.1
trans	0.02
ipr	0.2
t _{Total} rat	0.15
t ₀ rat	0.15

*c_{tr}rat and m_{tr}rat were increased to 0.03 and 0.2 respectively after the 9th iteration

Figure 4.22: Model, geometry and meshing parameters for a corner crack emanating from a notch in a plate used for the ACGP simulation.

Results and Discussion

To determine how well the ACGP can predict crack shapes when compared with experimental results, seven crack shapes were chosen. The ACGP geometries chosen for the comparison were taken at the iteration where the growth along the surface (c) was a comparable value to the experimental results. From the results displayed in Figure 4.23, the ACGP was able to predict the crack shape evolution very

well with the largest percent difference of 4.9% occurring where the crack intersects the notch. These results are especially encouraging because of the complex initial geometry that was used.

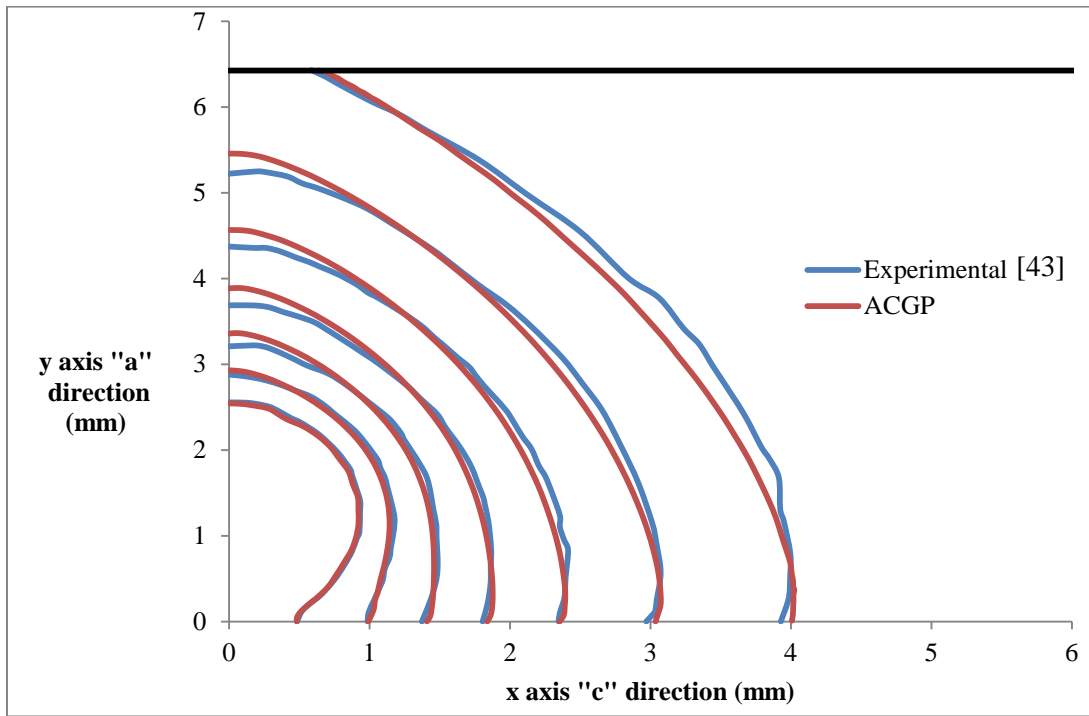


Figure 4.23: Crack shape at different points of propagation.

To determine how well the ACGP can predict fatigue life when compared to experimental results, the crack length along the surface versus the number of cycles is displayed in Figure 4.24.

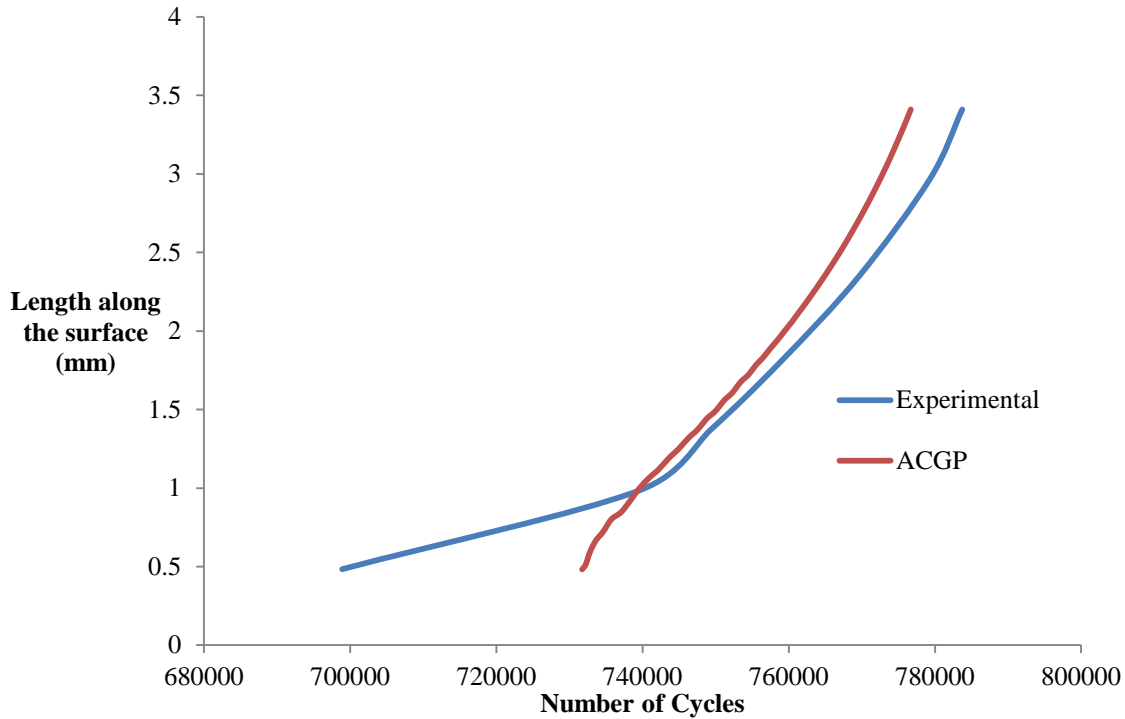


Figure 4.24: Growth along the surface vs. number of cycles comparing the thesis program to experimental results.

In the first stage (0.51mm-0.97mm) the results predicted using the ACGP were much more conservative and do not match the experimental data. For growth in the second stage the analytical results match the experimental results very well, giving a percent difference of 8% in the number of cycles when the crack has grown from 0.97mm to 3.35mm along the surface.

When observing the fatigue life for the experimental results there is a drastic growth rate change between the two stages of growth as seen when observing the slope of the surface growth versus number of cycles plot at 0.97mm. The growth rate along the surface is much lower during the first stage of growth which is unexpected. A gradual change in growth rate is expected for a constant applied stress and stress ratio as was predicted by the ACGP. The sharp change in slope may be due to some unknown parameter changed during the experimental testing such as load magnitude. Another reason for this could be due to residual stresses in the material reducing the growth rate when the crack is small as residual stresses were observed in some of the specimens used when generating the experimental results [48]. Another reason

may be due to the reduction of applied stress caused when marker bands are applied, though this is unlikely because the application of marker bands only occurred in blocks of 100-1000 cycles. Crack closure is an unlikely cause because the stress ratio was set to 0.8 and according to the study by Forth et al. [23] crack closure will not occur during fatigue tests for coupons made from 7075-T73 if the stress ratio is greater than 0.5.

Because the Paris parameters were calculated from results provided by Desnoyers and Liao [43] an analysis was performed to determine if their values caused the ACGP results to be much more conservative for the first stage. This study also showed the sensitivity of the fatigue life predictions on the Paris parameters. To conduct the analysis, two additional ACGP runs were completed with different Paris constants (C) and exponents (m). The C and m values were determined by plotting da/dN versus ΔK results and fitting two different power trend lines based on the number of selected points in the Paris region. The original Paris parameters were calculated using every point in the Paris region from the data provided. The “new 1” and “new 2” Paris parameters were calculated using 10 and 6 points from the Paris region respectively. Table 4.2 shows the Paris parameters used in the original and new ACGP tests. The da/dN versus ΔK plot and power equations for the two new Paris parameters and the original Paris parameters are shown in Figure 4.25.

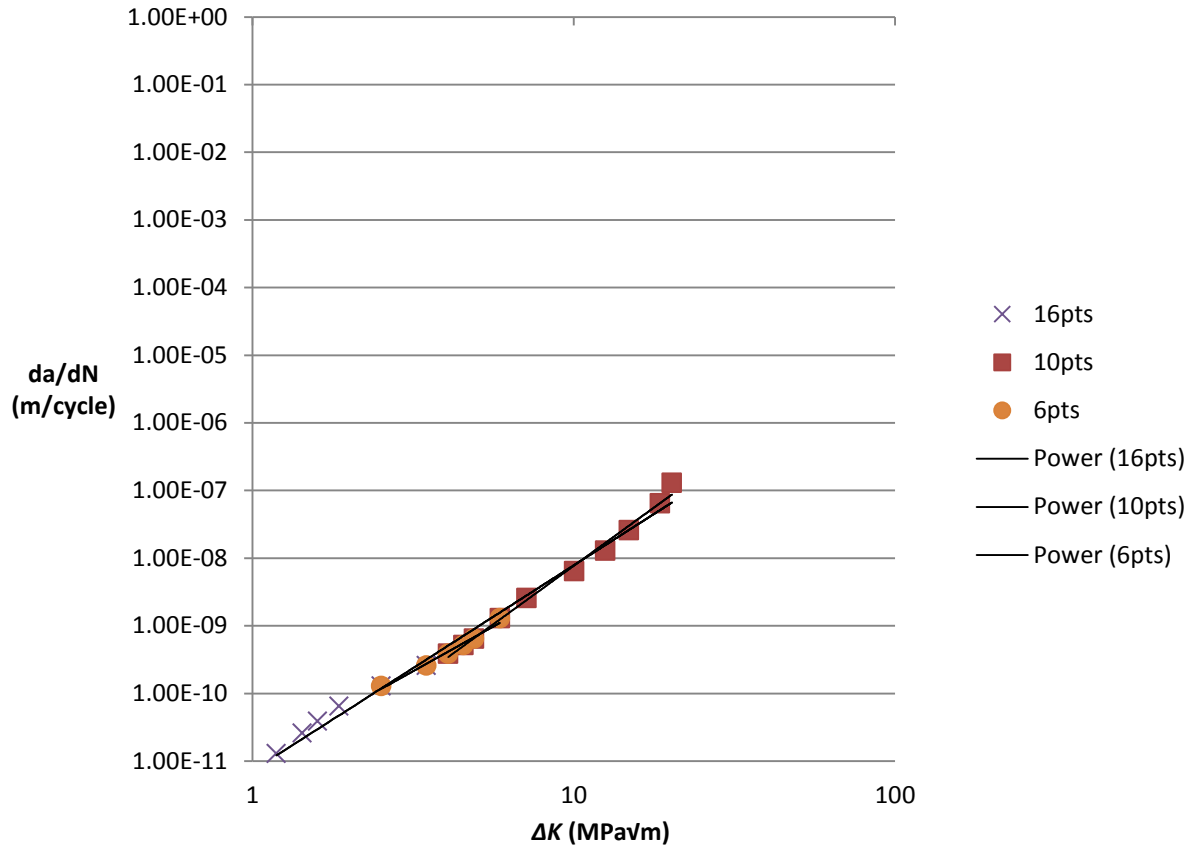


Figure 4.25: da/dN vs. ΔK used to generate Paris parameters.

Table 4.2: Paris parameters for three tests.

Test	Paris Constant C m/cycle, $MPa\sqrt{m}$	Paris Exponent m
Original	2.74E-12	3.448
New 1	7.12E-12	3.037
New 2	1.101E-11	2.647

To determine how the different Paris parameters affected growth, a plot of the aspect ratio versus crack length and plot of crack growth along the notch versus crack growth along the lower surface were created and shown in Figure 4.26 and Figure 4.27 respectively.

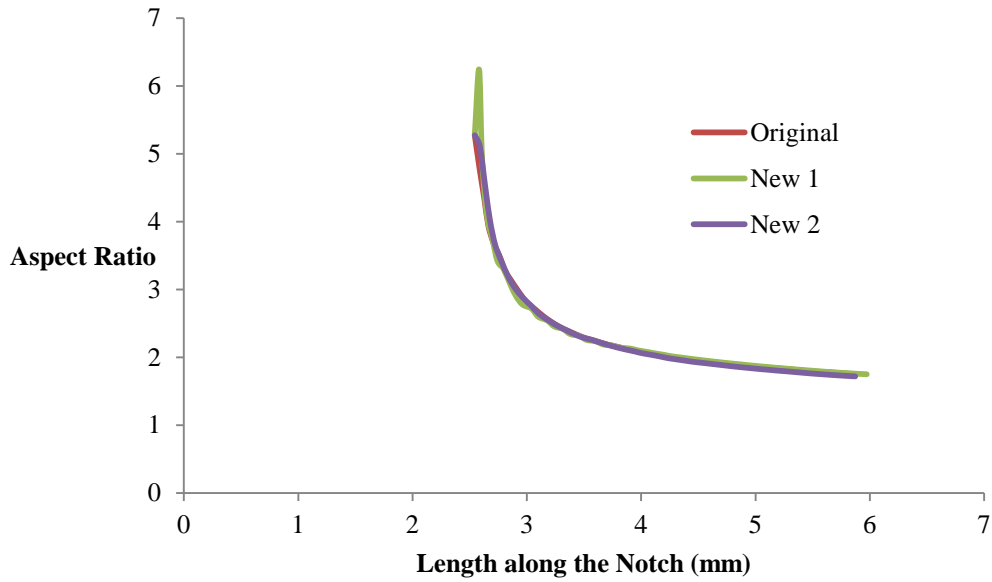


Figure 4.26: Aspect ratio vs. length along the notch for Paris tests.

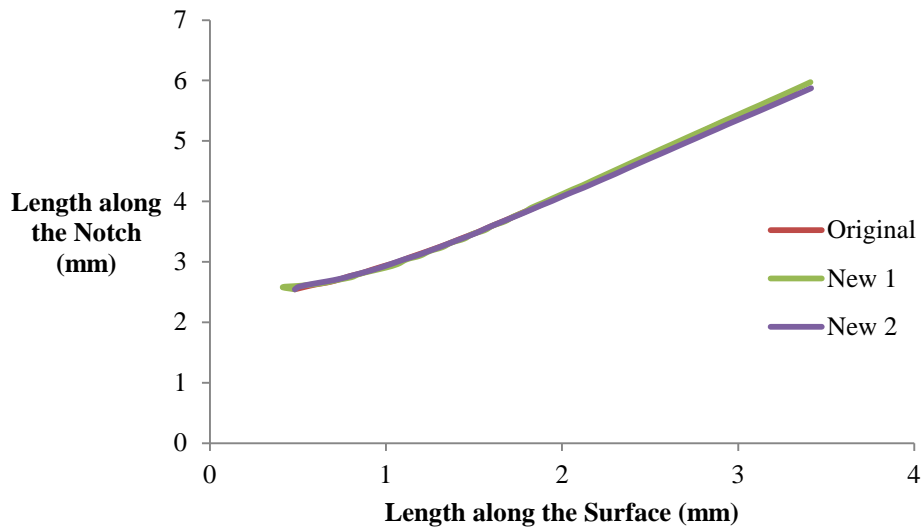


Figure 4.27: Length along the notch vs. length along the surface for Paris tests.

The results indicate that the different Paris parameters have little influence on the shape of the crack front. A C value one order of magnitude lower and a 23% change in m (Original to “new 2”) resulted in no appreciable change in the crack shape. By concluding that Paris parameters do not have a large effect the growth pattern, full ACGP tests do not have to be run to determine the fatigue life calculated from inputting different Paris parameters. The parameters can be changed during post-

processing to determine the new fatigue result. This conclusion was confirmed when the parameters from the original test were changed to the “new 2” Paris parameters during post-processing and compared to the “new 2” full ACGP run. The fatigue results shown in Figure 4.28 differ by a maximum of 2% when the growth along the surface has reached 3.43mm.

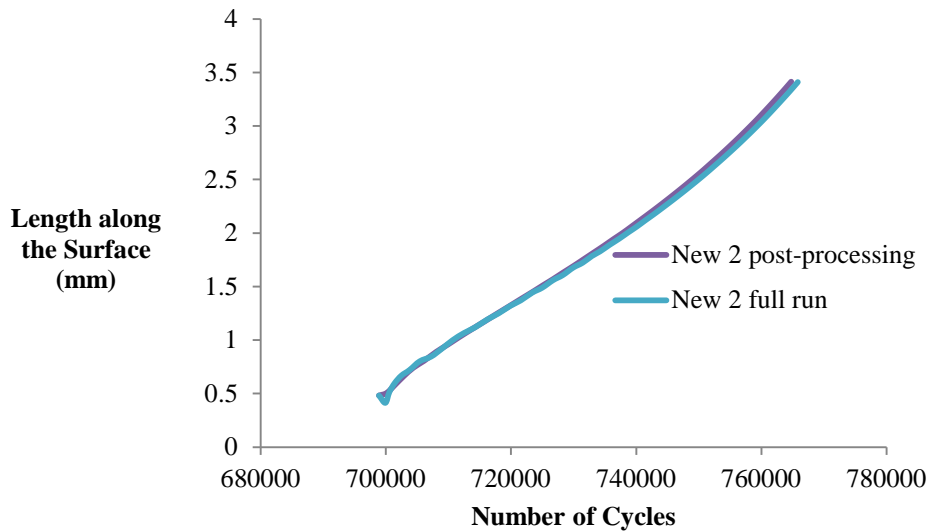


Figure 4.28: Length along the surface vs. number of cycles comparing results generated for a full test and during post-processing.

The fatigue life for each of the three tests was compared to the experimental results in Figure 4.29. The results from the original test and test “new 1” gave a very similar fatigue life predictions, while test “new 2” showed a 51% increase in fatigue life. The original and test “new 1” had a difference in fatigue life of 46% from the experimental results, and test “new 2” had a difference in fatigue life of 26% from the experimental results, once growth along the surface reached approximately 3.43mm. Results from test “new 2” may be less representative of Paris growth because fewer points from the Paris region were taken to generate the parameters. These results show that the discrepancy in the fatigue plot from Figure 4.24 is not due to inaccuracies of determining C and m parameters.

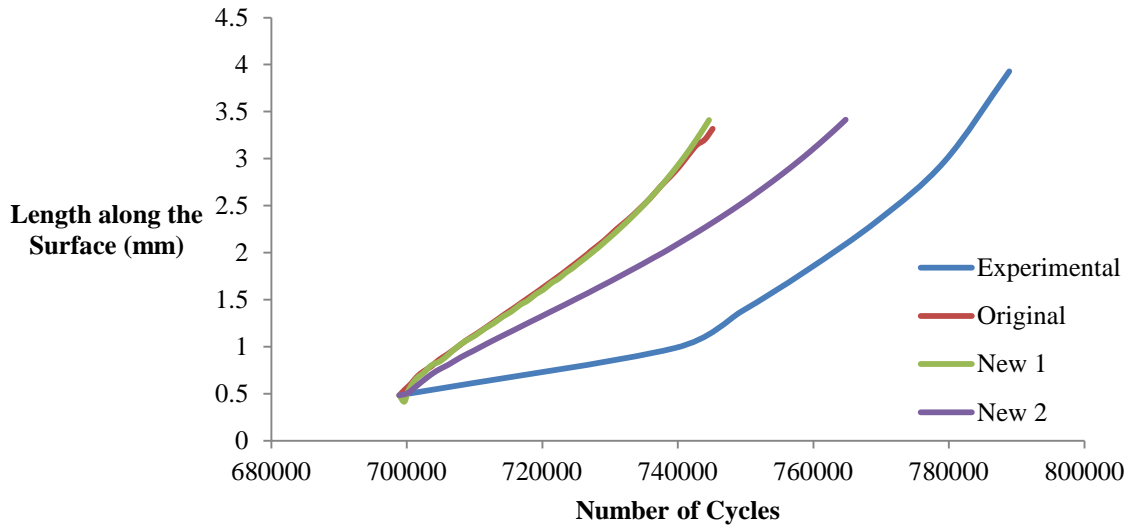


Figure 4.29: Length along the surface vs. number of cycles comparing the Paris tests to experimental results.

The accuracy of the program in predicting crack shape and the growth pattern of a crack was confirmed through analysis with experimental results from NRC. The crack shapes predicted represented the experimental results extremely well from a complex initial shape to a through crack. The distribution of the SIFs over the crack front, even for the complex initial shape, were accurately predicted and the growth model was able to take the extracted values and accurately project them to form new crack fronts. The ability of the program to predict fatigue, when compared to the experimental NRC results, was confirmed for the stage of crack growth when the crack assumed a “smooth” front and grew to a through crack. For the stage of growth from complex initial shape to “smooth” crack front the predicted results did not correspond. Further investigation into the experimental testing may be necessary to determine the reason for the unexpected sharp change in slope.

4.5 Summary

The accuracy of the ACGP in predicting SIFs was confirmed by comparing predicted results to those from Bakuckas and Fawaz and Andersson. The ACGP was able to accurately predict SIFs for simple crack geometries where the corner crack was almost semicircular in shape, as well as for crack fronts with small aspect ratios. SIFs were accurately predicted where the crack front intersected the free surfaces and finite width effects were also captured.

The accuracy of the growth model within the ACGP was confirmed through a comparison of the fatigue life to the results predicted by AFGROW. An interesting calculation technique employed by AFGROW was observed where the fatigue life that was output was less than what AFGROW calculates internally.

When the entire ACGP was run to predict fatigue life and crack shape, one major limitation of the AFGROW program was observed. The crack shape and growth pattern predictions matched reasonably well but the AFGROW fatigue life predictions were very low compared to the ACGP. The conservativeness in the AFGROW predictions was due to higher ΔK_{max} values being predicted which is most likely due to the use of an ellipse to fit the crack front. The use of a spline in the ACGP resulted in growth that was much slower than predicted by AFGROW.

The accuracy of the ACGP in predicting crack shape and the growth pattern of a crack was also confirmed through comparison with experimental results provided by NRC. The crack shapes predicted represented the experimental results extremely well from a complex initial shape to a through crack. The advantage of using a spline fit to many points along the crack front to accurately predict crack shapes and the growth pattern is displayed exceptionally well with these results. The ability of the ACGP to predict fatigue life, when compared to the experimental NRC results, was confirmed for the stage of crack growth when the crack assumed a “smooth” front and grew to a through crack. For the stage of growth from

complex initial shape to “smooth” crack front the predicted results did not match the experimental results, but this was thought to be due to some unknown features of the experimental results.

It was observed that Paris material parameters (C and m) have a significant influence on the fatigue life but not a large influence on crack shape. This conclusion enables Paris parameters to be altered during the ACGP post-processing stage to observe their effects on fatigue life without requiring a full analysis of growth to be completed.

To obtain the most accurate fatigue life predictions, the crack front should be represented by a spline with many extraction points along the crack front. Using this method the variation of SIFs along the crack front can be accounted for during analytical growth which will generate realistic predictions of crack shape evolution and fatigue life. If this type of method is not employed it may lead to inaccurate SIF predictions and result in inaccurate crack shape and fatigue life. The validation studies presented in this chapter confirm that the ACGP can be employed to predict fatigue life and crack growth in thin plates. This gives confidence that the ACGP can also predict crack shape and fatigue life for countersunk holes.

Chapter 5 Crack Growth in Countersunk Fastener Holes

This chapter presents and discusses a series of crack growth and fatigue life predictions for mode I cracks emanating from countersunk fastener holes in a plate undergoing cyclic tensile loading. First, predicted crack growth pattern and fatigue life were compared for the ACGP and AFGROW to ensure the ACGP was producing valid results. Then, the ACGP was used to determine the effects of different countersunk and plate geometry on fatigue life.

In aircraft fuselage skins, countersunk holes with an angle of $\gamma=100^\circ$ are typically found. One commonly used sheet material for fuselage skins is aluminum alloy 2024-T3 with the available countersunk geometries being dependent on the thickness of the plate due to strength requirements. Figure 5.1 shows a countersunk fastener used to hold two sheets together with the countersink angle (γ), the countersink depth (C_s) and the sheet thickness (t) labeled. For this thesis, only crack propagation through an isolated outer sheet is modelled. The countersunk fastener and inner sheet are ignored as they would require a more advanced finite element model that included contact. The outer sheet was assumed to be made of 2024-T3 and the corresponding countersunk radius and head height for the applied sheet thicknesses were found in MMPDS [49].

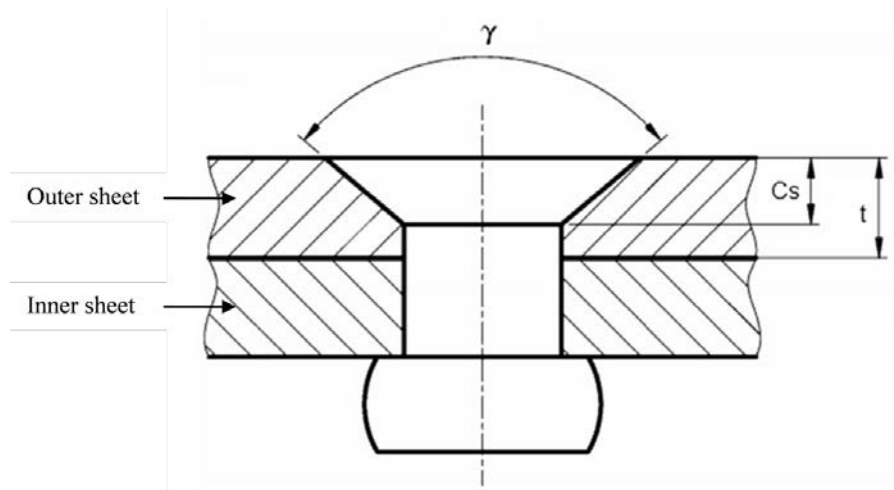


Figure 5.1: Geometry of a countersunk fastener used to hold two sheets together.

Three different initial crack locations are considered in this chapter, previously defined as corner, knee and upper, as shown in Figure 5.2. A corner crack is located at the intersection of the lower surface and the straight shank of the countersunk hole, a knee crack is located at the intersection of the straight shank and the countersink, and an upper crack is located at the intersection of the upper surface and the countersink. The corner and knee crack locations were chosen because they are the locations where cracks are typically initiated in aircraft countersunk fasteners. The upper location was analysed because it is a location where a crack can be spotted upon visual inspection.

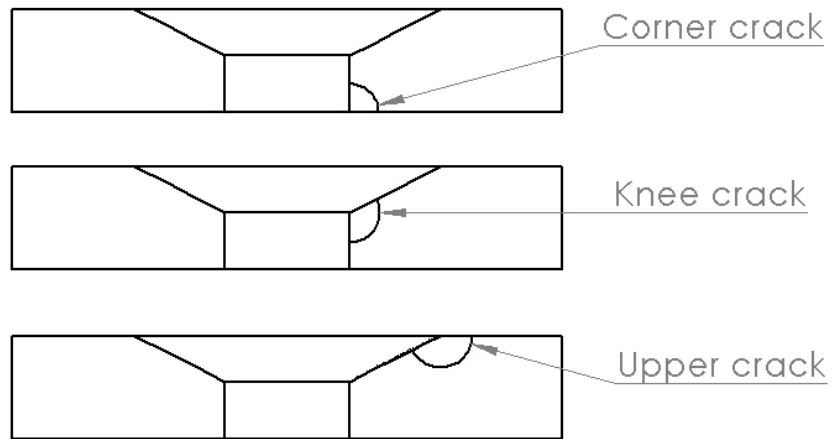


Figure 5.2: Three initial crack locations: corner, knee and upper.

To assist in the discussion of countersunk results, three stages of a cracks growth are defined. These stages are defined as type 1, type 2 and type 3, shown in Figure 5.3 for the three initial crack shapes. A crack is considered to be type 1 during growth from the initial shape to the first surface change, type 2 from the first surface change to just before the crack becomes a through crack, type 3 when the crack has propagated through the thickness of the sheet.

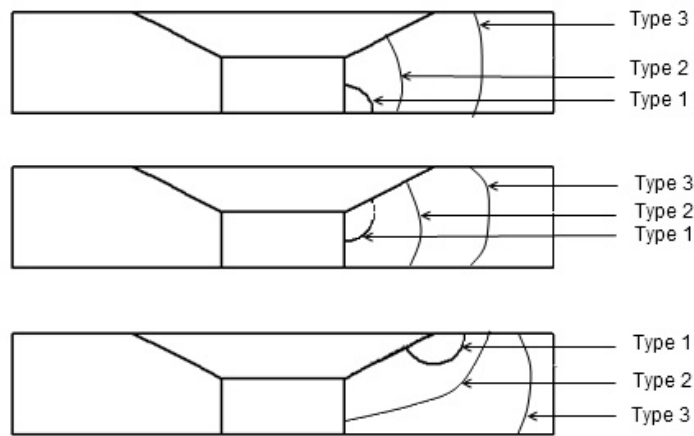


Figure 5.3: Three stages of crack growth in countersunk holes defined as type 1, type 2 and type 3.

To compare the SIF distribution along the crack front for different initial crack locations and crack types the SIF values were converted into beta factors calculated using Equation (33). To relate the beta factors between different cracks, a normalised angle was used to identify the position along the crack front. The direction of the calculated angles along the crack front for each initial crack location is shown in Figure 5.4. The angles were normalised by dividing each angle by the maximum angle found for each crack. For example, a countersunk angle of 100° would have a maximum angle of 90° for a corner crack, 130° for a knee crack and 140° for an upper crack.

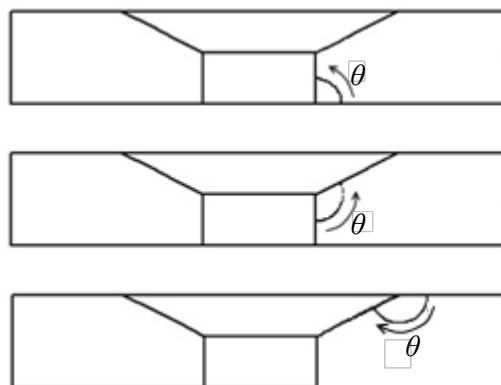


Figure 5.4: Direction of the normalized angles calculated for a corner, knee and upper crack in a countersunk hole.

5.1 Comparison between the ACGP and AFGROW Predictions

Problem Description

An analysis was performed to compare the ACGP predictions with those from AFGROW for cracks emanating from countersunk holes in a plate. To predict growth and fatigue life for cracks at countersunk holes, AFGROW uses a similar method as discussed in section 4.3. Growth is projected at two points along the crack front from embedded beta solutions and an ellipse is fit to the two points. For countersunk holes, the two points where growth is projected are defined as a and c , where c is the horizontal length along the lower surface and a is the vertical length along the straight shank, as shown in Figure 5.5. When the crack transitions along the countersink, a is defined as the distance through the thickness at the point of intersection with the countersink [47]. AFGROW uses the beta solutions generated by de Rijck who employed an hp -version FEM [50]. AFGROW is only able to generate results for mode 1 corner cracks under cyclic tensile loading with a countersink angle of 100° . The AFGROW countersunk geometry and interface is shown in Figure 5.5.

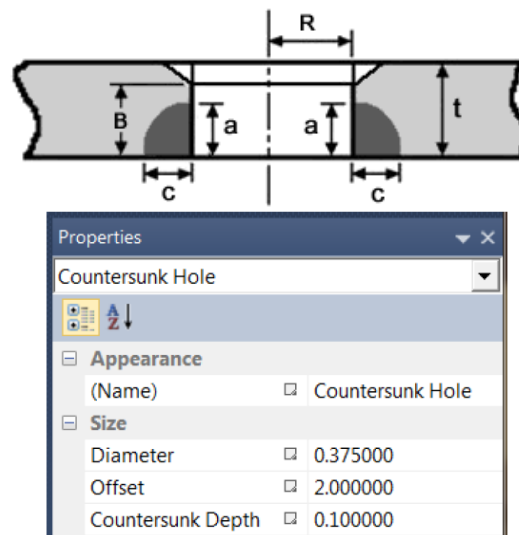
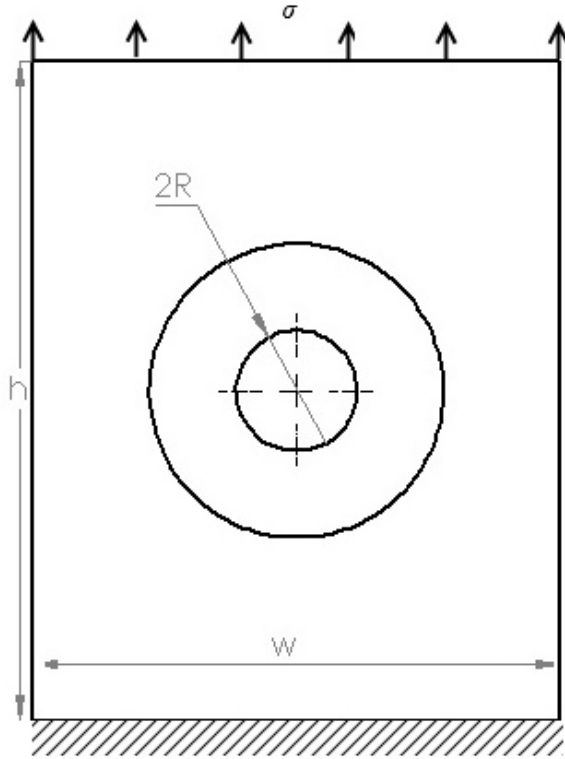


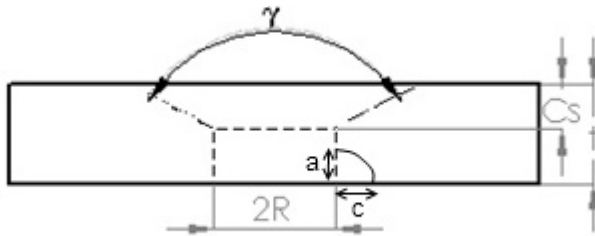
Figure 5.5: AFGROW countersunk geometry and interface.

The geometry chosen for this validation study was based on the geometry employed by Grandt Jr. [37] for a corner crack emanating from a countersunk hole in a plate and summarized Figure 5.6. The width of the plate (w) was 189.7mm, the height (h) was 381mm, the thickness (t) was 9.53mm, the radius of the straight shank (R) was 19.1mm, the countersunk depth (C_s) was set to 4.76mm and the countersunk angle (γ) was 100° . The corner crack was fit to a quarter ellipse with a horizontal radius of 2.17mm and a vertical radius of 2.41mm. The material properties for polymethyl methacrylate (PMMA) were a modulus of elasticity (E) of 2.76GPa and a Poisson's ratio (ν) of 0.35. The Paris exponent (m) was 5.474 and the Paris constant (C) was 3.22×10^{-6} m/cycle, $\text{MPa}\sqrt{\text{m}}$ [37] for a stress ratio of 0.3. A stress of 3.72MPa was applied with a stress ratio of 0.3.

Version 3 of the ACGP was used for this study, with the run parameters shown in Figure 5.6. The ACGP took 79 iterations using a maximum growth increment of 0.05mm (the crack increment was increased to 0.076mm after 27 iterations) to reach a final length along the surface of 9.14mm. $ctrat$, $mrat$ and $trans$ mesh values were 0.03, 0.2 and 0.02 respectively but the $ctrat$ was reduced to 0.025 after 13 iterations.



Images not to scale



Model Parameters

Stress	3.72MPa
Stress Ratio	0.3
E	2.76GPa
ν	0.3
Paris m	5.474
Paris C	3.22×10^{-6} m/cycle, MPa \sqrt{m}
Δc_{max}	0.052mm -> 0.076mm

Geometric Parameters

h	381mm
w	189.7mm
R	19.1mm
a	2.41mm
c	2.17mm
Cs	4.76mm
t	9.53mm
γ	100°

Meshing Parameters

ctrat	0.03*
mrat	0.2
trans	0.02
ipr	0.2
tTotalrat	0.15
tOrat	0.15

*ctrat reduced to 0.025 after 13 iterations

Figure 5.6: Model, geometric and meshing parameters for the ACGP comparison with AFGROW.

Results and Discussion

The predicted crack shape was compared using the aspect ratio (a/c) versus the crack length along the lower surface of the plate seen in Figure 5.7. The results indicate that a difference in the predicted crack shape occurs when the crack transitions from a type 1 to a type 2 crack, but as the crack continues to propagate, both programs predict the same overall crack shape. The largest aspect ratio occurs at the transition from type 1-type 2. The transition for type 2-type 3 occurs at an aspect ratio close to 1.

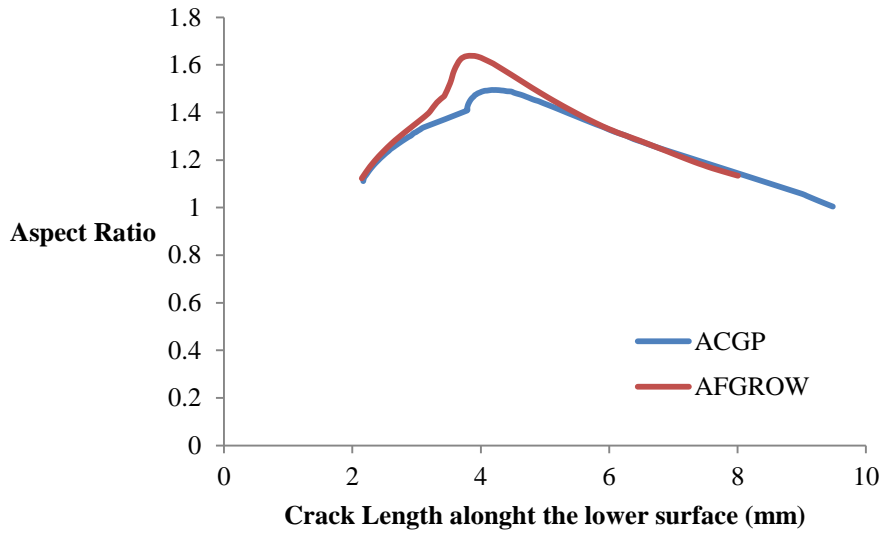


Figure 5.7: Aspect ratio vs. growth along the lower surface comparing the ACGP to AFGROW for a crack emanating from a countersunk hole.

The crack shapes at six locations predicted by the ACGP and AFGROW are shown in Figure 5.8. The crack fronts matched well as the crack propagates within the type 1 and type 2 stages but differences in the predicted shape occur after the crack has transitioned, from type 1-type 2 and from type 2-type 3.

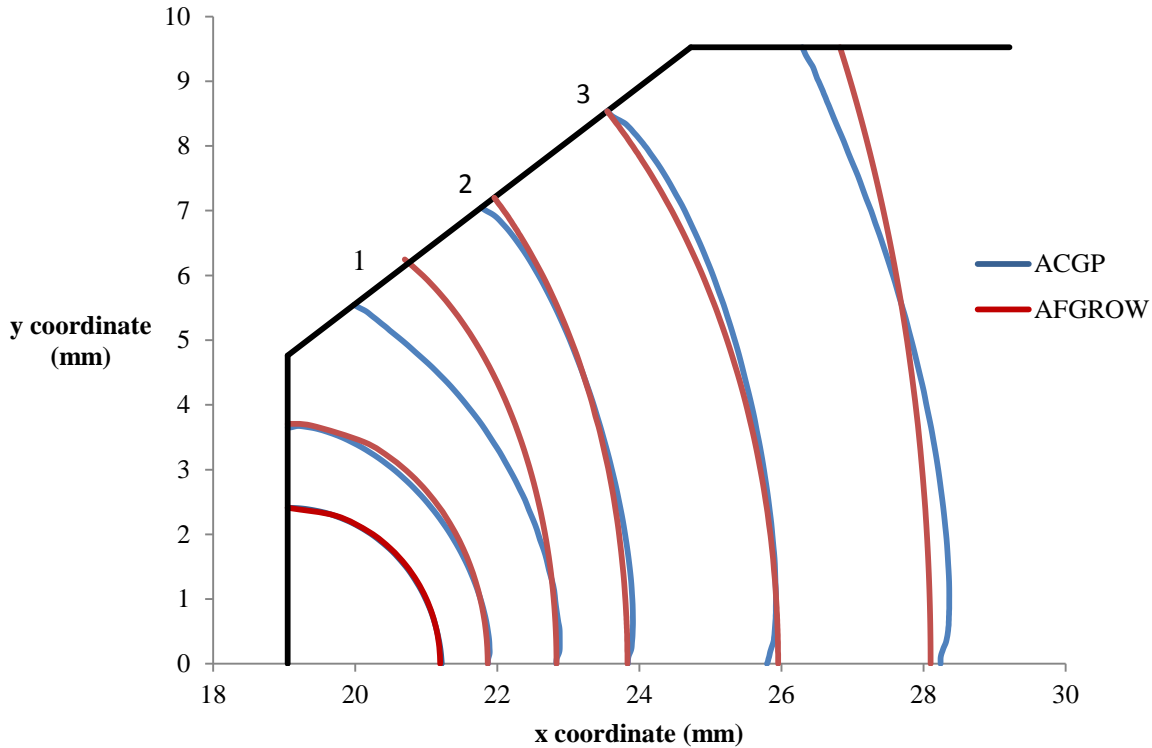


Figure 5.8: Crack shape comparison for different stages of growth from the ACGP and AFGROW predictions.

The differences in crack shape after transition are due to the different methods used to calculate growth through the transition stages. The ACGP uses the SIF values at a crack front that is just before transition point to predict the crack shape after transition. The predictions of SIFs cannot occur directly at the transition point because a proper crack tip mesh cannot be generated there. This fact may have led to the influence of the transition not being fully accounted for in the ACGP calculations. Figure 5.9 shows a mesh of the last iteration of a type 1 crack before transition. Any further growth of the crack along the straight shank will cause the outer radius of the mesh to intersect the edge of the countersink creating a mesh failure. The SIFs predicted along the crack front at this point are used to project the crack through transition to a type 2. Further developments of the program should focus on enabling the analysis of SIFs to occur closer to the transition to fully capture its effects. Figure 5.8 shows that the shape of crack 1 is almost the same as the crack prior to transition.

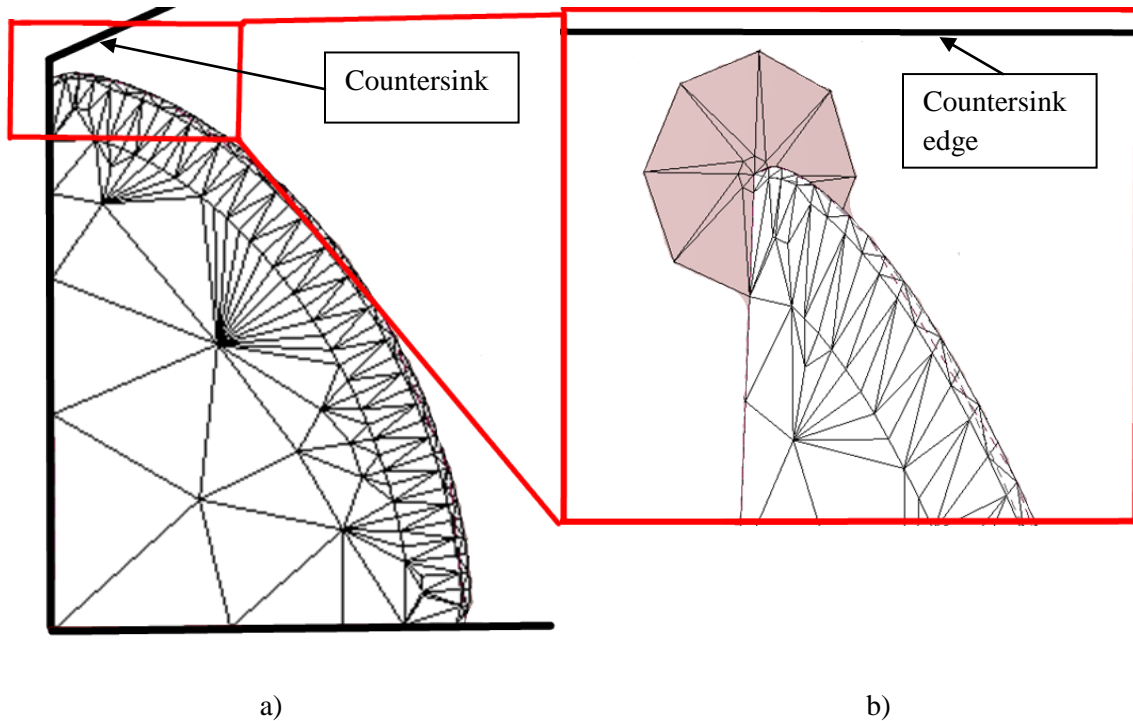


Figure 5.9: Mesh of a crack before transition a) crack front viewed from inside the plate b) crack tip near the intersection of the straight shank and countersink.

AFGROW enables the crack to be modelled directly at the transition point which allows the influence of the intersection to be captured. One issue with the AFGROW results are that the SIFs are only determined for five positions along the straight shank portion of the growth (the positions are not specified) and the remaining SIFs are interpolated from those five positions [47]. This fact may have led to an over prediction of SIFs just before transition.

Figure 5.10 shows that both programs predict the growth rate of a type 1 crack to be very similar and the number of cycles predicted for the growth up to the transition (approximately 23000 cycles) is within 1% for both a and c growth. Through transition there is a maximum difference between the growth of a of 12%.

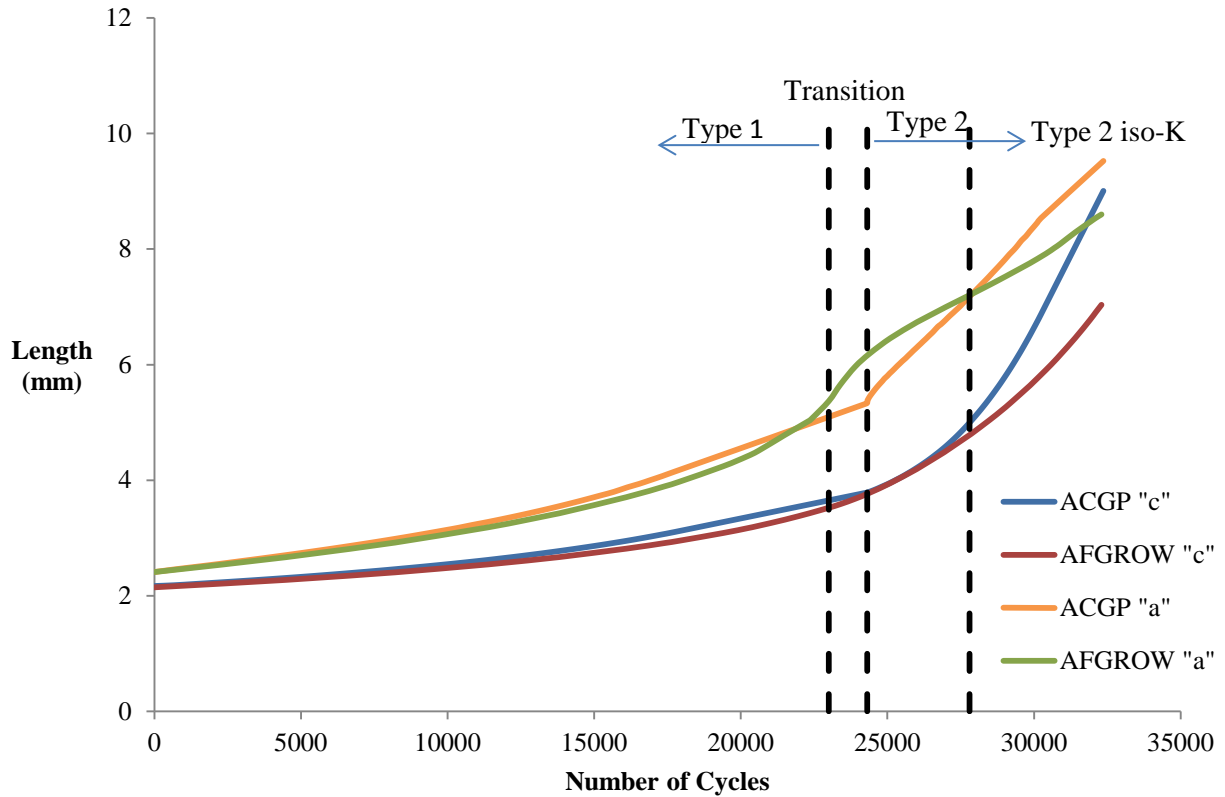


Figure 5.10: a and c growth vs. number of cycles comparing the ACGP and AFGROW for a crack emanating from a countersunk hole.

After transitioning to a type 2 crack, Figure 5.10 shows the predicted growth rate is higher in the ACGP. This is due to the fact that the ΔK_{max} value used within the ACGP to calculate the number of cycles is larger than any ΔK value predicted by AFGROW as shown in Figure 5.11 and Figure 5.12. These two figures compare the K distribution along the crack front produced for the ACGP and AFGROW geometries for crack numbers 2 and 3 (identified in Figure 5.8). The ACGP uses a spline to fit the crack front while AFGROW uses an ellipse. Figure 5.11 and Figure 5.12 show that both the ellipse and spline produce the same ΔK distribution with only slight variation near the free surfaces. For crack 3, the ΔK distribution is constant, indicating the crack has reached its preferred shape. For both cracks 2 and 3 the ACGP predicts a local rise in ΔK near the intersection with the countersink. This did not occur for the type 1 crack and is thought to be because of the angle at which the crack intersects the free surface. The spline representation intersects the free surface at a more severe angle than the ellipse. Heyder and

Kuhn found that SIFs will increase at the edge of a crack where it intersects the free surface if the angle of intersection of a convex crack front is increased [51]. The local increase in ΔK leads to an average percent difference of ΔK_{max} of 2.7% between the ACGP and AFGROW although the exact location where AFGROW considers ΔK_{max} to be is not known. As shown in section 4, a small difference in ΔK_{max} values will lead to a large difference in predicted fatigue life as the crack grows.

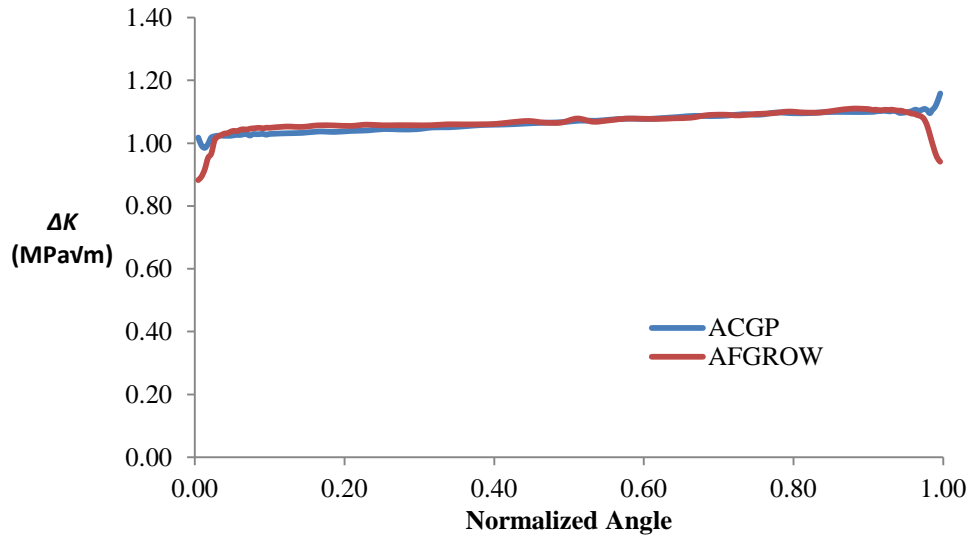


Figure 5.11: ΔK vs. normalized angle for shape 2 spline and ellipse.

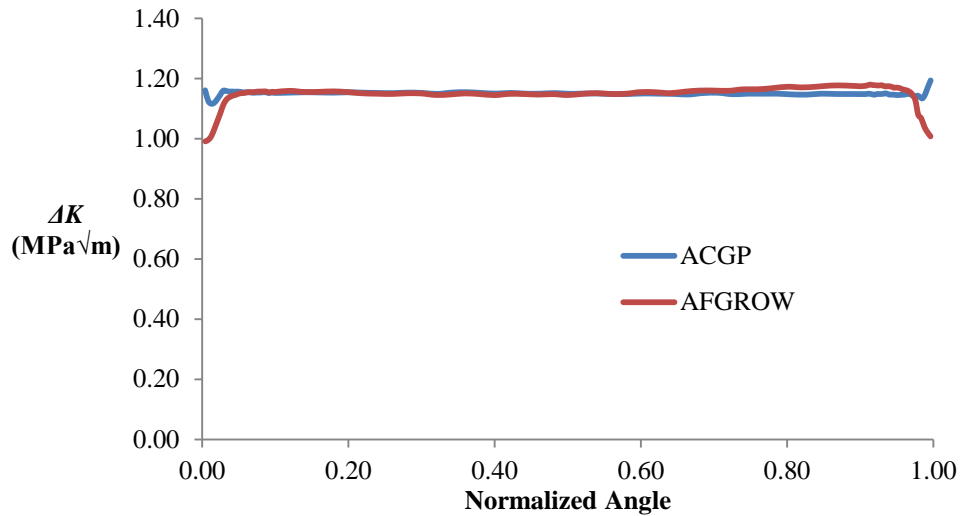


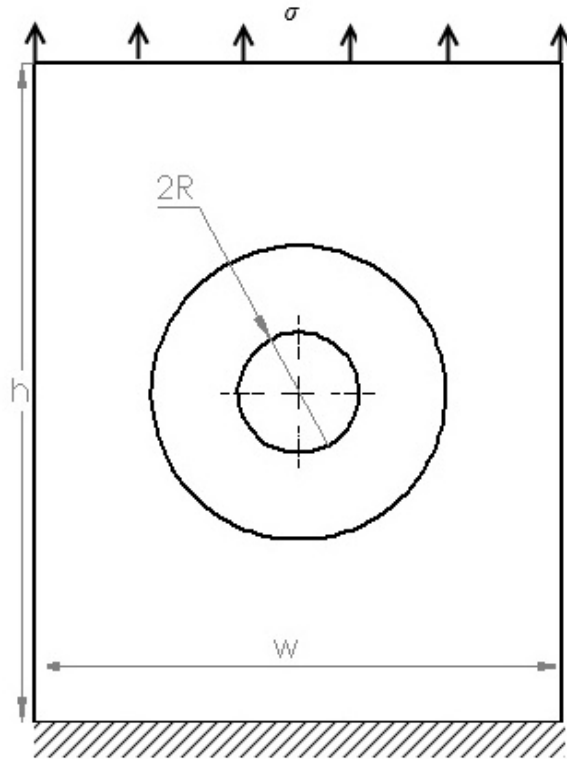
Figure 5.12: ΔK vs. normalized angle for shape 3 spline and ellipse.

The results from this section illustrate that some further work should be performed to improve the ACGP for predicting SIF values close to transition points. The consequence of not fully capturing the transition points is a slightly more conservative estimation of fatigue life which is not detrimental to damage tolerance analysis. The final shape however, is not influenced by the transition region, as it eventually grows to a shape that produces an iso- K distribution. The results also indicate that for growth of a type 2, crack the ACGP predicts a faster growth rate caused by a higher ΔK_{max} . The reason for the local increase in K at the countersink surface should be investigated further.

5.2 Influence of Countersunk Geometry on Fatigue Life

With the accuracy of the ACGP validated, a series of simulations using the ACGP for cracks emanating from countersunk holes were completed to determine how cracks behave for different countersunk geometries. These results provide information that could be beneficial for aircraft design and damage tolerance analysis, by identifying critical countersunk geometries and crack locations.

A series of 13 simulations were conducted to determine the effects of radius (R), angle (γ), sheet thickness (t) and initial crack location on crack shape and fatigue life predictions at countersunk holes. The geometry was based on a MS20426D countersunk rivet in a 2024-T3 Alclad aluminum sheet because of their use in aircraft fuselage skins [33]. The baseline hole radius (R) of 3.26mm, the baseline thickness (t) of 3.26mm and the baseline countersunk head height (C_s) of 2.41mm were taken from MMPDS [49] for this particular material. The Paris growth parameters $C=1.67 \times 10^{-12}$ m/cycle, $\text{MPa}\sqrt{\text{m}}$ and $m=3.07$ for a 2024-T3 Alclad aluminum sheet corresponded to those from Fawaz and Schijve [38] for a stress ratio of 0. The initial crack fronts were chosen to be a quarter circles with a radius of 0.4mm. A quarter circle was used to represent the crack front so that the influence of countersunk geometry on the crack shape could be isolated. Figure 5.13 shows the baseline model, geometric and meshing parameters used for the tests.



Model Parameters

Stress	100MPa
Stress Ratio	0
E	72.4GPa
ν	0.33
Paris m	3.07
Paris C	1.67×10^{-12} m/cycle, MPa \sqrt{m}
Δc_{max}	0.02mm

Geometric Parameters

h	130.6mm
w	98.0mm
Crack Radius	0.4mm
Crack Location	Corner
R	3.26mm
Cs	2.41mm
t	3.26mm
γ	100°

Meshing Parameters

ctrat	0.02
mrat	0.2
trans	0.02
ipr	0.2
tTotalrat	0.15
t0rat	0.15

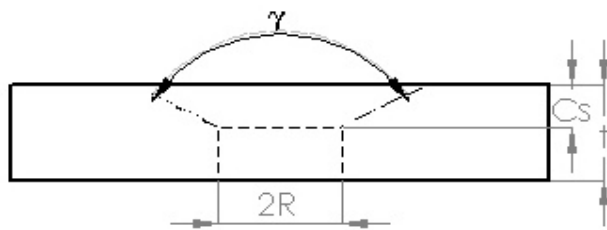
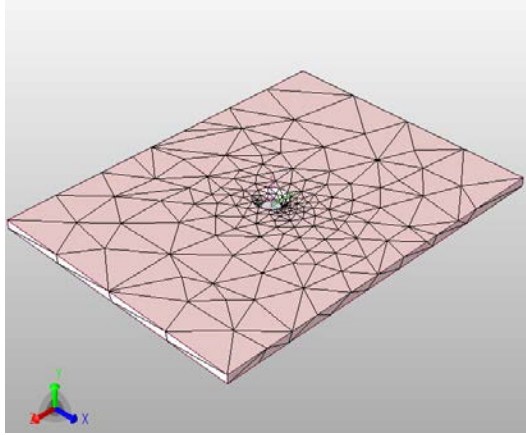
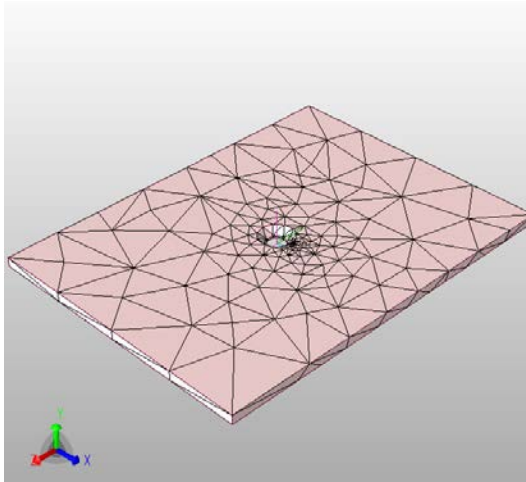
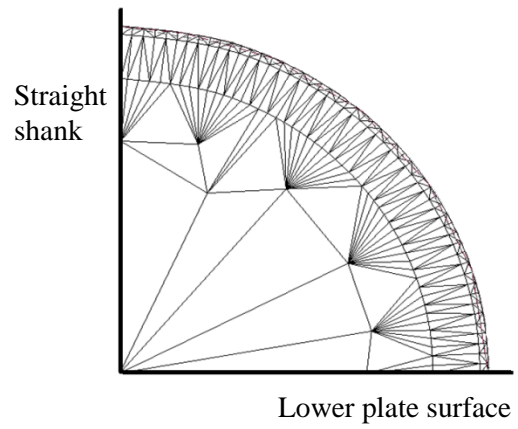


Figure 5.13: Baseline model, geometric and meshing parameters used for countersunk tests.

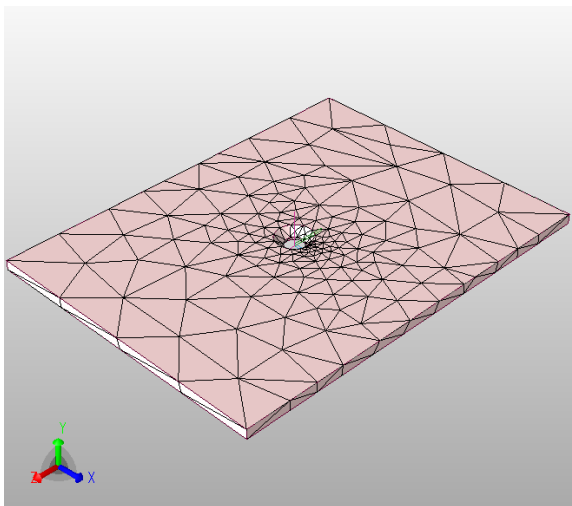
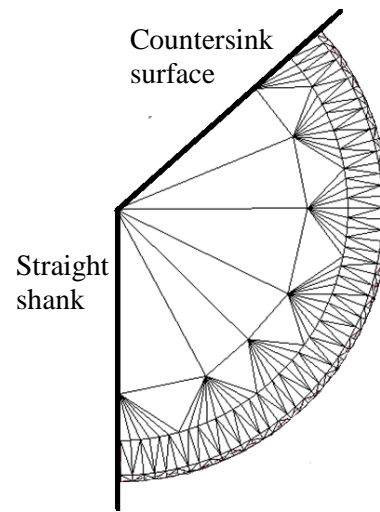
An image of the baseline mesh used for the tests is shown in Figure 5.14 for an initial corner, knee and upper crack.



a)



b)



c)

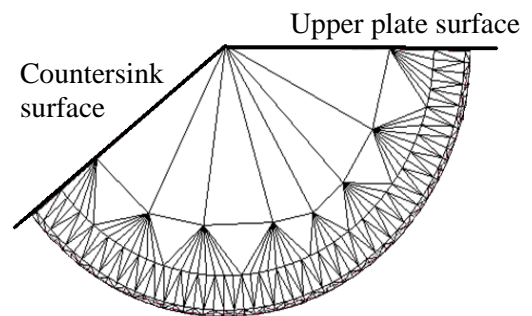


Figure 5.14: Baseline mesh for the countersunk tests for an initial a) corner crack b) knee crack c) upper crack.

The variations in hole radii were determined from MMPDS based on a MS20426D countersunk rivet in a 2024-T3 Alclad aluminum sheet. The plate thickness cases of 0.9R, 1R and 2R were chosen because plate thicknesses between 0.5R-2R represent aircraft structural joints [52]. The length and width of the plate were set to $w/r=40$ and $w/r=30$ to avoid the finite width effects [9]. Table 5.1 lists all 13 simulations that were completed with the initial geometry input for each test found in Appendix A. All simulations were run until the initial crack had propagated through the thickness.

Table 5.1: Simulations for cracks emanating from countersunk holes in a plate.

Simulation #	Initial Crack Location	γ	t (mm)	R (mm)
1	Corner	80°	3.26	3.26
2	Corner	100°	3.26	3.26
3	Corner	120°	3.26	3.26
4	Knee	80°	3.26	3.26
5	Knee	100°	2.61	3.26
6	Knee	100°	3.26	2.43
7	Knee	100°	3.26	3.26
8	Knee	100°	3.26	2.02
9	Knee	100°	6.53	3.26
10	Knee	120°	3.26	3.26
11	Upper	80°	3.26	3.26
12	Upper	100°	3.26	3.26
13	Upper	120°	3.26	3.26

These results were not validated with experimental or numerical results as acceptable results were not found in the literature, however trends were compared with literature results.

5.2.1 Discretization Error

To verify that the discretization error was acceptable a full test was run comparing the fatigue and crack shapes for p-levels of 4 and 5. The results from the aspect ratio versus c growth, shown in Figure 5.15 and c growth versus number of cycles, shown in Figure 5.16, indicate that the numerical results have converged using a p-level of 4. This lead to the conclusion that the results from tests run in this thesis

using a p-level of 5 had an acceptable discretization error. Wowk et al. [16] had noted that for an ellipse a p-level of 5 was sufficient.

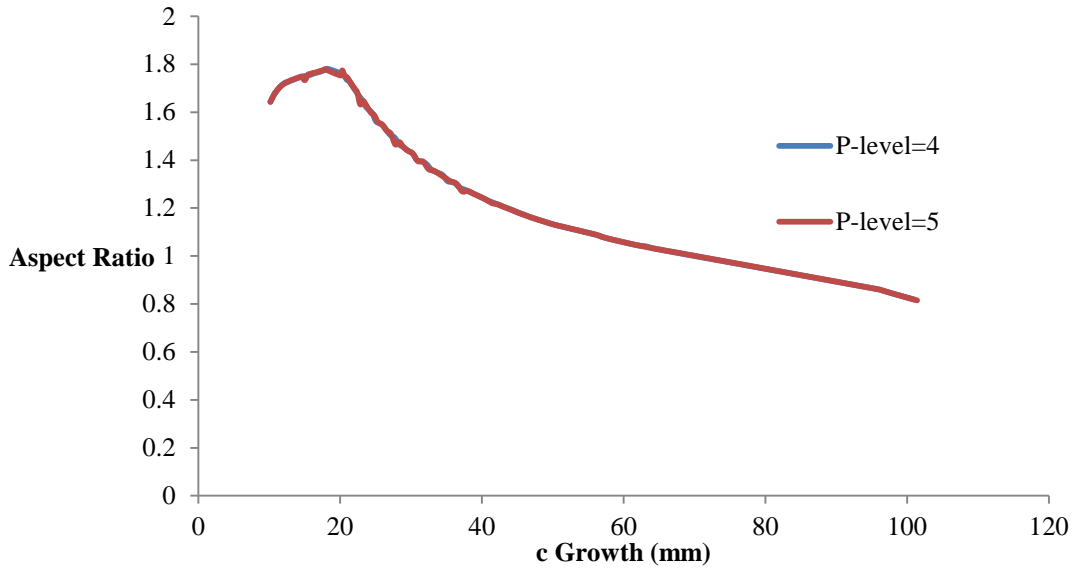


Figure 5.15: P-level test comparing aspect ratio vs. c growth.

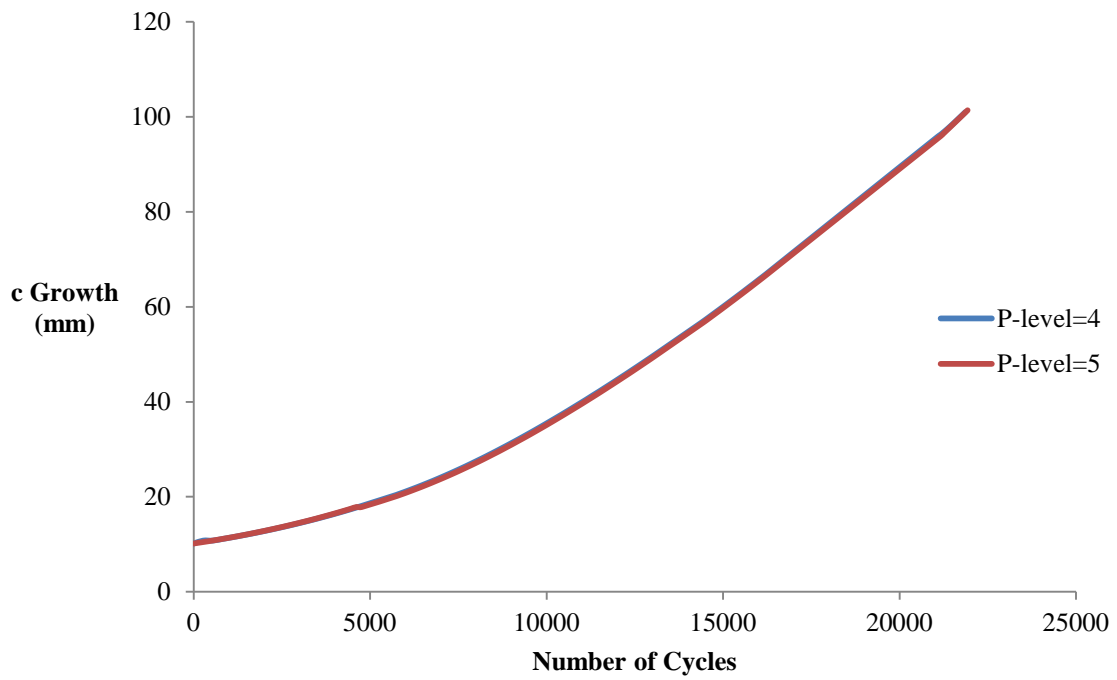


Figure 5.16: P-level test comparing c growth vs. fatigue life.

5.3 Effect of Initial Crack Location

The growth pattern for cracks emanating from countersunk holes was analysed for cracks propagating from three different initial locations. The results from simulations 2, 7 and 12 were used. All three simulations were conducted for a countersunk angle of 100° , a radius of 3.26mm and a thickness of 3.26mm. The only difference between the simulations was the initial crack location. Simulation 2 contained an initial corner crack, simulation 7 contained an initial knee crack and simulation 12 contained an initial upper crack. The analysis focused on crack growth rates through different stages of growth by analysing the predicted crack shape and crack front beta factors at either five or six crack locations identifying important stages of growth. A fatigue analysis was also completed to determine how starting location affects the fatigue life and to determine the critical initial crack location.

5.3.1 Initial Corner Crack (Simulation 2)

The growth pattern for an initial corner crack propagating through the thickness is shown in Figure 5.17 for six crack front geometries. Their corresponding beta factors are shown in Figure 5.18. The six locations represent the initial crack (shape 1), the crack before type 1-type 2 transition (shape 2), the crack after type 1 transition (shape 3), the type 2 iso-K crack (shape 4), the crack before type 2-type 3 transition (shape 5) and the crack after type 2-type 3 transition (shape 6).

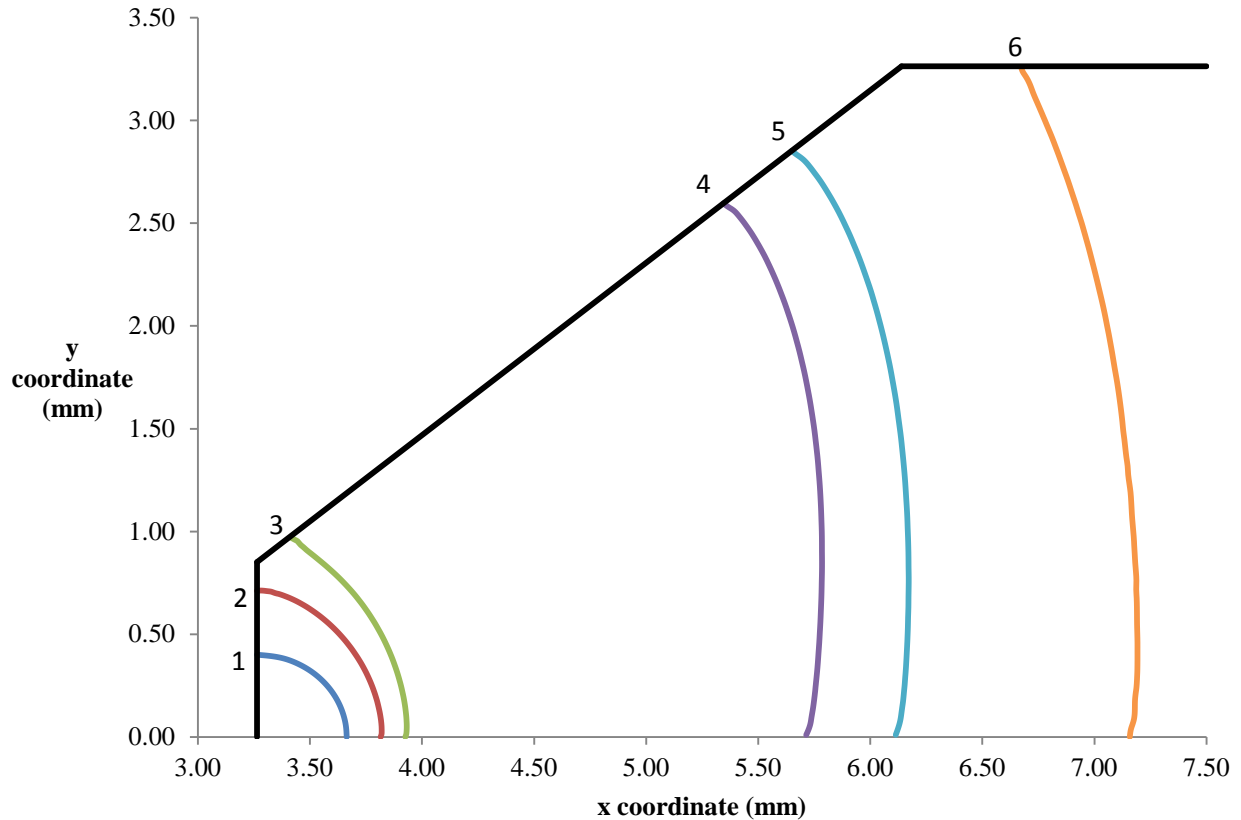


Figure 5.17: Six predicted crack shapes of a corner crack in a countersunk hole.

The results show that for a type 1 crack there was faster growth along the straight shank compared to the lower surface up to the type 1-type 2 transition. After the type 1-type 2 transition there was faster growth along the countersink until the preferred crack shape had been reached (shape 4), identified by an iso-K state. The crack grew with that preferred shape (shape 5) until it transitioned to a type 3 crack where growth along the upper surface was higher, determined by analysing the beta factors.

The predicted beta factors along each crack front, seen in Figure 5.18, indicate the relative growth rates at different stages. The initial crack (shape 1) will propagate faster along the straight shank due to the beta factor increase as the normalized angle approaches 1, where 1 is the edge of the straight shank. After type 1-type 2 transition the crack will propagate faster along the countersink, also seen by the larger beta factor at an angle of 1. The iso-K state observed at shape 4 is identified by the constant beta value

over the crack front, though a slight variation in beta values was found at the edges. Once the preferred shape has been reached, the crack propagates evenly until it transitions to a through crack where the crack will grow slightly faster along the upper surface as indicated by the beta increase as the normalized angle approaches 1 for shape 6. In shapes 1-3 the beta factors are higher for the portion of the crack close to the intersection of the countersink and straight shank, as a result of the local stress concentrations in this region as stated by Shivakumar et al. [52].

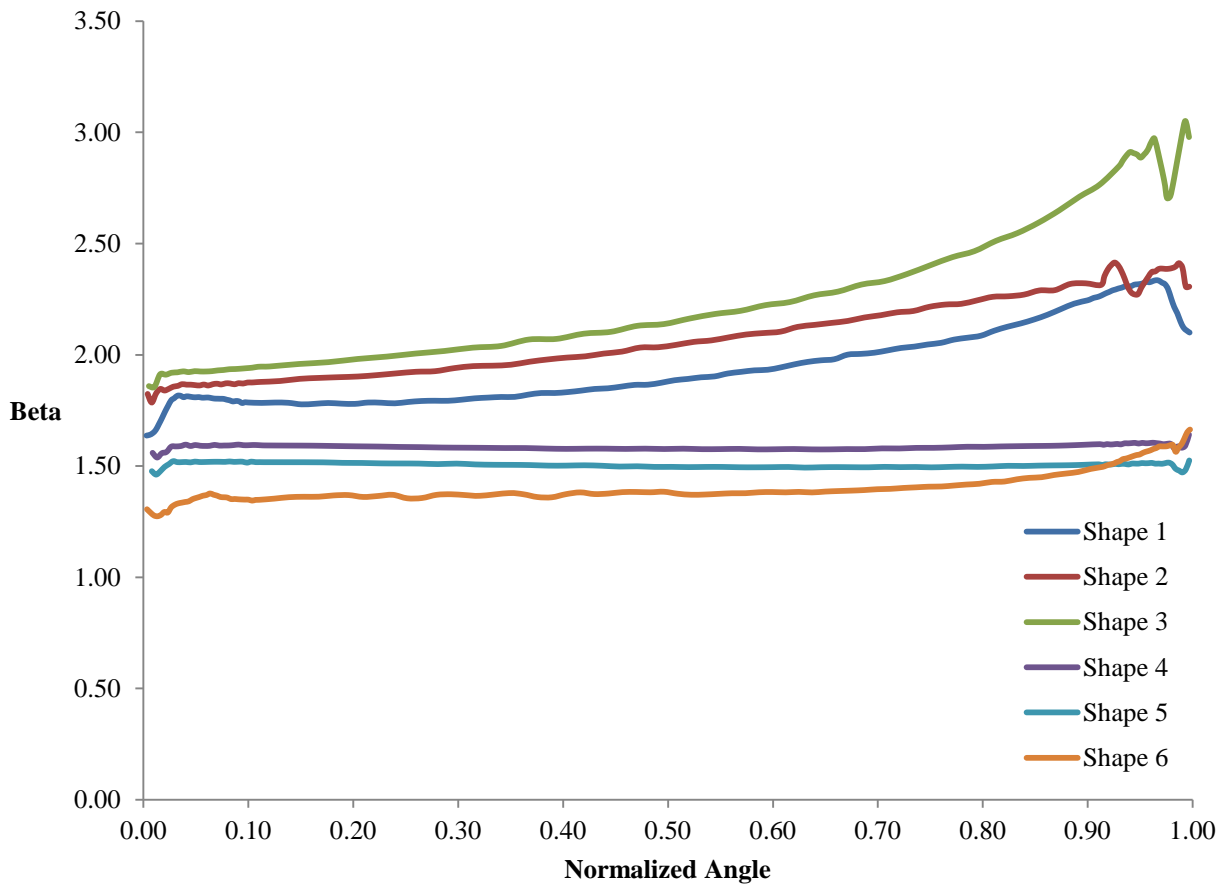


Figure 5.18: Beta vs. normalized angle for six crack shapes with initial corner crack.

5.3.2 Initial Knee Crack (Simulation 7)

The growth pattern for an initial knee crack propagating through the thickness is shown in Figure 5.19 for six crack front geometries. Their corresponding beta factors are shown in Figure 5.20. The six locations represent the same stages as stated in section 5.3.1.

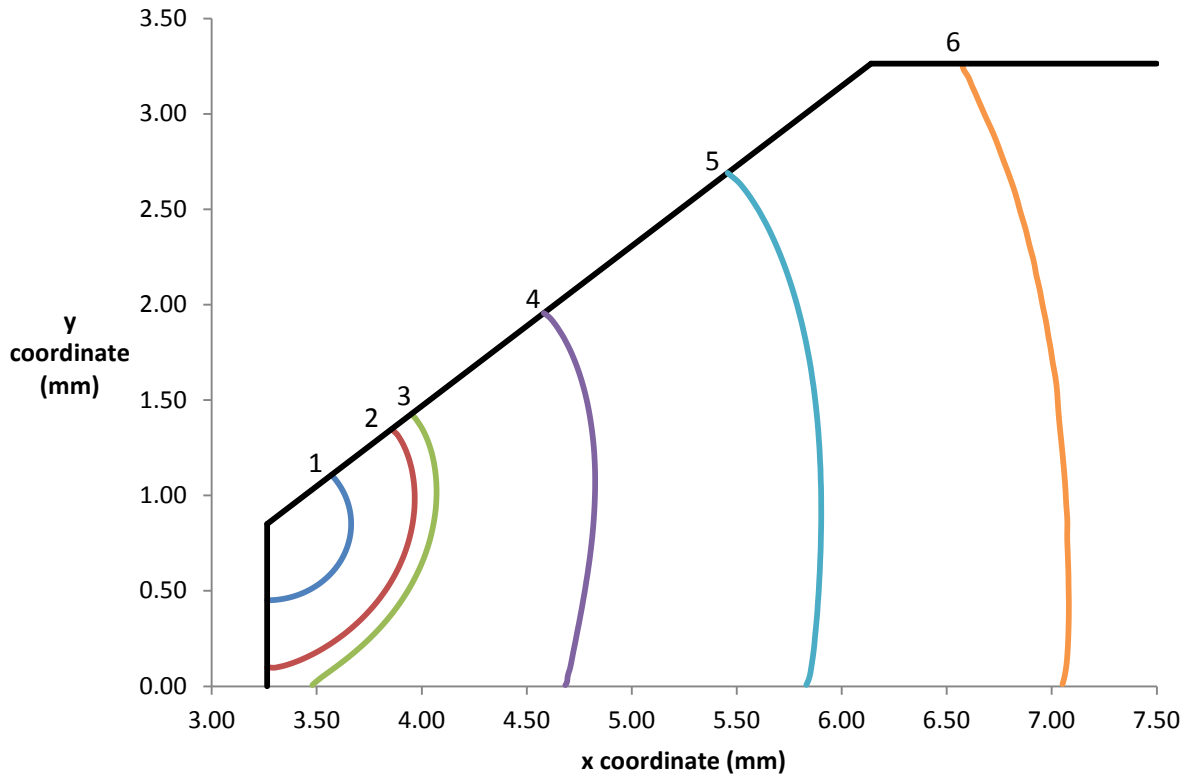


Figure 5.19: Six predicted crack shapes for a knee crack in a countersunk hole.

The results show that for a type 1 crack there was fairly symmetric growth along the crack front, with the highest growth occurring near the edges and slower growth occurring through the middle of the crack. As the crack grew to the type 1-type 2 transition it propagated faster along the lower surface. After the type 1-type 2 transition there continued to be faster growth along the lower surface until the preferred crack shape had been reached (shape 4), identified by an iso-K state. The crack continued with that preferred shape (shape 5) until it transitioned to a type 3 crack (shape 6) where growth along the upper surface was higher, as indicated by larger beta factors.

The predicted beta factors along each crack front can be seen in Figure 5.20. The initial crack (shape 1) will propagate symmetrically with the highest growth rate near the edges indicated by the larger beta factors. The influence of the lower surface on the type 1 crack before transition can be seen by the slightly larger beta for lower normalized angles (shape 2). After the type 1-type 2 transition (shape 3) the crack will propagate faster along the lower surface, also seen by the larger beta factors as the angle approaches 0. The iso-K state observed at shape 4 is identified by the constant beta value over the crack front. Once the preferred shape has been reached the crack propagates evenly until it transitions to a through crack where the crack will grow slightly faster along the upper surface as indicated by the larger beta values as the normalized angle approaches 1 for shape 6. For shape 3, the beta factors are larger for the portion of the crack closest to the lower surface, the accelerated growth along the lower surface is due to the crack front trying to propagate a preferred pattern.

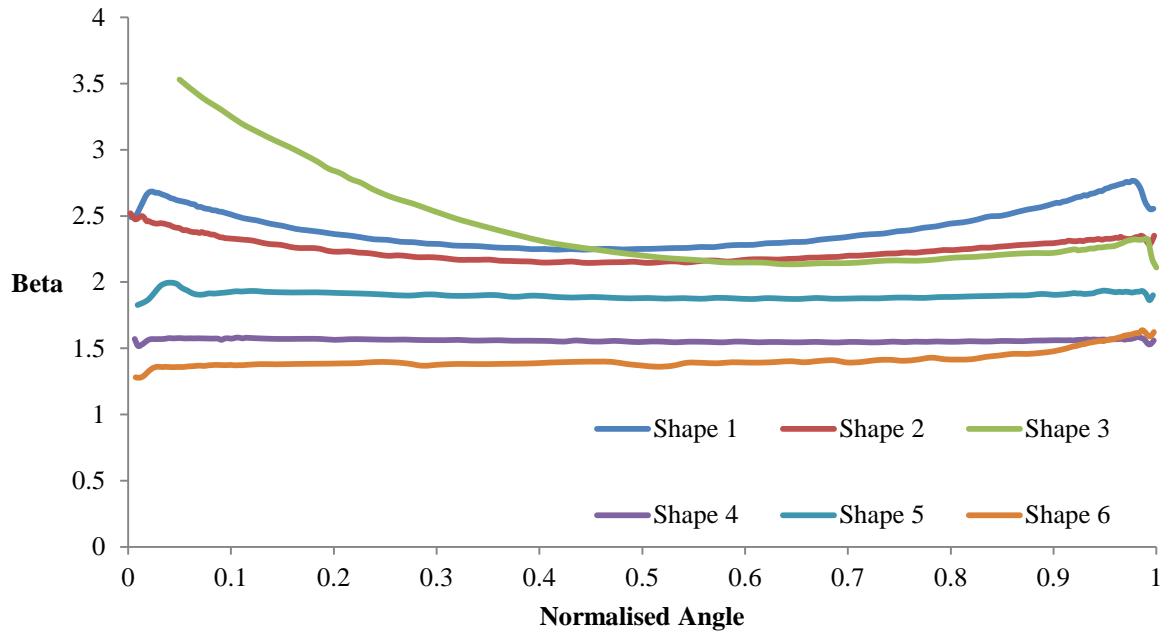


Figure 5.20: Beta vs. normalized angle for six crack shapes with initial knee crack.

5.3.3 Initial Upper Crack (Simulation 12)

The growth pattern for an initial upper crack propagating through the thickness is shown in Figure 5.21. Their corresponding beta factors are shown in Figure 5.22. The five locations represent the initial crack (shape 1), a type 1 crack (shape 2), the crack before type 1-type 2 transition (shape 3), the crack after type 1-type2 transition (shape 4) and the crack after type 2-type 3 transition (shape 5). An iso-K crack was not observed for growth of the initial upper crack, as the program was not run for a sufficient number of iterations.

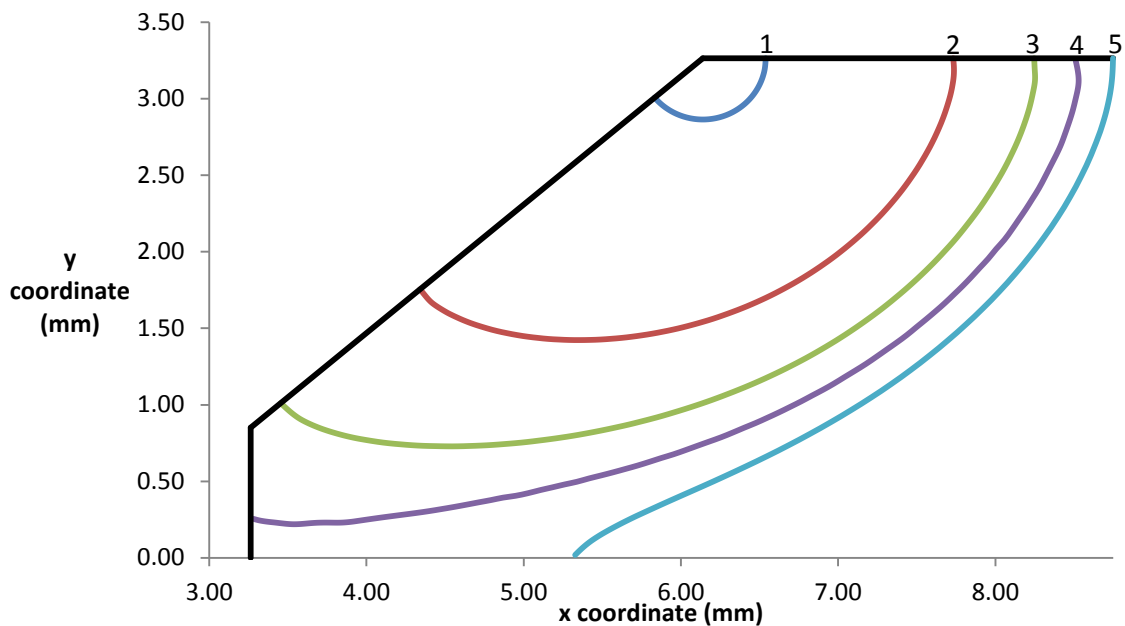


Figure 5.21: Six predicted crack shapes for an upper crack in a countersunk hole.

Figure 5.21 shows that for a type 1 crack there was faster growth along the countersink compared to the upper surface up to the type 1-type 2 transition. After the type 1-type 2 transition there was faster growth along the straight shank. After the crack transitioned to a type 3 crack the growth along the lower surface was faster than the upper surface. As seen in shape 5 the predicted crack shape takes on a “P”- shape, which has been observed in experimental fatigue tests for cracks in aluminum [53]. The upper crack did not reach an iso-K state during this simulation.

The predicted beta factors along each crack front can be seen in Figure 5.22. The initial crack (shape 1) will grow slightly faster along the countersink, and the growth rate along the countersink will increase relative to the rest of the crack front as the type 1 cracks approach the type 1-type 2 transition, the increase in growth rate along the crack front is seen for shapes 2 and 3 where larger beta factors are found as the normalized angle approaches 1. The influence of the straight shank and countersink intersection on the propagation rate along the countersink is more significant as the crack propagates closer to the intersection. After the type 1-type 2 transition the crack will propagate faster along the straight shank, seen by the beta increase as the angle approaches 1. After the type 2-type 3 transition the crack will propagate faster along the lower surface, also seen by the beta increase as the angle approaches 1. The increase in growth rate after the transition is due to the crack front trying to propagate a preferred pattern.

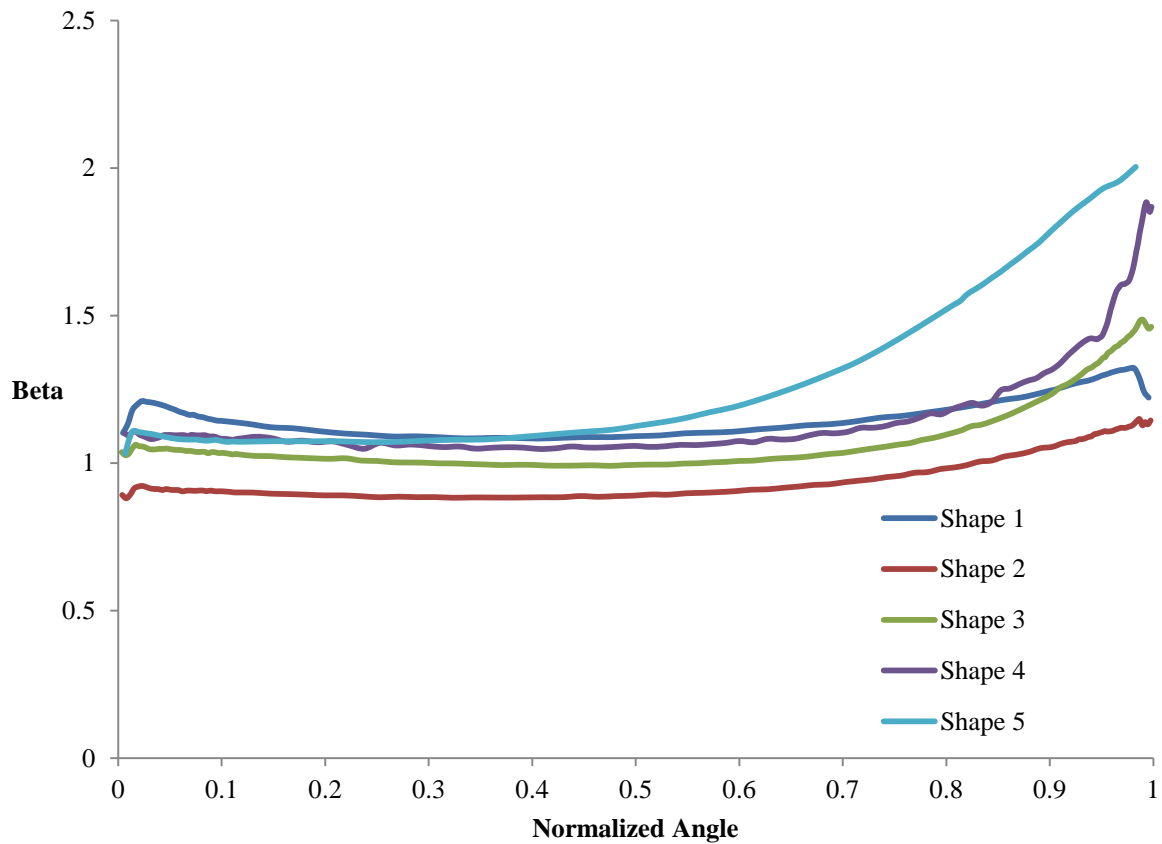


Figure 5.22: Beta vs. normalized angle for six crack shapes with initial upper crack.

5.3.4 Fatigue Life

A fatigue analysis using the results from simulations 2, 7 and 12 was completed to determine how the initial starting locations affected the growth rate by plotting the c length versus the number of cycles for the crack to grow from its initial geometry to through thickness as shown in Figure 5.24. The location of the c dimension used in the analysis for the three starting locations is shown in Figure 5.23.



Figure 5.23: Illustrating the location of c for three initial crack locations.

The initial knee crack propagated through the thickness the fastest, followed by the corner crack and finally the upper crack. It took 10% fewer cycles for the knee crack to reach a c value of 4.01mm when compared to the corner crack and 50% fewer cycles for the knee crack to reach a c value of 4.01mm when compared to the upper crack. Once the knee and corner crack achieved an iso-K state the growth rate between the two was the same as observed when analyzing the slope of the results. The difference in the number of cycles between the initial corner and knee crack is due to the fact that the knee crack propagates to the preferred shape faster.

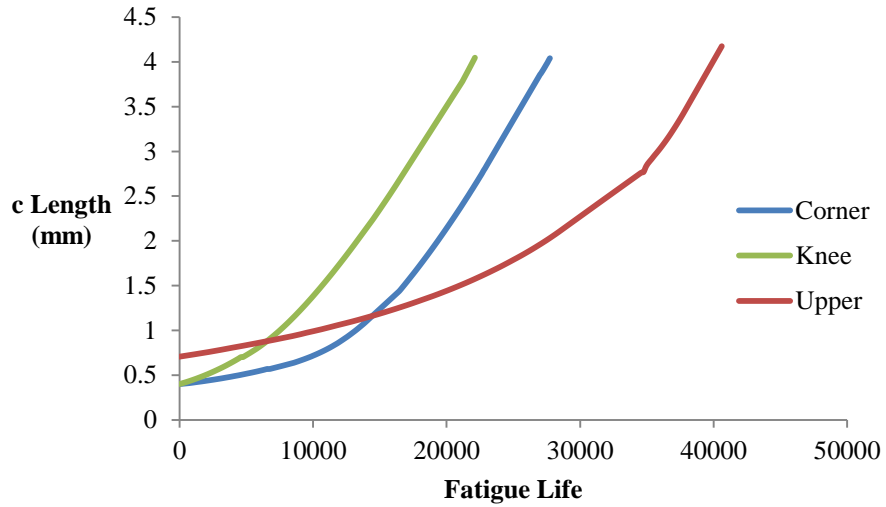


Figure 5.24: *c* growth vs. the number of cycles for three different initial crack locations. The simulated cracks were located in plates made of 2024-T3 Alclad aluminum undergoing cyclic loading of 100MPa with a stress ratio of 0.

The faster growth rate is expected for a knee crack due to the higher stress concentrations at the knee of the countersink, shown in Figure 5.25 for a countersunk fastener hole in a plate under tensile loading. The figure also shows lower stresses at the intersection of the upper surface and countersink, compared to the intersection of the straight shank and lower surface, which explains why corner cracks grow faster than upper cracks.

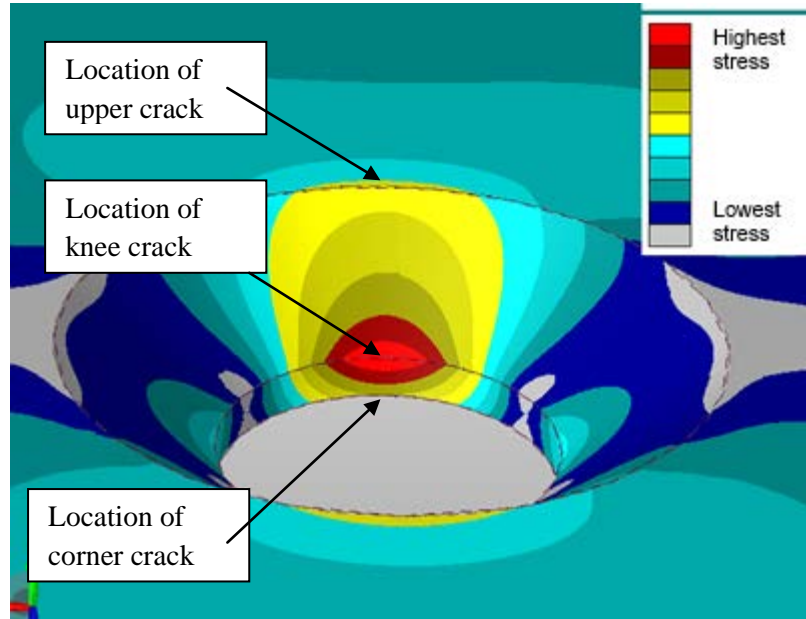


Figure 5.25: Relative Von Mises stress levels for simulation 2 where red shows higher stresses and blue shows lower stresses.

5.4 Effects of Countersunk Geometry

Three sets of simulations were completed to determine how the effect of certain changes in the geometry influences the fatigue life. Knee cracks were used in all cases with an initial quarter circle crack. The first analysis considered the effects of changing the hole radius, the second analysis considered the effects of changing plate thickness and the third analysis considered the effects of changing countersink angle.

5.4.1 The Countersunk Hole Radius

A fatigue analysis was completed to determine how the countersunk radius affected the growth rate by comparing the c length versus the number of cycles for three different radii. The results from simulations 6, 7 and 8 were used in the analysis. For simulation 6 the radius (R) was 2.43mm, for simulation 7 the radius (R) was 3.26mm and for simulation 8 the radius (R) was 2.02mm. The different plate geometries are shown in Figure 5.26.

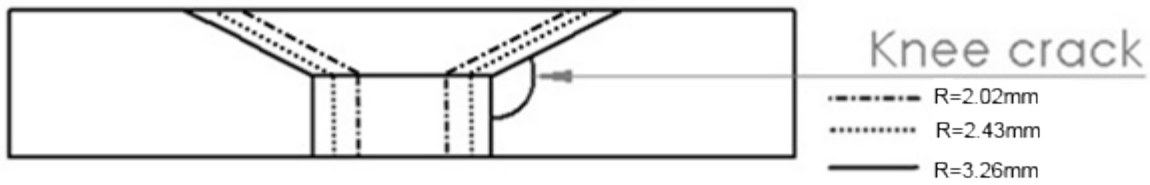


Figure 5.26: Three different countersunk radius geometries.

As shown in Figure 5.27, the crack in the plate with the largest countersunk radius propagated through the thickness the fastest. It took 33% fewer cycles for the crack with $R=3.26\text{mm}$ to reach a c value of 3.56mm compared to the crack with $R=2.02\text{mm}$. It took 19% fewer cycles for the crack with $R=2.43\text{mm}$ to reach a c value of 3.56mm compared to the crack with $R=2.02\text{mm}$.

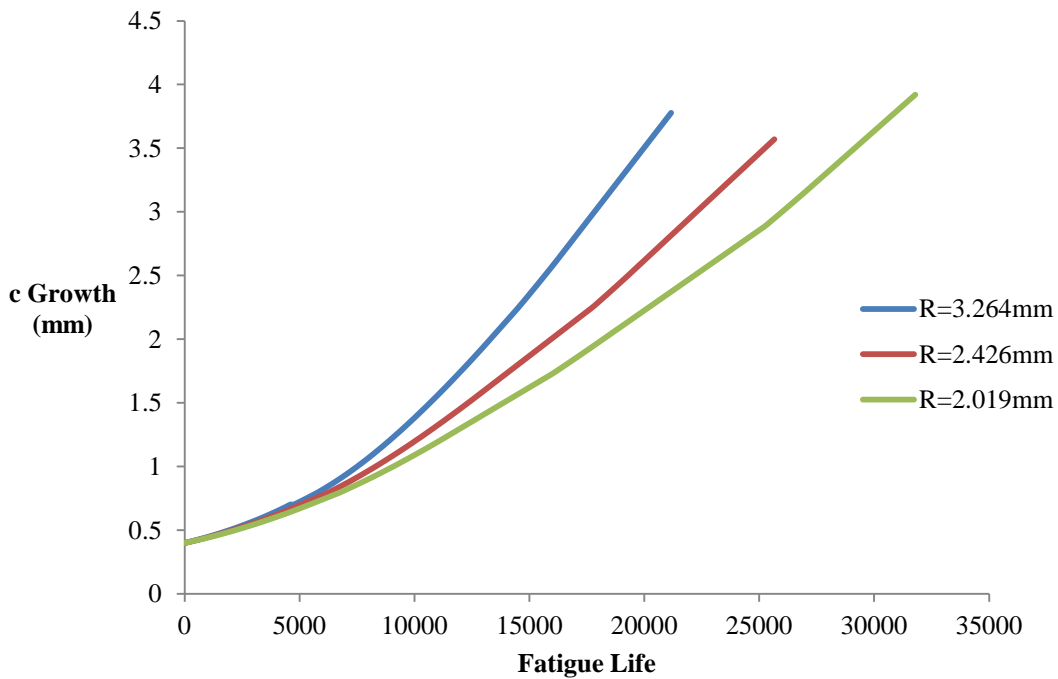


Figure 5.27: c growth vs. the number of cycles for three different countersunk radii.

It was found that for the applied loading and geometry considered in these simulations that cracks will propagate through the thickness of the material faster if the radius of the countersunk hole is increased. This conclusion is in agreement with the observations made by Shivakumar et al. [52], who

found that stress concentrations throughout a plate with a countersunk hole will increase if the radius of the countersink is increased.

5.4.2 Plate Thickness

A fatigue analysis was completed to determine how the thickness of the plate affected the growth rate by comparing the c length versus the number of cycles for three different thicknesses. The results from simulation 5, 7 and 9 were used in the analysis. For simulation 5 the thickness (t) was 2.61mm, for simulation 7 the thickness (t) was 3.26mm and for simulation 9 the thickness (t) was 6.53mm. The different plate geometries are shown in Figure 5.28, with a baseline $R=3.26$ mm.

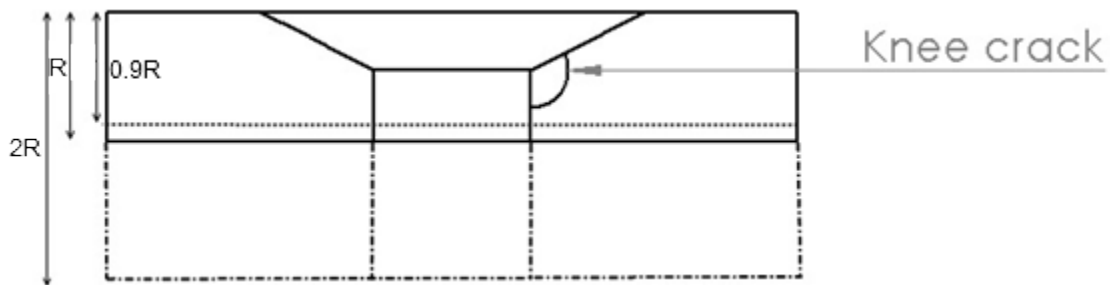


Figure 5.28: Three different countersunk plate thicknesses.

The cracks in the plate with the thicknesses of 2.61mm and 3.26mm propagated through the thickness at the same rate while, the crack in the plate with the largest thickness propagated through the thickness at a slower rate as shown in Figure 5.29. It took 31% more cycles for the crack in the plate with $t=6.53$ mm to reach a c value of 3.94mm compared to the crack with $t=3.26$ mm. There was a difference of 1% between the number of cycles for the cracks in the plates with thicknesses of $t=3.26$ mm and $t=2.61$ mm.

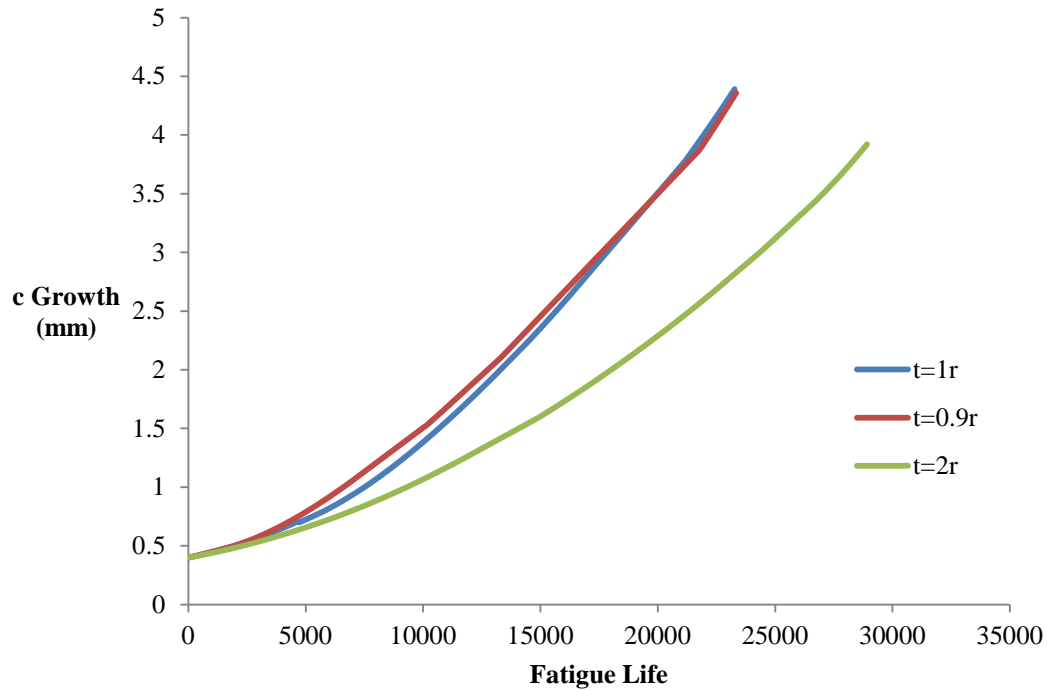


Figure 5.29: c growth vs. the number of cycles for three different plate thicknesses.

From the results, it can be concluded that for the applied loading and geometry cracks will propagate through the thickness faster if the thickness of the plate is reduced. This result was expected because the crack is required to travel a further distance in the thicker plate before it propagates through the thickness. An influence on crack propagation rates for the smaller thicknesses was not found indicating that further simulations should be done to analyse the effect of plate thickness on the crack propagation rates.

5.4.3 Countersink Angle

A fatigue analysis was completed to determine how the angle of the countersink affected the growth rate by comparing the c length versus the number of cycles for the cracks for three different angles. The results from simulation 4, 7 and 10 were used in the analysis. For simulation 4 the angle (γ) was 80° , for simulation 7 the angle (γ) was 100° and for simulation 10 the angle (γ) was 120° . The different countersink angles are shown in Figure 5.30. This simulation analyses the effects of changing

the countersink angle for initial knee cracks, although results were also considered for initial corner and upper cracks and will be summarized at the end of the section.

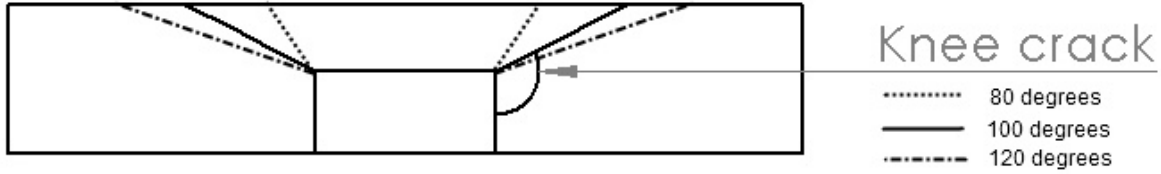


Figure 5.30: Three different countersink angles.

The crack in the plate with the largest countersunk angle propagated through the thickness the fastest followed by the crack in the plate with the second largest countersunk angle and finally the crack in the plate with the smallest countersunk angle as shown in Figure 5.31. It took 24% fewer cycles for the crack with $\gamma=120^\circ$ to reach a c value of 3.56mm compared to the crack with $\gamma=80^\circ$. It took 6.8% fewer cycles for the crack with $\gamma=100^\circ$ to reach a c value of 3.56mm compared to the crack with $\gamma=80^\circ$.

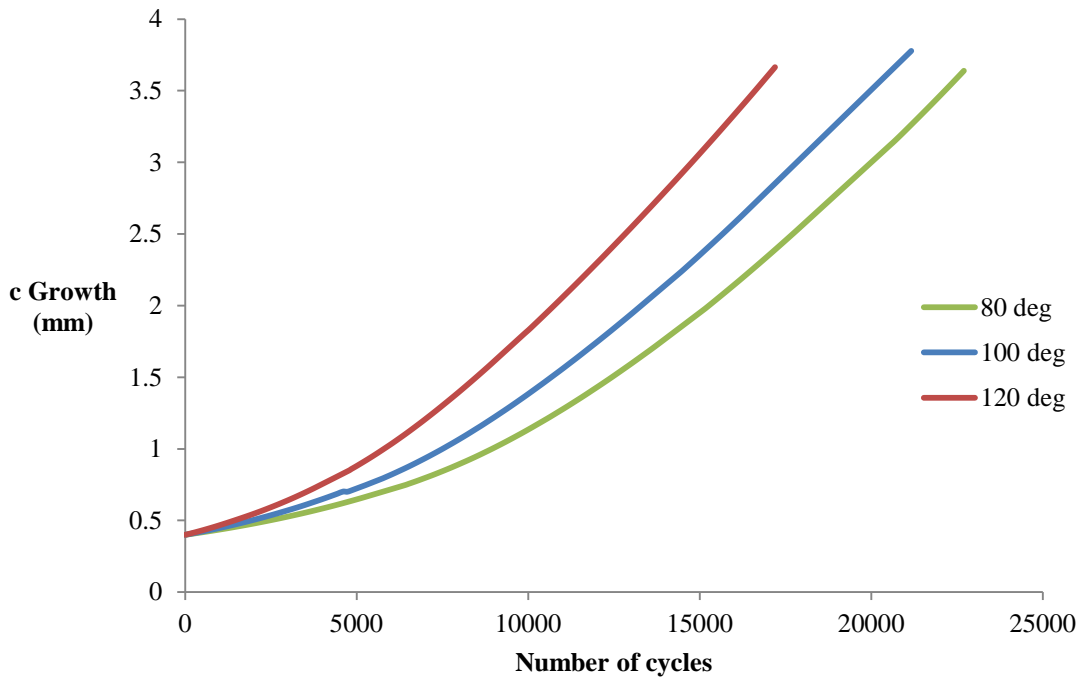


Figure 5.31: c growth vs. the number of cycles for three different countersink angles.

From the angle simulation it was found that knee cracks will propagate faster if the countersink angle is increased. This same result was found for corner cracks, but the opposite effect was found for upper cracks where cracks propagated faster if the angle was reduced. This result is in agreement with results found by Shivakumar et al. [52] regarding the stress concentrations along the countersink hole. They found that stress concentrations along the straight shank, from the lower surface to the intersection of the countersink, were increased if the angle of the countersink was increased, with the greatest increase occurring at the straight shank-countersink intersection. From the intersection of the straight shank, along the countersink, to the upper surface they found that stress concentrations were reduced if the angle of the countersink was reduced.

5.5 Discussion of the Countersunk Simulations

In general, the cracks emanating from countersunk holes quickly grew from their initial circular shape into an elliptical shape with the edges of the crack front displaying a tunneling effect close to the free surfaces. The influence of the geometric intersections (lower surface-straight shank, straight shank-countersink and countersink-upper surface), where higher stress concentrations are found, had an effect on the growth rate along the crack front. This influence could be seen when the cracks approached these geometric intersections by an increase in beta factors over the crack front closest to the intersections. An increase in the growth rate was also observed when the cracks propagated through a transition stage. An increase in beta factors was found along the crack front surface where the transition occurred due to the crack trying to propagate to a preferred pattern.

The conclusions made by Grandt Jr. [37] about the growth pattern of cracks in countersunk holes were in agreement with the predictions. Grandt Jr. concluded that corner cracks will grow along the countersink faster than the lower surface; this was observed with the ACGP results until a preferred shape was reached. A similar conclusion and agreement was made for a knee crack growing faster along the lower surface compared to the countersink.

The conclusion made by Lin and Smith [30], that cracks of different initial shapes will grow to a similar preferred pattern was confirmed when observing the growth pattern of the initial corner and initial knee cracks. The cracks were initiated in different locations but propagated to a similar preferred shape. The growth rate was the same when the cracks achieved the similar shape indicating that differences in fatigue life predictions are caused before the cracks have reached the similar iso-K state. The preferred shape is identified by an iso-K profile where the beta factors are constant along the crack front. This profile was reached by the corner and knee cracks as they propagated as type 2 cracks. It is assumed upper cracks will eventually reach an iso-K state as it propagates further through the thickness. There were slight variations in beta values near the edges for the iso-K cracks, but the variations were small which provides confidence in the programs ability to represent the crack shape at the free edges but future work should be done to examine the beta factors near the edges.

The knee crack was found to propagate through the thickness of the plate at the fastest rate. This indicates that for the geometry and loading considered in these simulations, the critical crack location for an initial crack is at the intersection of the countersink and the straight shank. This information is valuable for making accurate damage tolerance assessments as the initial location of the critical crack must be assumed.

Results from the geometry simulations comparing fatigue life, found that cracks grow faster when local stress are higher. The fatigue life patterns observed from the ACGP results followed the trends observed by Shivakumar et al. [52] for stress concentrations along the countersink. Based on the geometry studies, the fatigue life of a knee crack (critical initial location) emanating from the countersunk fastener hole can be improved by reducing the angle of the countersink, reducing the radius of the countersunk hole and increasing the thickness of the plate.

Chapter 6 Conclusions and Recommendations

6.1 Summary and Conclusions

The purpose of this thesis was to develop a tool that could be used to predict crack growth in non-standard geometries such as countersunk holes, where the shape of the crack front is unknown. The reason for undertaking this endeavour was to develop a program to predict fatigue life using commercially available finite element software without applying many of the assumptions used in current methods. The tool that was developed is called the Automated Crack Growth Program (ACGP). The accuracy of the ACGP in terms of fatigue life and predictions of crack front shape for mode 1 cracks in three-dimensional geometries was evaluated using experimental and accepted numerical results. The program was also utilized to provide insight into the growth and behaviour of cracks emanating from countersunk fastener holes in a plate.

The ability of the ACGP to accurately predict SIFs over the crack front, including near the free surfaces, was confirmed using accepted numerical results from the literature comparing two symmetrical cracks at a hole and comparing one shallow crack at a hole in a plate. The accuracy of the fatigue and growth predictions was confirmed through comparison with an experimental result containing an initial complex crack front. Comparisons of fatigue life predictions were also made between the commercial software AFGROW and the ACGP. AFGROW's predictions were shown to be conservative due to the crack front being fit to an ellipse.

The conclusions of the analyses completed for the validation of the program are as follows:

- 1) The ACGP accurately predicts growth over the entire crack front including the locations near the free surfaces.
- 2) Cracks generally propagate to a shape that is elliptical over the middle of the crack front with maximum growth occurring close to the free surfaces and a tunneling effect where growth is retarded near the free surfaces.

- 3) Using a spline to fit the crack front gives a better representation of the evolving crack front compared to an ellipse.
- 4) Crack growth was shown to occur in two distinct stages. During the first stage of growth cracks attempt to propagate to a preferred pattern, the second stage is identified as an iso-K state where the crack propagates evenly over the entire surface.
- 5) The predicted crack shape is influenced by the distribution of SIFs over the crack front rather than overall magnitude.
- 6) Differences in the magnitude of the Paris parameters do not have a large influence on the growth pattern, therefore the influence of Paris parameters on fatigue life can be analysed without performing a full run of the program.

For the countersunk studies, the accuracy of the numerical solutions was confirmed using a p-level convergence study. The analysis of cracks emanating from countersunk holes in a plate identified an initial knee crack to be the critical location for mode 1 loading, the different stages of crack growth in a plate were observed as well as the effects of countersink and plate geometries on fatigue life. When compared to the commercial software AFGROW, both programs predicted similar crack shapes as the cracks propagated through the material except after transition stages observed in the countersunk results due to the different methods used to grow cracks through transition. Differences were found in fatigue predictions between the ACGP and AFGROW. The conclusions of the analyses completed for cracks emanating from countersunk holes are as follows:

- 1) The portion of a crack front closest to a geometric intersection (e.g. intersection of the countersink with the straight shank) will propagate faster than other regions of the crack front due to the local stress concentrations. After transition of a crack through a geometric intersection the region of the crack front that transitioned over the intersection will propagate faster than other regions of the crack front in an attempt to achieve a preferred, iso-K shape.

- 2) An initial knee crack, located at the intersection of the countersink and the straight shank, is the critical initial location for a crack emanating from a countersunk hole for the geometry and loading investigated in this thesis.
- 3) Cracks emanating from countersunk fastener holes in a plate will propagate to through thickness faster for larger countersink radii and smaller plate thicknesses.
- 4) For initial corner and knee cracks a larger countersink angle will cause cracks to propagate to through thickness faster. The opposite behaviour is observed for initial upper cracks.

The ACGP is a very powerful tool for predicting fatigue life and crack growth and further investigation is required to optimise the program and validate the predictions for a variety of complex cases. It is anticipated that this program will be able to assist the National Research Council Canada (NRC) and Defence Research and Development Canada (DRDC) in performing damage tolerance studies.

6.2 Recommendations

The work outlined in this thesis has taken the ACGP from a preliminary state to a working program that can predict fatigue life and crack growth for mode I cracks emanating from features (notch, hole countersunk hole) in a plate. The current program provides a base for further advances to apply the program for propagation situations where there are no solutions. A list of recommendations is provided below on how to optimise and further validate the program for the current geometric configurations and how to advance the program for more complex analyses.

- 1) The capabilities of the program should be extended by incorporating a pin loading condition to represent the loads of a fastener and adding the ΔK modifiers to incorporate crack closure and residual stress solutions.
- 2) To improve computational times, the meshing and extraction stages of the ACGP should be optimised.

- 3) Additional analyses to validate the program with experimental results are suggested. It is important to test the results for different materials, loading and stress ratios as well as test the program with crack growth from countersunk experiments results.
- 4) Further investigation of certain aspects of the ACGP is recommended for study of the growth of the cracks through transition stages to determine the influence of the geometric intersection, the distribution of ΔK values along a crack front near the intersection of a free surface and how the shape and angle of the intersection affects these values. Additional simulations to determine the influence of plate thickness on cracks in countersunk holes should be performed.

References

- [1] ASM International, "Fracture Toughness and Fracture Mechanics," in *Mechanical Testing and Evaluation, Vol 8, ASM Handbook*, 2000, pp. 563-575.
- [2] D. Broek, *The Practical Use of Fracture Mechanics*, Netherlands: Kluwer Academic Publishers, 1989.
- [3] D. DuQuesnay, *ME535-Fatigue and Fracture L03*, 2012.
- [4] "Lecture 12.13: Fracture Mechanics Applied to Fatigue," European Steel Design Education Programme, [Online]. Available: <http://www.fgg.uni-lj.si/~pmoze/ESDEP/master/wg12/11300.htm>. [Accessed 08 08 2014].
- [5] M. Woodward and R. Stover, "Damage Tolerance, Composites, Vol21," in *ASM Handbook*, ASM International, 2001, pp. 295-301.
- [6] D. Rooke and D. Cartwright, *Compendium of Stress Intensity Factors*, London: Hillingdon Press, 1976.
- [7] Y. Murakami, *Stress Intensity Factors Handbook*, Oxford: Pergamon Press, 1987.
- [8] Air Force Research Laboratory, "DTD Handbook Principal of Superposition," LexTech, Inc, [Online]. Available: http://www.afgrow.net/applications/DTDHandbook/Sections/page11_2_1.aspx. [Accessed 10 09 2014].
- [9] S. Fawaz and B. Andersson, "Accurate Stress Intensity Factor Solutions for Corner Cracks at a Hole," *Engineering Fracture Mechanics*, pp. 1235-1254, 2004.
- [10] J. Newman and I. Raju, "An Empirical Stress-Intensity Factor Equation for the Surface Crack," *Engineering Fracture Mechanics*, pp. 185-192, 1981.
- [11] T. Nishioka and S. Atluri, "Analytical Solution for Embedded Elliptical Cracks, and Finite Element Alternating Method for Elliptical Surface Cracks Subjected to Arbitrary Loadings," *Engineering Fracture Mechanics*, pp. 247-268, 1983.
- [12] W. Zhao, X. Wu and M. Yan, "Weight Function Method for Three-Dimensional Crack Problems," *Engineering Fracture Mechanics*, pp. 593-607, 1989.
- [13] J. Bakuckas Jr, "Comparison of Boundary Correction Factor Solutions for Two Symmetric Cracks in a Straight-Shank Hole," *Engineering Fracture Mechanics*, pp. 1095-1106, 2001.
- [14] X. Lin and R. Smith, "Finite Element Modelling of Fatigue Crack Growth of Surface Cracked Plates

- Part I: The Numerical Technique," *Engineering Fracture Mechanics*, pp. 503-522, 1999.
- [15] C. Fernando, E. Menandro, J. Moyer and H. Liebowitz, "A Methodology for Crack Tip Mesh Design," *Engineering Fracture Mechanics*, pp. 713-726, 1995.
- [16] D. Wowk, K. Gamble and R. Underhill, "Influence of p-method finite element parameters on predictions of crack front geometry," *Department of Mechanical and Aerospace Engineering, Royal Military College of Canada*, 2013.
- [17] Engineering Software Research & Development, Inc, "StressCheck Master Guide Release 9.2," 2011.
- [18] J. Newman Jr and I. Raju, "Prediction of Fatigue Crack Growth Patterns and Lives in Three-Dimensional Cracked Bodies," in *Sixth International Conference on Fracture*, New Delhi, 1984.
- [19] R. Branco and F. Antunes, "Finite Element Modelling and Analysis of Crack Shape Evolution in Mode-I Fatigue Middle Cracked Tension Specimens," *Engineering Fracture Mechanics*, pp. 3020-3037, 2008.
- [20] W. Elber, "Fatigue Crack Closure Under Cyclic Tension," *Engineering Fracture Mechanics*, pp. 37-45, 1970.
- [21] C. Park, A. Grandt Jr. and J. Suh, "Stress Intensity Factors for Surface Cracks at Countersunk Holes," *Engineering Fracture Mechanics*, pp. 1878-1898, 2006.
- [22] S. Ray, R. Perez and A. Grandt Jr, "Fatigue Crack Closure of Corner Cracks Located at Holes Loaded in Tension or Bending," *Fatigue and Fracture of Engineering Materials and Structures*, pp. 239-250, 1987.
- [23] S. Forth, M. James, W. Johnson and J. Newman Jr, "Anomalous Fatigue Crack Growth Phenomena In High Strength Steel," in *International Congress on Fracture*, Italy, 2007.
- [24] J. LaRue and S. Daniewicz, "Predicting the Effect of Residual Stress on Fatigue Crack Growth," *International Journal of Fatigue*, pp. 508-515, 2007.
- [25] M. Gilchrist and R. Smith, "Finite Element Modelling of Fatigue Crack Shapes," *Fatigue and Fracture of Engineering Materials and Structures*, pp. 617-626, 1991.
- [26] D. Child, N. Moyle and A. Grandt, "Experimental Validation Of Stress Intensity Factor Solutions For The Pin Loaded Lug," in *ICAF 2009, Bridging the Gap between Theory and Operational Practice*, 2009, pp. 871-898.
- [27] M. Sevcik, P. Hutar, M. Zouhar and L. Nahlik, "Numerical Estimation of the Fatigue Crack Front Shape for a Specimen with Finite Thickness," *International Journal of Fatigue*, pp. 75-80, 2012.

- [28] X. Lin and R. Smith, "Fatigue Shape Analysis for Corner Cracks at Fastener Holes," *Engineering Fracture Mechanics*, pp. 73-87, 1998.
- [29] M. Gozin and M. Aghaie-Khafri, "Fatigue Crack Growth Prediction of Elliptical Corner Crack using 3D FEM," *Applied Mechanics and Materials*, pp. 469-474, 2013.
- [30] X. Lin and R. Smith, "Finite Element Modelling of Fatigue Crack Growth of Surface Cracked Plates Part III: Stress Intensity Factor and Fatigue Crack Growth Life," *Engineering Fracture Mechanics*, pp. 541-556, 1999.
- [31] E. Favier, V. Lazarus and J. Leblond, "Coplanar Propagation Paths of 3D Cracks in Infinite Bodies Loaded in Shear," *International Journal of Solids and Structures*, pp. 2091-2109, 2006.
- [32] N. Couroneau and J. Royer, "Simplified Model for the Fatigue Growth Analysis of Surface Cracks in Round Bars Under Mode 1," *International Journal of Fatigue*, pp. 711-718, 1998.
- [33] A. Skorupa and M. Skorupa, *Riveted Lap Joints in Aircraft Fuselage, Solid Mechanics and Its Applications*, Dordrecht: Springer Science+Business Media, 2012.
- [34] J. de Rijck and S. Fawaz, "Stress Intensity Factor Solutions for Countersunk Holes Subjected to Combined Loading," in *The 7th Joint DOD/FAA/NASA Conference on Aging Aircraft*, New Orleans, 2003.
- [35] O. Partl and J. Schijve, "Multiple Site Damage in 2024 T3 Alloy Sheet, Report LR660," Faculty of Aerospace Engineering, 1992.
- [36] C. Park and A. F. Grandt Jr, "Effect of Load Transfer on the Cracking Behaviour at a Countersunk Fastener Hole," *International Journal of Fatigue*, pp. 146-157, 2007.
- [37] J. A. Grandt, "Analysis of Fatigue Crack Growth from Countersunk Fastener Holes," U.S. Department of Transportation Federal Aviation Administration, Washington, 2008.
- [38] S. Fawaz and J. Schijve, "Fatigue Crack Growth Predictions in Riveted Joints," in *2nd Joint NASA/FAA/DOD Conference on Aging Aircraft*, Williamsburg, VA, 1998.
- [39] G. Shi and G. Li, "Modelling and Analyses of Cracks in Fuselage Lap Joints with a Single Countersunk Rivet," in *Computational Methods*, Springer, 2006, pp. 1715-1725.
- [40] B. Andersson and U. Falk, "Reliable Stress and Fracture Mechanics Analysis of Complex Components using a h-p Version of FEM," *International Journal for Numerical Methods in Engineering*, pp. 2135-2163, 1995.
- [41] W. Anderson and L. James, "A Simple Experimental Procedure for Stress Intensity Factor Calibration," *Engineering Fracture Mechanics*, pp. 565-568, 1969.

- [42] S. Fawaz and D. Hill, "Validation of Stress Intensity Factors of Diametrically Opposed Corner Cracks in a Hole," *International Journal of Fatigue*, pp. 712-718, 2009.
- [43] R. Desnoyers and M. Liao, "Crack Growth Measurements Report on SENT Coupon J407-1E-2," NRC CNRC, 2011.
- [44] J. Harter, AFGROW Users Guide and Technical Manual, Air Force Research Laboratory, 2008.
- [45] A. Grandt and D. Macha, "Digitized Measurements of the Shape of Corner Cracks at Fastener Holes," *Engineering Fracture Mechanics*, 1983.
- [46] J. Newman Jr, "Fracture Mechanics Parameters for Small Fatigue Cracks," in *Small-Crack Test Methods, ASTM STP 1149*, Philadelphia, American Society for Testing and Materials, 1992, pp. 6-33.
- [47] J. Harter, AFGROW Users Guide and Technical Manual, Air Force Research Laboratory, 2014.
- [48] R. Guillaume, M. Liao, Y. Bombardier, R. Desnoyers and W. Beres, "Three-Dimensional Modelling of Small Cracks in 7075-T73 Forging Specimens," Institute for Aerospace Research, National Research Council Canada, Ottawa.
- [49] R. Rice, J. Jackson, J. Bakuckas and S. Thompson, "Metallic Materials Properties Development and Standardization," U.S. Department of Transportation Federal Aviation Administration, 2003.
- [50] R. De Rijck, "Stress Analysis of Fatigue Cracks in Mechanically Fastened Joints," Ph.D. Thesis, Delft University, 2005.
- [51] M. Heyder and G. Kuhn, "3D Fatigue Crack Propagation: Experimental Studies," *International Journal of Fatigue*, pp. 627-634, 2006.
- [52] K. Shivakumar, A. Bhargava and J. Newman Jr, "Tensile Stress Concentration Equations for Countersunk Holes," *Journal of Aircraft*, pp. 194-200, 2007.
- [53] K. Jones and M. Dunn, "Predicting Corner Crack Fatigue Propagation from Cold Worked Holes," *Engineering Fracture Mechanics*, pp. 2074-2090, 2009.
- [54] X. Lin and R. Smith, "Finite Element Modelling of Fatigue Crack Growth of Surface Cracked Plates Part II: Crack Shape Change," *Engineering Fracture Mechanics*, pp. 523-540, 1999.
- [55] J. Harter, AFGROW Users Guide and Technical Manual, Air Force Research Laboratory, 2008.
- [56] J. Gere, *Mechanics of Materials*, Brooks/Cole Thomas Learning, 2001.
- [57] S. Suresh, *Fatigue of Materials*, Cambridge University Press, 1998.

Appendices

Appendix A Initial Crack Front Geometries

This appendix lists the initial crack geometries input into the ACGP for each simulation that was run. The names of the tables correspond to the section titles of the thesis where the simulation was presented. The values from each table are input into the ACGP as seen in Figure A.1. The ACGP requires the coordinates to be input in inches for version 1 and 2 of the program but tables A.1 – A.4 are shown in mm for consistency in this document.

Crack Front		
le	x(mm)	y(mm)
	0.47630043	0
	0.476204275	0.0059879
	0.466765374	0.0558143
	0.463886826	0.0615053
	0.460724704	0.0670441
	0.45726088	0.072407
	0.453490113	0.0775673
	0.411773126	0.1170045
	0.406470804	0.1207052
	0.3695	0.1377573

Plate w Hole

Figure A.1 Location where the initial crack geometry is input into the ACGP

The order of the points from first to last for each initial crack are shown in Figure A.2.

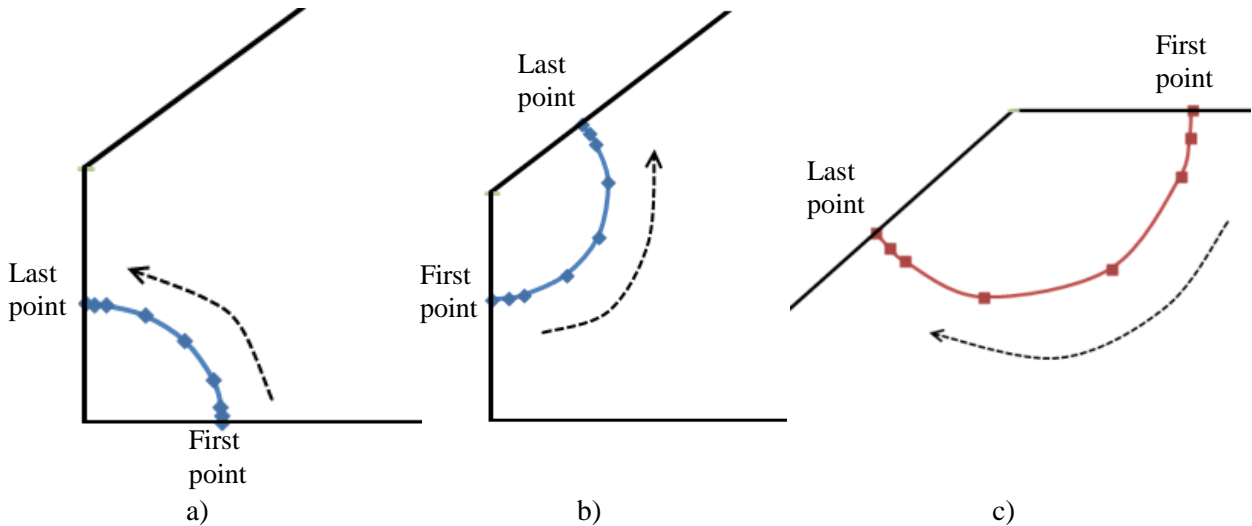


Figure A.2: Order of points along the crack front to be input during pre-processing for a) a corner crack b) a knee crack c) an upper crack.

Table A.1: Initial geometry used for the simulation from Section 4.1.1 Two symmetric Corner Cracks Emanating from a Hole in a Plate

x (mm)	y(mm)
28.575	0
28.57246	0.10158
28.56992	0.143627
28.5623	0.226957
28.55468	0.286908
28.5496	0.320645
28.5369	0.392313
28.5242	0.452547
28.448	0.7112
28.194	1.206434
27.94	1.524
27.686	1.762694
27.432	1.951673
27.178	2.104372
26.924	2.228261
26.67	2.327948
26.416	2.406442
26.162	2.465763
25.908	2.507277
25.781	2.521646
25.7556	2.524018
25.7302	2.526226
25.7048	2.528268
25.6794	2.530145
25.654	2.531859
25.6286	2.533409
25.5524	2.537071
25.5016	2.5387
25.4508	2.539675
25.4	2.54

Table A.2: Initial geometry used for the simulation from Section 4.1.2 A single Corner Crack Emanating from a Hole in a Plate

x(mm)	y(mm)
101.6	0
101.5997	0.016066
101.5975	0.0508
101.5873	0.113586
101.5746	0.160625
101.5492	0.227127
101.4984	0.321127
101.4476	0.3932
101.3968	0.453913
101.346	0.507365
101.092	0.716623
100.584	1.010907
100.33	1.128801
99.06	1.58623
96.52	2.214321
93.98	2.676055
88.9	3.360105
81.28	4.064
76.2	4.39941
73.66	4.536585
71.12	4.655896
68.58	4.758687
66.04	4.84601
63.5	4.91869
60.96	4.977364
58.42	5.022525
55.88	5.054537
53.34	5.073645
53.086	5.074854
52.832	5.07492
52.578	5.07746
52.324	5.07746
52.07	5.07746
51.816	5.08
51.562	5.08
51.308	5.08
51.054	5.08
50.8	5.08

Table A.3: Initial geometry used for the simulation from Sections 4.2

x(mm)	y(mm)
9.574225	0
9.574225	0.000288
9.572667	0.071392
9.572536	0.074343
9.572424	0.076772
9.572255	0.080274
9.572087	0.08363
9.571368	0.096672
9.57058	0.109174
9.569647	0.122337
9.56855	0.136182
9.567275	0.150675
9.56582	0.165659
9.564197	0.180901
9.562425	0.196185
9.5605	0.211519
9.558504	0.226311
9.556341	0.241295
9.554042	0.256243
9.536831	0.347866
9.514371	0.438574
9.486766	0.527868
9.454097	0.61547
9.416455	0.701114
9.373998	0.784438
9.326773	0.865327
9.275103	0.943234
9.219088	1.018037
9.158864	1.089558
9.094661	1.157519
9.026694	1.221689
8.955193	1.281862
8.880271	1.337927
8.802395	1.389543

Table A3 continued

8.721701	1.436627
8.638423	1.479045
8.552841	1.516649
8.465294	1.549291
8.376009	1.576888
8.285308	1.599344
8.193486	1.616583
8.175849	1.619278
8.169026	1.620266
8.164932	1.620848
8.133607	1.624935
8.088459	1.629748
8.043312	1.633301
7.997949	1.635608
7.952637	1.636654
7.9375	1.63576

Table A.4: Initial geometry used for the simulation from Section 4.4 Comparison with Experimental

x(mm)	y(mm)
2.184123	0
2.183877	0.035408
2.222352	0.13051
2.288277	0.225831
2.358153	0.317233
2.44696	0.495684
2.5123	0.666046
2.56992	0.820552
2.604013	0.97487
2.619083	1.053988
2.621344	1.267303
2.624096	1.417424
2.59168	1.551468
2.562939	1.717139
2.542871	1.77623
2.487019	1.898236
2.451112	1.976953
2.395352	2.08711
2.324061	2.177397
2.225197	2.283264
2.189598	2.322485
2.114816	2.357443
2.063316	2.42316
2.008079	2.46634
1.968681	2.49174
1.897913	2.5146
1.823254	2.53238
1.7018	2.54508

Table A.5: Initial geometry used for the simulations from Chapter 5 for simulations 1, 2 and 3

x(mm)	y(mm)
3.6639	0.0000
3.6608	0.0500
3.6512	0.1000
3.6347	0.1500
3.6103	0.2000
3.5761	0.2500
3.5285	0.3000
3.5039	0.3200
3.4383	0.3600
3.3888	0.3800
3.2639	0.4000

Table A.6: Initial geometry used for the simulation from Chapter 5 for simulation 4

x(mm)	y(mm)
3.2639	0.4509
3.2939	0.4520
3.3339	0.4571
3.3639	0.4636
3.4639	0.5045
3.5639	0.5863
3.6039	0.6402
3.6439	0.7260
3.6639	0.8509
3.6439	0.9758
3.6139	1.0445
3.5939	1.0770
3.5639	1.1155
3.5139	1.1631
3.4639	1.1973
3.4139	1.2217
3.3639	1.2382
3.3139	1.2478
3.2939	1.2498
3.2639	1.2509

Table A.7: Initial geometry used for the simulations from Chapter 5 for simulations 5, 7 and 9

x(mm)	y(mm)
3.2639	0.1245
3.2939	0.1256
3.3339	0.1307
3.3639	0.1372
3.4639	0.1781
3.5639	0.2599
3.6039	0.3138
3.6439	0.3996
3.6639	0.5245
3.6439	0.6494
3.6139	0.7182
3.5939	0.7506
3.5639	0.7891
3.5139	0.8368
3.4639	0.8709
3.4139	0.8953
3.3639	0.9118
3.3139	0.9214
3.2939	0.9234
3.2639	0.9245

Table A.9: Initial geometry used for the simulation from Chapter 5 for simulation 8

x(mm)	y(mm)
2.0193	0.4509
2.0493	0.4520
2.0893	0.4571
2.1193	0.4636
2.2193	0.5045
2.3193	0.5863
2.3593	0.6402
2.3993	0.7260
2.4193	0.8509
2.3993	0.9758
2.3693	1.0445
2.3493	1.0770
2.3193	1.1155
2.2693	1.1631
2.2193	1.1973
2.1693	1.2217
2.1193	1.2382
2.0693	1.2478
2.0493	1.2498
2.0193	1.2509

Table A.8: Initial geometry used for the simulation from Chapter 5 for simulation 6

x(mm)	y(mm)
2.4257	0.4509
2.4557	0.4520
2.4957	0.4571
2.5257	0.4636
2.6257	0.5045
2.7257	0.5863
2.7657	0.6402
2.8057	0.7260
2.8257	0.8509
2.8057	0.9758
2.7757	1.0445
2.7557	1.0770
2.7257	1.1155

Table A.10: Initial geometry used for the simulation from Chapter 5 for simulation 10

x(mm)	y(mm)
3.2639	0.4509
3.2939	0.4520
3.3339	0.4571
3.3639	0.4636
3.4639	0.5045
3.5639	0.5863
3.6439	0.7260
3.6639	0.8509
3.6439	0.9758
3.6139	1.0445
3.5939	1.0770
3.5639	1.1155
3.5139	1.1631
3.4639	1.1973
3.4139	1.2217
3.3639	1.2382
3.3139	1.2478
3.2939	1.2498
3.2639	1.2509

Table A.11: Initial geometry used for the simulation from Chapter 5 for simulation 11

x(mm)	y(mm)
5.6886	3.2639
5.6836	3.2009
5.6686	3.1390
5.6486	3.0895
5.6286	3.0532
5.6086	3.0239
5.5886	2.9993
5.5686	2.9782
5.5486	2.9599
5.5286	2.9439
5.5086	2.9298
5.4886	2.9175
5.4686	2.9067
5.4486	2.8973
5.4286	2.8892
5.4086	2.8823
5.3886	2.8766
5.3686	2.8720
5.3486	2.8684
5.3286	2.8659
5.3086	2.8644
5.2886	2.8639
5.2686	2.8644
5.2486	2.8659
5.2286	2.8684
5.2086	2.8720
5.1886	2.8766
5.1686	2.8823
5.1486	2.8892
5.1286	2.8973
5.1086	2.9067
5.0886	2.9175
5.0686	2.9298
5.0486	2.9439
5.0286	2.9599
5.0086	2.9782
4.9886	2.9993
4.9686	3.0239
4.9486	3.0532
4.9286	3.0895

Table A.12: Initial geometry used for the simulation from Chapter 5 for simulation 12

x(mm)	y(mm)
6.5396	3.2639
6.5330	3.1915
6.5196	3.1390
6.4996	3.0895
6.4796	3.0532
6.4596	3.0239
6.4396	2.9993
6.4196	2.9782
6.3996	2.9599
6.3796	2.9439
6.3596	2.9298
6.3396	2.9175
6.3196	2.9067
6.2996	2.8973
6.2796	2.8892
6.2596	2.8823
6.2396	2.8766
6.2196	2.8720
6.1996	2.8684
6.1796	2.8659
6.1596	2.8644
6.1396	2.8639
6.1196	2.8644
6.0996	2.8659
6.0796	2.8684
6.0596	2.8720
6.0396	2.8766
6.0196	2.8823
5.9996	2.8892
5.9796	2.8973
5.9596	2.9067
5.9396	2.9175
5.9196	2.9298
5.8996	2.9439
5.8796	2.9599
5.8596	2.9782
5.8396	2.9993
5.8196	3.0239
5.7996	3.0532
5.7796	3.0895

Table A.13: Initial geometry used for the simulation from Chapter 5 for simulation 13

x(mm)	y(mm)
7.8433	3.2639
7.8383	3.2009
7.8233	3.1390
7.8033	3.0895
7.7833	3.0532
7.7633	3.0239
7.7433	2.9993
7.7233	2.9782
7.7033	2.9599
7.6833	2.9439
7.6633	2.9298
7.6433	2.9175
7.6233	2.9067
7.6033	2.8973
7.5833	2.8892
7.5633	2.8823
7.5433	2.8766
7.5233	2.8720
7.5033	2.8684
7.4833	2.8659
7.4633	2.8644
7.4433	2.8639
7.4233	2.8644
7.4033	2.8659
7.3833	2.8684
7.3633	2.8720
7.3433	2.8766
7.3233	2.8823
7.3033	2.8892
7.2833	2.8973
7.2633	2.9067
7.2433	2.9175
7.2233	2.9298
7.2033	2.9439
7.1833	2.9599
7.1633	2.9782
7.1433	2.9993
7.1233	3.0239
7.1033	3.0532
7.0833	3.0895

Appendix B – Visual Basic Code for the ACGP (Version 3)

This appendix contains the Visual Basic code used to create the ACGP. The program is split into four modules: module 1 contains the code to run the program for one iteration, module 2 contains the code to run the program for a full test, module 3 contains the code to terminate the StressCheck application and module 4 is contains the post-processing code.

```
'Crack Path Analysis program.
'Completed by Lucas Alousis August 2014 using existing
code from Kyle Gamble and Ross Underhill
'This program passes information between Microsoft Excel
and StressCheck to perform Crack Growth analyses.
'The program accepts geometrical information from Excel,
produces the geometry in StressCheck and meshes and
'applies the material properties, loads and boundary
conditions. Once the stress intensity factors are determined
'they are sent back to excel for post-processing to determine
the beta factors and draw the shape of the crack front.
```

Option Explicit 'declare all variables

```
'-----GLOBAL
VARIABLES-----
-
```

```
'File Handling objects
Public FSO As FileSystemObject
Public filename As String
Public newcrack() As String
Public crackpts(2) As Integer
```

```
'StressCheck Collections & Entities
Public SCApp As StressCheck.Application
Public SCDoc As StressCheck.Document
Public SCModel As StressCheck.Model
Public SCSystems As StressCheck.Systems
Public SCPlanes As StressCheck.Planes
Public SCPoints As StressCheck.Points
Public SCSplines As StressCheck.SplineCurves
Public SCLines As StressCheck.Lines
Public SCBodies As StressCheck.Bodies
Public SCSets As StressCheck.Sets
Public SCAutomeshes As StressCheck.Automeshes
Public SCConsts As StressCheck.Constraints
Public SCLoads As StressCheck.Loads
Public ScMatDefs As StressCheck.Materials
Public SCMatAssigns As
StressCheck.MaterialAssignments
Public SCSols As StressCheck.Solutions
```

```
Public SCPlots As StressCheck.Plots
Public SCEextractions As StressCheck.Extractions
Public SCEExtractionSettings As
StressCheck.ExtractionSettings
Public SCBoxes As StressCheck.Boxes
Public SCCylinders As StressCheck.Cylinders
Public SCCones As StressCheck.Cones
Public SCCons As StressCheck.Constraints
Public SCMaterials As StressCheck.Materials
Public SCMaterialAssignments As
StressCheck.MaterialAssignments
Public SCSolidIDs As StressCheck.SolutionIDs
Public SCLinearSols As StressCheck.Solutions
Public SCEllipses As StressCheck.EllipseCurves
Public SCComposites As StressCheck.CompositeCurves
```

```
'Variables that are used in various parts of the program
Dim lngSplinePoints() As Long 'List of Spline Points
Dim dblCrackLength As Double 'Estimate of length of
crackfront. Calculated in MakeCrack and used in
MeshModel
Dim intSplinePoints As Integer 'number of spline points.
Calculated in MakeCrack and used in MeshModel
Dim dblContourRadius As Double 'used in the extraction of
k values calculated in MeshModel and used in ExtractK
Dim rots(2) As Variant 'Used for rotations
```

```
'Utility Variables
Dim X As Double, Y As Double, z As Double
```

```
'Indices Variables
Dim i As Long, j As Long, k As Long, l As Long, s As
Long
```

```
Dim m As String, n As String
```

```
'-----Main Program---
```

```
Sub CrackInitiation(iteration As Integer, path As String)
```

```

' Turn off alerts such that the simulation will run
smoothly
Application.DisplayAlerts = False

'Initialize the StressCheck application
InitializeStressCheck
SCModel.StandardUnits = unitInchLbSec
If ActiveSheet.Range("Units") = "in" Then
SCModel.StandardUnits = unitInchLbSec
Else
If ActiveSheet.Range("Units") = "mm" Then
SCModel.StandardUnits = unitMMNewtonSec
Else
End If
End If

'Create the plate with hole
CreatePlate

'Add the crack to the model.
'Note: We pass the current body by reference. This
reference is updated in the MakeCrack
'routine to point to the cracked body. in principle this
could be called multiple times to
'insert multiple cracks.
Dim PlateBody As StressCheck.Body
Set PlateBody =
SCModel.Bodies.Body(SCModel.Bodies.LastNumber)
SCModel.Update

MakeCrack PlateBody

'Mesh the model

MeshModel

'Need some code to determine if the meshing worked

' Dim SCSet As StressCheck.Set
' For Each SCSet in SCSets
' If SCSet.Name = "_CONVERT_FAIL" Then ' check
to see if the mesh converted to geometry successfully
' Err.Raise vbObjectError + 1, , "Unable to convert
mesh"
' End If
' Next SCSet

'Apply the Loads and Constraints
AssignLoads
AssignConstraints

'Assign Materials
DefineMaterials
AssignMaterials

'Prepare the solution
PrepSolution
iteration = CStr(iteration)
SCDoc.Model.Write path & iteration & ".sci"

'Solve the problem
Dim plevel As Integer

```

```

plevel = ActiveSheet.Range("plevel").Value
SolveLinear (plevel)

'Extract the stress intensity factors (K1)
ExtractK

InitialAnalysis

CrackAnalysis

' StressCheck must be closed at the document and
application level
' Release SCModel model object
Set SCModel = Nothing

' Close and release the SCDoc database object
SCDoc.Close
DoEvents
Set SCDoc = Nothing

' Close and release SCApp StressCheck application object
SCApp.Close False
Set SCApp = Nothing
Application.Wait (Now + TimeValue("00:00:05"))

TerminateStressCheck

'Turn on alerts after simulation complete for when
manually examining the data
Application.DisplayAlerts = False

End Sub

'-----InitializeStressCheck
Routine-----

Sub InitializeStressCheck()

'Start a new copy of StressCheck and initialize the
StressCheck collections.
'The database will reside in this dataset folder.

Set FSO = New FileSystemObject

filename = ThisWorkbook.path & "\test1"
If FSO.FolderExists(filename) = True Then
FSO.DeleteFolder (filename)
End If
DoEvents
FSO.CreateFolder (filename)

'The following code makes SCApp a new StressCheck
application (StressCheck.exe should appear in the process
list in your task manager)
'Note: Make sure that there are no other StressCheck.exe
listings in the process list. End the process if there are.

Set SCApp = New StressCheck.Application

'The following code is used to create a new database
called 'test and belonging to SCApp.

```

'Note: in this case, the "With" statement is used to modify SCApp at the Document (database) hierarchy.
 'Document object commands including deleting (Document.Delete) and opening new or existing documents (Document.Open)

With SCApp
 'Open a new 'test' database and assign it to the SCApp Document (database) object
 'Note: the following Do..Loop code is only necessary for StressCheck v8.0 and previous

```
Do While SCApp.Document.Name = ""
    .Document.Open filename & "\test"
Loop
End With
```

'Instantiate a StressCheck Document (database) Object (SCDoc)
 'Assign SCDoc to be the SCApp Document (database) object
 Set SCDoc = SCApp.Document
 'This last action was performed such that we do not have to use "SCApp.Document" when we can simply use "SCDoc"

'Now, SCDoc will have all the properties that SCApp.Document has

```
'Instantiate a STressCheck Model Object (SCModel)
'Assign SCModel to be the SCDoc Model Object
Set SCModel = SCDoc.Model
'Systems Collection
Set SCSystems = SCModel.Systems
'Planes Collection
Set SCPlanes = SCModel.Planes
'Points Collection
Set SCPoints = SCModel.Points
'Splines Collection
Set SCSplines = SCModel.SplineCurves
'Lines Collection
Set SCLines = SCModel.Lines
'Bodies Collection
Set SCBodies = SCModel.Bodies
'Set Collection
Set SCSets = SCModel.Sets
'Automesh Collection
Set SCAutomeshes = SCModel.Automeshes
'Boxes Collection
Set SCBoxes = SCModel.Boxes
'Cylinders Collection
Set SCCylinders = SCModel.Cylinders
'Cones Collection
Set SCCones = SCModel.Cones
'Constraints Collection
Set SCCons = SCModel.Constraints
'Loads Collection
Set SCLoads = SCModel.Loads
'Material Assignments Collection
Set SCMaterialAssignments =
SCModel.MaterialAssignments
'Ellipse Curve Collection
Set SCEllipses = SCModel.EllipseCurves
'Composite Curve Collection
```

```
Set SCComposites = SCModel.CompositeCurves
'Solution IDs Collection
Set SCSolIDs = SCModel.SolutionIDs
'Show StressCheck
SCApp.Show True
```

End Sub

'-----CreatePlate Routine-----

```
Sub CreatePlate()
    Dim tools() As Long
    Dim SCSystem As StressCheck.System
    Dim SCSystem2 As StressCheck.System
    Dim SCSystem3 As StressCheck.System
    Dim PlateBody As StressCheck.Body
    Dim PWidth As Double, PLength As Double, PDepth As
    Double
    Dim holex As Double, holey As Double, holez As
    Double
    Dim HRadius As Double, HHHeight As Double
    Dim CR1 As Double, CR2 As Double, CHeight As
    Double, CSinky As Double
```

'This routine creates the geometry of the plate with a hole. Error checking will be added to ensure the hole location is

'entirely within the domain of the plate.

```
rots(0) = -90
rots(1) = 0
rots(2) = 0
```

```
Set SCSystem = SCSystems.AddGlobal(1, stCartesian,
0, 0, 0, rots)
SCModel.Update
```

```
'Create the solid box defining the plate
PWidth = ActiveSheet.Range("PWidth").Value
PLength = ActiveSheet.Range("PLength").Value
PDepth = ActiveSheet.Range("PDepth").Value
SCBoxes.Add 1, SCSystem.Number, PWidth, PLength,
PDepth
```

```
SCModel.Update
'Create the solid cylinder defining the hole
'Create a system for the cylinder
holex = ActiveSheet.Range("holex").Value
holey = ActiveSheet.Range("holey").Value
holez = ActiveSheet.Range("holez").Value
```

```
rots(0) = -90
rots(1) = 0
rots(2) = 0
```

```
Set SCSystem2 = SCSystems.AddGlobal(2, stCartesian,
holex, holey, holez, rots)
SCModel.Update
```

```
HRadius = ActiveSheet.Range("HRadius").Value
HHHeight = ActiveSheet.Range("HHHeight").Value
```



```

SCCylinders.Add 2, SCSystem2.Number, HRRadius,
HHeight, True
SCModel.Update

CSinky = ActiveSheet.Range("csinky").Value

Set SCSystem3 = SCSystems.AddGlobal(3, stCartesian,
holex, CSinky, holez, rots)
SCModel.Update

CR1 = ActiveSheet.Range("HRadius").Value
CR2 = (ActiveSheet.Range("HHeight").Value -
ActiveSheet.Range("csinky").Value) / (Tan((90 -
(ActiveSheet.Range("CSinkDeg").Value / 2)) / (180 /
3.14159265359))) + CR1
CHHeight = ActiveSheet.Range("HHeight").Value -
ActiveSheet.Range("csinky").Value

SCCones.Add 3, SCSystem3.Number, CR1, CR2,
CHHeight, True
SCModel.Update

'Add the cylinder number to a list of tools("tools") which
will be used for boolean-subtraction operation
ReDim tools(1)
tools(0) = 2
tools(1) = 3
Set PlateBody = SCModel.Bodies.AddBoolean(0,
bodySubtract, 1, 2, tools)

SCModel.Display.Orientation = vtIsometric
SCModel.Display.Orientation = vtCentre
SCModel.Update

'Create the solid Cone defining the countersunk hole
'Using the system from the hole

'CR1 = ActiveSheet.Range("HRadius").Value
'CR2 = (ActiveSheet.Range("HHeight").Value -
ActiveSheet.Range("csinky").Value *
(Tan(ActiveSheet.Range("CSinkDeg").Value / 2))) + CR1
'CHHeight = ActiveSheet.Range("csinky").Value

" SCCones.Add 3, SCSystem2.Number, CR1, CR2,
CHHeight, True
' SCModel.Update

'SCModel.Cones.Add 1, SCSystem2.Number, CR1, 0.78,
CHHeight, True
' Set SCSystem2 = SCSystems.AddGlobal(2, stCartesian,
holex, holey, holez, rots)
' SCModel.Cones.Add 1, SCSystem2.Number, CR1, 0.78,
CHHeight, True
'Add the cylinder number to a list of tools("tools") which
will be used for boolean-subtraction operation
ReDim tools(0)
tools(0) = 3
***** Set PlateBody =
SCModel.Bodies.AddBoolean(0, bodySubtract, 1, 1, tools)
' SCModel.Display.Orientation = vtIsometric

```

```

' SCModel.Display.Orientation = vtCentre
' SCModel.Update

End Sub

'-----MakeCrack
Routine-----

Sub MakeCrack(ByRef PlateBody As StressCheck.Body)

'This routine puts a crack in the body using data in the
spreadsheet. The structure of the spreadsheet has certain
restrictions.
'There must be a cell named "Crack0", which contains
the x co-ordinate of the origin of the crack
'The next two cells to the right contain the Y and Z co-
ords of the crack origin.

'It is assumed that the plane containing the crack is
oriented such that the crack will grow in the
'local positive X direction. The crack can grow in either
(or both) Y directions

'Rotations are not order independent!
'They are applied in the order 1) rotation about global Z,
2) rotation about Global Y, 3) rotation about Global X

'There must be a cell named "CrackPlane" that contains
the rotation about the x axis
'Immediately to the right of this is the rotation about the
Y axis and then, in the next cell to the right, rotation about
the Z axis.

'There must be a cell called PlaneSize. It gives the local
X dimension of the crack plane.
'Immediately to the right must be a cell containing the Y
dimension of the crack plane.
'The X and Y size MUST be large enough to completely
contain the crack at its largest extent.

On Error Resume Next

Dim CrackSys As Long
Dim SX As Double, SY As Double

X = CDBl(ActiveSheet.Range("PWidth").Value) / 4
Y = CDBl(ActiveSheet.Range("Crack0").Cells(1,
2).Value)
z = CDBl(ActiveSheet.Range("Crack0").Cells(1,
3).Value)
SX = CDBl(ActiveSheet.Range("PlaneSize").Cells(1,
1).Value)
SY = CDBl(ActiveSheet.Range("PlaneSize").Cells(1,
2).Value)

Dim rots(2) As Variant

'Rotations (about Global X, Y, Z)
rots(0) =
CDBl(ActiveSheet.Range("CrackPlane").Cells(1, 1).Value)

```

```

rots(1) =
Cdbl(ActiveSheet.Range("CrackPlane").Cells(1, 2).Value)
rots(2) =
Cdbl(ActiveSheet.Range("CrackPlane").Cells(1, 3).Value)
CrackSys = SCSys.LastNumber + 1
SCSystems.AddGlobal CrackSys, stCartesian, X, Y, z,
rots
SCModel.Update

' Add the plane
' When the plane is added in the next step, 5 boundaries
will be created, the plane surface and the 4 lines
' that form its edges. We will need the plane for later in
the Create>Body-Imprint step

Dim CrackPlane As Long, NextPlane As Long
CrackPlane = SCModel.LastBoundaryNumber + 1
NextPlane = SCPlanes.LastNumber + 1
SCPlanes.AddLocal NextPlane, CrackSys, SX, SY
SCModel.Update

Dim SplineSys As Long

X = 0
Y = 0
z = 0

```

```

'Rotations (about Global X, Y, Z)
rots(0) =
Cdbl(ActiveSheet.Range("CrackPlane").Cells(1, 1).Value)
rots(1) =
Cdbl(ActiveSheet.Range("CrackPlane").Cells(1, 2).Value)
rots(2) = 0
SplineSys = SCSys.LastNumber + 1
SCSystems.AddGlobal SplineSys, stCartesian, X, Y, z,
rots
SCModel.Update

```

```

Dim FirstSplinePoint As Integer 'number of first spline
point minus 1 in the points collection
Dim r As Double
Dim oldx As Double, oldy As Double
Dim PData As Range
Set PData =
ActiveSheet.Range("CrackFront").CurrentRegion

```

```

intSplinePoints = PData.Rows.Count - 1 '-1 to take into
account the header row

```

```

ReDim lngSplinePoints(intSplinePoints - 1)
FirstSplinePoint = SCPoints.LastNumber
dblCrackLength = 0 'used to obtain an estimate of the
cracklength
For i = 1 To intSplinePoints
If ActiveSheet.Range("CrackFront").Cells(i, 1).Value
> 0 Then
X =
Cdbl(ActiveSheet.Range("CrackFront").Cells(i, 1).Value)
Y =
Cdbl(ActiveSheet.Range("CrackFront").Cells(i, 2).Value)

```

```

SCPoints.AddLocal FirstSplinePoint + i, SplineSys,
X, Y, 0
lngSplinePoints(i - 1) = FirstSplinePoint + i
r = Sqr(X ^ 2 + Y ^ 2)
If r < dblCrackLength Then dblCrackLength = r
If i <> 1 Then
dblCrackLength = dblCrackLength + Sqr((X -
oldx) ^ 2 + (Y - oldy) ^ 2)
End If
oldx = X
oldy = Y
SCModel.Update
Else
intSplinePoints = i - 1
Exit For
End If
Next i
SCModel.Update

'Create the spline
Dim aSpline As StressCheck.SplineCurve
Set aSpline =
SCSplines.AddPoints(SCSplines.LastNumber + 1,
intSplinePoints, lngSplinePoints)
SCModel.Update

```

```

'Create the closing boundary by assumption there is no
crack for x<0 and the points are ordered
'check one end of the line

```

```

Dim FirstLine As Long
Dim lPoints(1) As Long
FirstLine = SCLines.LastNumber + 1
X = Cdbl(ActiveSheet.Range("CrackFront").Cells(1,
1).Value)
Y = Cdbl(ActiveSheet.Range("CrackFront").Cells(1,
2).Value)
Dim p1 As Long, p2 As Long
Dim ExtraLines() As Long 'dim this in 2 steps so it can
be redimmed later
ReDim ExtraLines(2)
Dim NewLines As Integer
NewLines = 0
Dim aLine As StressCheck.Line

```

```

'If x > -0.01 we need to first draw a line to the x = -0.01
axis.

```

```

If X > -0.01 Then
p1 = SCPoints.LastNumber + 1
SCPoints.AddLocal p1, SplineSys, -0.01 +
ActiveSheet.Range("Crack0").Cells(1, 1), 1 * Y, 0
'draw the line to the x = -0.01 axis
lPoints(0) = FirstSplinePoint + 1
lPoints(1) = p1
Set aLine = SCLines.AddTwoPoint(FirstLine, lPoints)
'draw the line
ExtraLines(NewLines) = aLine.Number
NewLines = NewLines + 1
Else
p1 = FirstSplinePoint + 1
End If

```

```

SCModel.Update

'Now repeat for the other end of the spline

X =
CDBl(ActiveSheet.Range("CrackFront").Cells(intSplinePoints, 1).Value)
Y =
CDBl(ActiveSheet.Range("CrackFront").Cells(intSplinePoints, 2).Value)
If X > -0.01 Then
    p2 = SCPoints.LastNumber + 1
    SCPoints.AddLocal p2, SplineSys, -0.01 +
ActiveSheet.Range("Crack0").Cells(1, 1), 1 * Y, 0
'draw the line to the x=-0.01 axis
lPoints(0) = FirstSplinePoint + intSplinePoints
lPoints(1) = p2
Set aLine =
SCLines.AddTwoPoint(SCLines.LastNumber + 1, lPoints)
'draw the line
    ExtraLines(NewLines) = aLine.Number
    NewLines = NewLines + 1
Else
    p2 = FirstSplinePoint + intSplinePoints
End If
SCModel.Update

'draw a line joining the two endpoints
lPoints(0) = p1
lPoints(1) = p2
Set aLine =
SCLines.AddTwoPoint(SCLines.LastNumber + 1, lPoints)
    ExtraLines(NewLines) = aLine.Number
    NewLines = NewLines + 1
SCModel.Update

SCModel.Update

'Expand the extralines array for use in the body imprint
command
ReDim Preserve ExtraLines(NewLines + 1)
ExtraLines(NewLines) = aSpline.Number 'add the spline
curve to the list of boundaries
ExtraLines(NewLines + 1) = ExtraLines(0)
ExtraLines(0) = CrackPlane
Dim CrackBody As StressCheck.Body

Set CrackBody =
SCModel.Bodies.AddImprint(SCModel.Bodies.LastNumber + 1, bodyImprintCurveNormal, NewLines + 2,
ExtraLines, 0.000001)
SCModel.Update

'Now trim the plane away
'First pick the plane
Dim bndrys(0) As Long
Dim obj_num As Long, entity_num As Long

X = CDBl(ActiveSheet.Range("Crack0").Cells(1,
1).Value)
Y = CDBl(ActiveSheet.Range("Crack0").Cells(1,
2).Value)

```

```

z = CDBl(ActiveSheet.Range("Crack0").Cells(1,
3).Value)
SCModel.EnableSelection opSurface
SCModel.PickObject opSurface, X, Y, z, obj_num,
entity_num, 0.05, ptCentre

' obj_num = 31 'manually choose the plane

'Now complete the trim operation
If obj_num = 0 Then
    Err.Raise vbObjectError + 2, , "Unable to pick the
plane containing the ellipse"
Else
    'obj_num = 55 '*have to pick the plane to be trimmed
    bndrys(0) = obj_num
    Set CrackBody = SCModel.Bodies.AddTrimHeal(0,
bodyCopyNoHeal, 1, bndrys)
End If
SCModel.Update

'Finally create a body union of the part and the crack
Dim tools(0) As Long
tools(0) = CrackBody.Number + 1 ' +1 to account for the
"Result" body in the AddTrimHeal method
Set CrackBody = SCModel.Bodies.AddBoolean(0,
bodyUnion, PlateBody.Number, 1, tools)
SCModel.Update

'Return the union so that if we want to put in more cracks
we can
Set PlateBody = CrackBody
SCModel.Display.SurfaceShade = False

' aComposite.Status = NoStatus

End Sub

Sub MeshModel()
'Sub MeshModel()

'This routine creates a mesh for the model. If there is
more than one crack some of this code
'i.e. the definition of the crack face and the boundary
layer will need to be moved to the make crack routine.

On Error Resume Next

'Create a set for the body
Dim meshobj(0) As Long
meshobj(0) = SCModel.LastBodyNumber
'Create body set ("BODY")
SCSets.Add "BODY", stBody, sotList, meshobj
SCModel.Update

'The following is specific for each crack and would need
to be moved to the MakeCrack routine if there
'was more than one crack

'Create a set for the crackface.

Dim objnum() As Long, entnum() As Long

```

```

Dim objtype() As ObjectType

ReDim objnum(0)
ReDim entnum(0)
ReDim objtype(0)
objtype(0) = otBoundary

Dim xyz As Variant

i = Int((UBound(IngSplinePoints) + 1) / 2) - 1 ' pick the
middle spline point
xyz =
SCPoints.Point(IngSplinePoints(i)).Location(ItGlobal)

X = CDbI(ActiveSheet.Range("Crack0").Cells(1,
1).Value)
If xyz(1) > CDbI(ActiveSheet.Range("csinky").Cells(1,
1).Value) Then
Y = CDbI(ActiveSheet.Range("csinky").Cells(1,
1).Value) / 2
Else
Y = xyz(1)
End If
z = CDbI(ActiveSheet.Range("Crack0").Cells(1,
3).Value)
SCModel.EnableSelection opSurface

On Error Resume Next
SCModel.PickObject opSurface, X, Y, z, objnum(0),
entnum(0), 0.05, ptCentre
'there is an error here

'objnum(0) = 56 ' manually select the required edge
If objnum(0) = 0 Then
'set the obj manually
objnum(0) = 56
SCSets.AddObjectSet "CRACKSURF", objtype, objnum
'Err.Raise vbObjectError + 2, , "Unable to pick the
crack surface"
Else
SCSets.AddObjectSet "CRACKSURF", objtype,
objnum
End If
SCModel.Update

'Now create a set for the crack edge
'Pick the first point and use the base point instead of the
centre
SCModel.ObjectOperation opAnyObject, atCancel
xyz =
SCPoints.Point(IngSplinePoints(i)).Location(ItGlobal)
X = xyz(0) 'gives an automation error
Y = xyz(1)
z = xyz(2)
objtype(0) = otBoundary
Dim myobject As StressCheck.ObjectPick
myobject = opCurve
SCModel.EnableSelection opCurve
'the tolerance needs to be increased to find the edge

```

```

SCModel.PickObject opPSurfCurve, X, Y, z, objnum(0),
entnum(0), 0.1, ptCentre

Err.Clear

' objnum(0) = 74
If objnum(0) = 0 Or 56 Then
'set the obj manually
objnum(0) = 75
SCSets.AddObjectSet "CRACKEDGE", objtype,
objnum
'Err.Raise vbObjectError + 2, , "Unable to pick the
crack edge"
Else
objnum(0) = 75
SCSets.AddObjectSet "CRACKEDGE", objtype,
objnum
End If

SCModel.Update
i = 0

'Now set up the automesh
Dim BodyMesh As StressCheck.Automesh
Dim CrackMesh As StressCheck.Automesh
Dim CrackEdgeMesh As StressCheck.Automesh

'The following code tries to establish the parameters for
the boundary layer mesh

Dim mrat As Double
Dim trans As Double
Dim ctrat As Double
Dim Layers As Integer
Dim t0 As Double
Dim int0 As Double
Dim tTotal As Double

mrat = ActiveSheet.Range("mrat").Value
trans = ActiveSheet.Range("trans").Value
ctrat = ActiveSheet.Range("ctrat").Value
Layers = 2
int0 = ActiveSheet.Range("int").Value
tTotal = int0 * ActiveSheet.Range("tTotalrat").Value
t0 = tTotal * ActiveSheet.Range("t0rat").Value
dblContourRadius = ActiveSheet.Range("ipr").Value *
tTotal

Set BodyMesh = SCAutomeshes.AddMeshSimGlobal(1,
"BODY", meshTetra, True, False, False, True, mrat, , ,
trans)

Set CrackMesh =
SCAutomeshes.AddMeshSimCrackFace(2,
"CRACKSURF", True)
Set CrackEdgeMesh =
SCAutomeshes.AddMeshSimBoundaryLayer(3,
"CRACKEDGE", True, True, True, True, , ctrat, Layers, t0,
tTotal, , True)

```

```

SCModel.Update

SCAutomeshes.Automesh(CrackEdgeMesh.Number).Value
(apRatio) = ctrat

SCAutomeshes.Automesh(CrackEdgeMesh.Number).Value
(apLayers) = Layers

SCAutomeshes.Automesh(CrackEdgeMesh.Number).Value
(apMinEdgeLen) = t0

SCAutomeshes.Automesh(CrackEdgeMesh.Number).Value
(apSize) = tTotal

SCAutomeshes.Automesh(CrackEdgeMesh.Number).Update

SCAutomeshes.Automesh(CrackEdgeMesh.Number).Value
(apRatio) = ctrat

SCAutomeshes.Automesh(CrackEdgeMesh.Number).Update
  SCModel.Update

  DoEvents
' call automesh
  SCModel.Automesh
  SCModel.Update
  Dim n As Long
  n = SCModel.Elements.Count
  'ActiveSheet.Range("Nelem").Value = n
  If (n < 4000) Or (n > 20000) Then
  TerminateStressCheck
  Application.Wait (Now + Time Value("00:00:20"))

  InitializeStressCheck
  SCModel.StandardUnits = unitInchLbSec
  CreatePlate
  Dim PlateBody As StressCheck.Body
  Set PlateBody =
SCModel.Bodies.Body(SCModel.Bodies.LastNumber)
  SCModel.Update

  MakeCrack PlateBody

  MeshModel
  Else
  End If

  'Convert mapping to geometric
  SCModel.ConvertElementMapping cemGeometric

  'Update model
  SCModel.Update

End Sub

```

```

'-----AssignLoads
Routine-----
Sub AssignLoads()

  'Assign loads to parts of the plate
  'if this was done through the sci file then this routine can
  be dropped
  'by commenting it out in the main routine

  'In the present case it finds the top surface and applies a
  traction load.

  Dim objnum() As Long, entnum() As Long
  Dim objtype() As ObjectType

  ReDim objnum(0)
  ReDim entnum(0)
  ReDim objtype(0)
  objtype(0) = otBoundary

  X = 0
  Y = 0
  z = Cdbl(ActiveSheet.Range("PLength").Value) / 2
  On Error Resume Next
  SCModel.EnableSelection opSurface
  SCModel.PickObject opSurface, X, Y, z, objnum(0),
  entnum(0), , ptCentre

  If objnum(0) = 0 Then
  Err.Raise vbObjectError + 2, , "Unable to pick the
  load surface"
  Else
  SCSets.Add "TOPLOAD", stBoundary, sotList,
  objnum
  End If
  SCModel.Update

  SCModel.ObjectOperation opAnyObject, atCancel

  'Now apply the load to the set
  'Define load structure
  Dim loadvals(2) As Variant, loadtypes(2) As
  LoadValueType
  loadvals(0) = ActiveSheet.Range("AppliedStress").Value
  'magnitude
  loadvals(1) = 0# 'direction (degrees)
  loadvals(2) = 0# 'z direction
  loadtypes(0) = lvtConstant
  loadtypes(1) = lvtConstant
  loadtypes(2) = lvtConstant

  SCLoads.AddBoundaryLoad "LOAD", "TOPLOAD",
  lmTraction, dtNormalTangent, atSet, 1, loadvals, loadtypes,
  "TOPLOAD"
  SCModel.Update

End Sub

'-----AssignConstraints
Routine-----
Sub AssignConstraints()

```

'Assign Constraints to parts of the plate
 'If this was done through the Sci File then this routine
 can be dropped by commenting it out in the main routine
 'In the present case it fixes the surface furthest from the
 cylindrical hole

```
Dim objnum() As Long, entnum() As Long
Dim objtype() As ObjectType

ReDim objnum(0)
ReDim entnum(0)
ReDim objtype(0)
objtype(0) = otBoundary

'Pick the back face
X = 0 ' middle of bottom face
Y = 0
z = -CDbl(ActiveSheet.Range("PLength").Value) / 2

SCModel.EnableSelection opSurface
On Error Resume Next

SCModel.PickObject opSurface, X, Y, z, objnum(0),
entnum(0), , ptCentre

If objnum(0) = 0 Then
  Err.Raise vbObjectError + 2, , "Unable to pick the
  constraint surface"
Else
  SCSets.Add "BOTFIX", stBoundary, sotList, objnum
End If

SCModel.Update
'Now apply the constraint
Dim ConsVals(2) As Variant, ConsTypes(2) As
ConstraintValueType
ConsVals(0) = 0# 'x displacement
ConsVals(1) = 0# 'y displacement
ConsVals(2) = 0# 'z displacement
ConsTypes(0) = cvtConstant
ConsTypes(1) = cvtConstant
ConsTypes(2) = cvtConstant

SCCons.AddHomogeneousBoundaryConstraint "FIX",
"BOTFIX", hcmBuiltIn
SCModel.Update
```

End Sub

```
'-----DefineMaterials
Routine-----
Sub DefineMaterials()
```

'Defines materials. If this is done in the sci file then this
 routine can be dropped

```
With SCModel
  '-----Define Materials-----
```

```
Dim MatCoef(23) As Variant
Set SCMaterials = .Materials
Dim SCMaterialDef As StressCheck.MATERIALDEF
Dim SCMaterial As StressCheck.Material
```

'Define Material Properties for the Aluminum 2014-T6
 (from StressCheck data)

```
For i = 1 To 23
  MatCoef(i - 1) = 0#
Next i
MatCoef(0) = ActiveSheet.Range("Youngs").Value
MatCoef(3) = ActiveSheet.Range("Poissons").Value
With SCMaterialDef
  .Name = "MAT"
  .Type = matIsotropic
  .Description = "Dr.Wowk's Magic Material"
  .Case = ptPlaneStrain
  .Data = MatCoef
End With
Set SCMaterial =
SCMaterials.AddRecord(SCMaterialDef)
```

'repeat for other materials

```
.Update
End With
```

End Sub

```
'-----AssignMaterials
Routine-----
Sub AssignMaterials()
```

'Assigns the material created above to the plate body (the
 Body set established in the MeshIt routine)
 'Can be commented out if this is done in the sci file

```
Dim SCMaterialAssignment As
StressCheck.MaterialAssignment
With SCModel
  Set SCMaterialAssignment =
  SCMaterialAssignments.Add("MAT", "BODY", atAll, 0,
  matHomogeneous, "Steel", , True)
.Update
End With
```

End Sub

```
'-----PrepSolution Routine--
-----
Sub PrepSolution()
```

'The equivalent of the Solution tab in the preprocessor
 'Note that the constraints MUST have a name of "CONS"
 and the loads MUST have a name of "LOADS"

```
SCSolIDs.Add "SOL", "FIX", "LOAD", ssActive
```

End Sub

'-----ExtractK Routine-----

```

Sub ExtractK()

'Recover the stress concentration data
Dim efcn(1) As StressCheck.ElasticityFunctions
Dim ExNum As Long
Dim ef As Object
Dim ndp As Integer
Dim expts As Integer
SCDoc.Info.Parameters.Add "_crack_360", "", 0
'ndp gives the # of data points along the interior of the
crack edge
'the total number of data points is ndp + 2 for the two end
points

expts = 0
ndp = ActiveSheet.Range("ExtractNum").Value - 2
efcn(0) = efK1
efcn(1) = efK1 Angle

Dim PK As Extraction
Dim PointPlot As StressCheck.POINTEXTRACTION
k = 1

For j = 1 To 3

'The if statements provide more extraction points near the
material surface

If j = 1 Then
m = 0
n = 0.1
ExNum = Int(ActiveSheet.Range("int").Value *
ActiveSheet.Range("ExRatio").Value + 13)
crackpts(0) = ExNum + 2
Else
If j = 2 Then
m = 0.1
n = 0.9
ExNum = Int(ActiveSheet.Range("int").Value *
ActiveSheet.Range("ExRatio").Value * 2.5 + 13)
crackpts(1) = ExNum + 2
Else
If j = 3 Then
m = 0.9
n = 1
ExNum = Int(ActiveSheet.Range("int").Value *
ActiveSheet.Range("ExRatio").Value + 13)
crackpts(2) = ExNum + 2
Else
End If
End If
End If

PointPlot.ContourRadius = dblContourRadius '
calculated in MeshModel
PointPlot.setname = "CRACKEDGE"
PointPlot.EFunctionsArray = efcn

```

```

PointPlot.Name = "SIFData"

PointPlot.OBJECTCOORD =
GetSetInfo("CRACKEDGE")
PointPlot.NumObjectCoords =
GetSetNumObj("CRACKEDGE")
PointPlot.Object =
StressCheck.PointsObject.poBoundary
PointPlot.SolutionID = "SOL" 'Solution name
PointPlot.RunMin = 1 'Run min
PointPlot.RunMax = 1 'Run max
PointPlot.P1Min = m
PointPlot.P2Min = 0
PointPlot.P3Min = 0
PointPlot.P1Max = n
PointPlot.P2Max = 1
PointPlot.P3Max = 1
PointPlot.NumberOfPoints = ExNum
PointPlot.NumberOfPoints2 = ExNum
PointPlot.NumberOfPoints3 = ExNum

Set PK =
SCDoc.Extractions.AddPointExtraction(PointPlot)
PK.ElasticityFunction = efK1

PK.DisplayPoints = True
PK.DoChart = True
PK.DoTable = True
PK.Points_NumberOfPoints = ExNum
PK.Points_NumberOfPoints2 = ExNum
PK.Points_NumberOfPoints3 = ExNum

PK.Update

'Now obtain the data and writing to excel
On Error Resume Next
Dim dt As StressCheck.DataTable
Set dt = SCDoc.Extractions.ExtractData("SIFData")
For i = k To (k - 1 + dt.Rows.Count)
ActiveSheet.Range("NewCrackData").Cells(i - expts,
1).Value = dt.Data(i - k, 1)
ActiveSheet.Range("NewCrackData").Cells(i - expts,
2).Value = dt.Data(i - k, 2)
ActiveSheet.Range("NewCrackData").Cells(i - expts,
3).Value = dt.Data(i - k, 3)
ActiveSheet.Range("NewCrackData").Cells(i - expts,
4).Value = dt.Data(i - k, 5)

'Overwrite extractions that dont record
If (ActiveSheet.Range("NewCrackData").Cells(i -
expts, 1) = 0 And
ActiveSheet.Range("NewCrackData").Cells(i - expts, 2) =
0) Then
expts = expts + 1

If expts > 1 Then
'MsgBox "StressCheck has not extracted properly.
There are missing extraction points."
ActiveSheet.Range("MissXpts") =
ActiveSheet.Range("MissXpts") + expts
Else
End If

```

```

Else
End If

Next
'Make sure crackpts account for expts
If j = 1 Then
crackpts(0) = ExNum + 2 - expts
Else
If j = 2 Then
crackpts(1) = ExNum + 2 - expts
Else
If j = 3 Then
crackpts(2) = ExNum + 2 - expts
Else
End If
End If
End If

k = i - 1 - expts
'Reset if extraction didn't work
If expts > 15 Then
TerminateStressCheck
InitializeStressCheck
SCModel.StandardUnits = unitInchLbSec
CreatePlate
Dim PlateBody As StressCheck.Body
Set PlateBody =
SCModel.Bodies.Body(SCModel.Bodies.LastNumber)
SCModel.Update

MakeCrack PlateBody

MeshModel
AssignLoads
AssignConstraints

'Assign Materials
DefineMaterials
AssignMaterials

'Prepare the solution
PrepSolution
'iteration = CStr(iteration)
'SCDoc.Model.Write path & iteration & ".sci"
Dim plevel As Integer
plevel = ActiveSheet.Range("plevel").Value
SolveLinear (plevel)

'Extract the stress intensity factors (K1)

ExtractK

Else
End If

expts = 0
SCDoc.Extractions.Clear

Next

'Excluding points if ipr falls out of the material
For i = 1 To k - 1
With ActiveSheet
.Range("Point").Cells(i, 1).Value = i
.Range("NewCrackData").Cells(i, 5).Value =
(Sin(Atn(Abs((.Range("NewCrackData").Cells(i + 1,
1).Value - .Range("NewCrackData").Cells(i, 1).Value) /
(.Range("NewCrackData").Cells(i + 1, 2).Value -
.Range("NewCrackData").Cells(i, 2).Value)))) *
.Range("IPRtot").Value)
If .Range("NewCrackData").Cells(i, 5).Value >=
(.Range("NewCrackData").Cells(i, 2).Value) Then
.Range("NewCrackData").Cells(i, 6) =
"EXCLUDE"
Else
.Range("NewCrackData").Cells(i, 6) = "KEEP"
End If

' Excluding results within 2.5 degrees of the crack
edge
m = (180 / 3.14159265358979) *
(Atn(.Range("newCrackData").Cells(i, 2).Value /
(.Range("newCrackData").Cells(i, 1).Value + 0.00000001 -
.Range("Crack0"))))
If m < 0 Then
.Range("NewCrackData").Cells(i, 6) =
"EXCLUDE"
Else
End If
End With
Next

ActiveSheet.Range("NewCrackData").Cells(1, 6) =
"EXCLUDE"

For i = 2 To k
,
' If ActiveSheet.Range("NewCrackData").Cells(i, 1) = 0
And ActiveSheet.Range("NewCrackData").Cells(i, 2) = 0
Then
' j = j + 1
' Else
' End If

With ActiveSheet
.Range("Point").Cells(i, 1).Value = i
' Err = 0
.Range("NewCrackData").Cells(i, 5).Value =
(Cos(Atn(Abs((.Range("NewCrackData").Cells(i - 1,
1).Value - .Range("NewCrackData").Cells(i, 1).Value) /
(.Range("NewCrackData").Cells(i - 1, 2).Value -
.Range("NewCrackData").Cells(i, 2).Value)))) *
.Range("IPRtot").Value)
If .Range("NewCrackData").Cells(i, 5).Value >=
(.Range("NewCrackData").Cells(i, 1).Value) -
(.Range("Crack0").Cells(1, 1).Value) Then
.Range("NewCrackData").Cells(i, 6) =
"EXCLUDE"
Else
.Range("NewCrackData").Cells(i, 6) = "KEEP"
End If

```



```

' Excluding results within 2.5 degrees of the crack edge
m = (180 / 3.14159265358979) *
(Atn(.Range("newCrackData").Cells(i, 2).Value /
.Range("newCrackData").Cells(i, 1).Value +
0.000000000000001 - .Range("Crack0")))
If m > 90 Then
.Range("NewCrackData").Cells(i, 6) =
"EXCLUDE"
Else
End If
End With

```

Next

```

ActiveSheet.Range("NewCrackData").Cells(k, 6) =
"EXCLUDE"

```

```

For i = 1 To k
With ActiveSheet
If .Range("NewCrackData").Cells(i, 6) = "KEEP"
Then
.Range("NewCrackData").Cells(i, 7).Value =
.Range("NewCrackData").Cells(i, 4).Value * (1 -
.Range("StressRatio").Value)
Else
.Range("NewCrackData").Cells(i,
7).ClearContents
End If
End With
Next

```

```

ActiveSheet.Range("ExtractNum") = i - 1

```

End Sub

```

'-----SolveLinear Routine---

```

```

Sub SolveLinear(plevel As Integer)

```

```

'Performs the solution

```

```

Set SCLinearSols = SCDoc.Solutions
SCLinearSols.Solution("1-Linear").plevel = plevel
SCLinearSols.Solution("1-Linear").PLimit = plevel
SCLinearSols.Solution("1-Linear").Update

```

```

SCDoc.xSolve ("1-Linear")

```

End Sub

```

'-----InitialAnalysis

```

```

Routine-----

```

```

Sub InitialAnalysis()

```

```

'Check if there are already charts on the worksheet, if so
delete them

```

```

If ActiveSheet.ChartObjects.Count <> 0 Then
ActiveSheet.ChartObjects.Delete
End If

```

```

'First create the plot of the K1 values as a function of
extraction point number

```

```

Range("S6").Select
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter

```

```

With ActiveChart

```

```

.SeriesCollection.NewSeries
.SeriesCollection(1).XValues =
ActiveSheet.Range("$H$25:$H$266")
.SeriesCollection(1).Values =
ActiveSheet.Range("$L$25:$L$266")
.Axes(xlCategory).HasTitle = True
.Axes(xlCategory).AxisTitle.Caption = "Extraction
Point Number"
.Axes(xlValue).HasTitle = True
.Axes(xlValue).AxisTitle.Caption = "Stress Intensity
Factor (psi(in)^0.5)"
.Axes(xlValue).MinimumScale = 400
ActiveSheet.Shapes(2).Left = ActiveSheet.Cells(22,
29).Left
ActiveSheet.Shapes(2).Top = ActiveSheet.Cells(22,
29).Top
ActiveSheet.Shapes(2).Height =
ActiveSheet.Range("AJ22:AJ35").Height
ActiveSheet.Shapes(2).Width =
ActiveSheet.Range("AC35:AJ35").Width
End With

```

```

Dim rowCount As Double

```

```

Dim rowIndex As Double

```

```

Dim xL() As Double

```

```

Dim xVal() As Double

```

```

Dim K1L() As Double

```

```

Dim Eq() As Variant

```

```

ReDim Eq(2)

```

```

With ActiveSheet

```

```

rowCount = Range(.Range("N25"),
.Range("N65535")).End(xlUp)).Count

```

```

'Starting at point 1 determine how many Excludes there
are so that they can be linearly interpolated

```

```

'* if there are three keeps in a row everything previous
must be interpolated

```

```

j = 0

```

```

For i = 1 To rowCount - 1

```

```

If .Range("NewCrackData").Cells(i, 6) = "KEEP"

```

Then

```

j = j + 1

```

```

Else

```

```

j = 0

```

```

End If

```

```

If j = 3 Then

```

```

rowIndex = i - 3

```

```

Exit For

```

```

End If

```

Next

```

'* Average the values by adding rowIndex=8

```

```

'*rowIndex = 4

```

```

ReDim xL(3, 0)
ReDim K1L(3, 0)
ReDim xVal(3, 0)

For i = 1 To 4
    xL(i - 1, 0) = .Range("Point").Cells(rowIndex + i,
3).Value
    'xL(i - 1, 1) = (.Range("Point").Cells(rowIndex + i,
1).Value) ^ 2
    K1L(i - 1, 0) =
.Range("NewCrackData").Cells(rowIndex + i, 7).Value
    xVal(i - 1, 0) = xL(i - 1, 0)
Next

End With
'Next do a linear fit for the first few points to account for
the error in the K1 value at the surface
Range("S6").Select
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter

With ActiveChart
    .SeriesCollection.NewSeries
    .SeriesCollection(1).XValues = xL
    .SeriesCollection(1).Values = K1L
    .SeriesCollection(1).Select
    .SeriesCollection(1).Trendlines.Add
    ActiveSheet.ChartObjects(2).Activate
    .SeriesCollection(1).Trendlines(1).Select
    Selection.DisplayEquation = True
    .SeriesCollection(1).Trendlines(1).DataLabel.Select
    Selection.Left = 120
    Selection.Top = 120
    ActiveSheet.Shapes(3).Left = ActiveSheet.Cells(37,
29).Left
    ActiveSheet.Shapes(3).Top = ActiveSheet.Cells(37,
29).Top
    ActiveSheet.Shapes(3).Height =
ActiveSheet.Range("AJ37:AJ50").Height
    ActiveSheet.Shapes(3).Width =
ActiveSheet.Range("AC50:AJ50").Width
End With

Eq() = Application.WorksheetFunction.LinEst(K1L, xL)
ActiveSheet.Range("Slope1").Value = Eq(1)
ActiveSheet.Range("Intercept1").Value = Eq(2)
'ActiveSheet.Range("Constant1").Value = Eq(3)
For i = 1 To rowIndex *- 1
    ' ActiveSheet.Range("NewCrackData").Cells(i,
7).Value = (Eq(1) * ActiveSheet.Range("Point").Cells(i,
3).Value + Eq(2))
Next

'Create a plot of the linear extrapolation of the K1 value
at the end of the crack.
With ActiveSheet
    rowCount = Range(.Range("N25"),
Range("N65535").End(xlUp)).Count

```

```

'Starting at point 1 determine how many Excludes there
are so that they can be linearly interpolated
'* if there are three keeps in a row everything previous
must be interpolated
j = 0

For i = rowCount / 2 To rowCount
    If .Range("NewCrackData").Cells(i, 6) =
"EXCLUDE" Then
        rowIndex = i
        Exit For
    Else
        End If
Next

' rowIndex = rowCount - 3

ReDim xL(3, 0)
ReDim K1L(3, 0)
ReDim xVal(3, 0)

For i = 1 To 4
    xL(i - 1, 0) = .Range("Point").Cells(rowIndex - i - 0,
2).Value
    'xL(i - 1, 1) = (.Range("Point").Cells(rowIndex - i - 0,
1).Value) ^ 2
    K1L(i - 1, 0) =
.Range("NewCrackData").Cells(rowIndex - i - 0, 7).Value
    xVal(i - 1, 0) = xL(i - 1, 0)
Next

'For i = 1 To 2
'    xL(i - 1, 0) = .Range("Point").Cells(rowIndex - i,
1).Value
'    ' xL(i - 1, 1) = (.Range("Point").Cells(rowIndex - i,
1).Value) ^ 2
'    K1L(i - 1, 0) =
.Range("NewCrackData").Cells(rowIndex - i, 7).Value
'    xVal(i - 1, 0) = xL(i, 0)
'Next

End With
'Next do a linear fit for the first few points to account
for the error in the K1 value at the surface
Range("S6").Select
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter

With ActiveChart
    .SeriesCollection.NewSeries
    .SeriesCollection(1).XValues = xL
    .SeriesCollection(1).Values = K1L
    .SeriesCollection(1).Select
    .SeriesCollection(1).Trendlines.Add
    ActiveSheet.ChartObjects(2).Activate
    .SeriesCollection(1).Trendlines(1).Select
    Selection.DisplayEquation = True
    .SeriesCollection(1).Trendlines(1).DataLabel.Select
    Selection.Left = 40
    Selection.Top = 80

```

```

ActiveSheet.Shapes(7).Left = ActiveSheet.Cells(236,
29).Left
ActiveSheet.Shapes(7).Top = ActiveSheet.Cells(236,
29).Top
ActiveSheet.Shapes(7).Height =
ActiveSheet.Range("AJ236:AJ249").Height
ActiveSheet.Shapes(7).Width =
ActiveSheet.Range("AC249:AJ249").Width
End With

```

```

Eq() = Application.WorksheetFunction.LinEst(K1L, xL)
ActiveSheet.Range("Slope2").Value = Eq(1)
ActiveSheet.Range("Intercept2").Value = Eq(2)
' ActiveSheet.Range("Constant2").Value = Eq(3)

```

```

Dim ExtractNum As Double
ExtractNum = ActiveSheet.Range("ExtractNum").Value

```

```

For i = rowCount To rowIndex Step -1
'ActiveSheet.Range("NewCrackData").Cells(i,
7).Value = (Eq(1) * (ActiveSheet.Range("Point").Cells(i,
2).Value) + Eq(2))
Next
End Sub

```

```

Sub CrackAnalysis()
ActiveSheet.Calculate
s = ActiveSheet.Range("CrackType").Cells(1, 1)

```

```

Dim rowCount As Double
Dim rowIndex As Double
rowCount = Range(ActiveSheet.Range("I25"),
ActiveSheet.Range("I65535").End(xlUp)).Count
'shifting the x values
For i = 1 To rowCount
With ActiveSheet
.Range("NewCrackData").Cells(i, 8).Value =
.Range("NewCrackData").Cells(i, 1).Value -
.Range("Crack0").Cells(1, 1).Value
.Range("NewCrackData").Cells(i, 9).Value =
.Range("NewCrackData").Cells(i, 2).Value -
.Range("Crack0").Cells(1, 2).Value
.Range("NewCrackData").Cells(i, 10).Value =
.Range("NewCrackData").Cells(i, 3).Value -
.Range("Crack0").Cells(1, 3).Value
End With
Next

```

'Calculating da/dN, slope and new x and y values using the growth rules

```

With ActiveSheet
'Growth for range of K values for the first point

```

```

If (.Range("NewCrackData").Cells(1, 7).Value >
.Range("ParisValues").Cells(2, 1)) And
(.Range("NewCrackData").Cells(1, 7).Value <
.Range("ParisValues").Cells(2, 3)) Then
.Range("NewCrackData").Cells(1, 11).Formula =
"=(Offset(NewCrackData, " & 1 - 1 & ", 6) / (maxK)) ^
(Paris1) * (Cinc)"

```

```

Else
If (.Range("NewCrackData").Cells(1, 7).Value >
.Range("ParisValues").Cells(3, 1)) And
(.Range("NewCrackData").Cells(1, 7).Value <
.Range("ParisValues").Cells(3, 3)) Then
.Range("NewCrackData").Cells(1, 11).Formula =
"=(Offset(NewCrackData, " & 1 - 1 & ", 6) / (maxK)) ^
(Paris2) * (Cinc)"
Else
If (.Range("NewCrackData").Cells(1, 7).Value >
.Range("ParisValues").Cells(4, 1)) And
(.Range("NewCrackData").Cells(1, 7).Value <
.Range("ParisValues").Cells(4, 3)) Then
.Range("NewCrackData").Cells(1,
11).Formula = "=(Offset(NewCrackData, " & 1 - 1 & ", 6) /
(maxK)) ^ (Paris3) * (Cinc)"
Else
End If
End If
End If

```

```

.Range("NewCrackData").Cells(1, 12).Value = -
.Range("NewCrackData").Cells(1, 8).Value -
.Range("NewCrackData").Cells(1 + 1, 8).Value) /
(.Range("NewCrackData").Cells(1, 9).Value -
.Range("NewCrackData").Cells(1 + 1, 9).Value)

```

```

'Growth for range of K values for the last point
If (.Range("NewCrackData").Cells(rowCount,
7).Value > .Range("ParisValues").Cells(2, 1)) And
(.Range("NewCrackData").Cells(rowCount, 7).Value <
.Range("ParisValues").Cells(2, 3)) Then
.Range("NewCrackData").Cells(rowCount,
11).Formula = "=(Offset(NewCrackData, " & rowCount - 1
& ", 6) / (maxK)) ^ (Paris1) * (Cinc)"
Else
If (.Range("NewCrackData").Cells(rowCount,
7).Value > .Range("ParisValues").Cells(3, 1)) And
(.Range("NewCrackData").Cells(rowCount, 7).Value <
.Range("ParisValues").Cells(3, 3)) Then
.Range("NewCrackData").Cells(rowCount,
11).Formula = "=(Offset(NewCrackData, " & rowCount - 1
& ", 6) / (maxK)) ^ (Paris2) * (Cinc)"
Else
If (.Range("NewCrackData").Cells(rowCount,
7).Value > .Range("ParisValues").Cells(4, 1)) And
(.Range("NewCrackData").Cells(rowCount, 7).Value <
.Range("ParisValues").Cells(4, 3)) Then
.Range("NewCrackData").Cells(rowCount,
11).Formula = "=(Offset(NewCrackData, " & rowCount - 1
& ", 6) / (maxK)) ^ (Paris3) * (Cinc)"
Else
End If
End If
End If
' .Range("NewCrackData").Cells(rowCount,
11).Formula = "=(Offset(NewCrackData, " & rowCount - 1
& ", 6) / (maxK)) ^ (ParisN) * (Cinc)"

```

```

.Range("NewCrackData").Cells(rowCount,
12).Value = -(Range("NewCrackData").Cells(rowCount -
1, 8).Value - .Range("NewCrackData").Cells(rowCount,
8).Value) / (.Range("NewCrackData").Cells(rowCount - 1,
9).Value - .Range("NewCrackData").Cells(rowCount,
9).Value)

```

```

For i = 1 To rowCount

```

```

    If (i <> 1) And (i <> rowCount) Then

```

```

        'Growth for range of K values for all but first
and last point

```

```

        If (.Range("NewCrackData").Cells(i, 7).Value >
.Range("ParisValues").Cells(2, 1)) And
(.Range("NewCrackData").Cells(i, 7).Value <
.Range("ParisValues").Cells(2, 3)) Then
            .Range("NewCrackData").Cells(i, 11).Formula =
"=(Offset(NewCrackData, " & i - 1 & ", 6) / (maxK)) ^
(Paris1) * (Cinc)"

```

```

        Else

```

```

            If (.Range("NewCrackData").Cells(i, 7).Value >
.Range("ParisValues").Cells(3, 1)) And
(.Range("NewCrackData").Cells(i, 7).Value <
.Range("ParisValues").Cells(3, 3)) Then

```

```

                .Range("NewCrackData").Cells(i, 11).Formula =
"=(Offset(NewCrackData, " & i - 1 & ", 6) / (maxK)) ^
(Paris2) * (Cinc)"

```

```

            Else

```

```

                If (.Range("NewCrackData").Cells(i, 7).Value
> .Range("ParisValues").Cells(4, 1)) And
(.Range("NewCrackData").Cells(i, 7).Value <
.Range("ParisValues").Cells(4, 3)) Then

```

```

                    .Range("NewCrackData").Cells(i, 11).Formula =
"=(Offset(NewCrackData, " & i - 1 & ", 6) / (maxK)) ^
(Paris3) * (Cinc)"

```

```

                Else

```

```

                    End If

```

```

            End If

```

```

        End If

```

```

        ' .Range("NewCrackData").Cells(i, 11).Formula =
"=(Offset(NewCrackData, " & i - 1 & ", 6) / (maxK)) ^
(ParisN) * (Cinc)"

```

```

        .Range("NewCrackData").Cells(i, 12).Value = -
(.Range("NewCrackData").Cells(i - 1, 8).Value -
.Range("NewCrackData").Cells(i + 1, 8).Value) /
(.Range("NewCrackData").Cells(i - 1, 9).Value -
.Range("NewCrackData").Cells(i + 1, 9).Value)

```

```

        Else

```

```

        End If

```

```

    If .Range("NewCrackData").Cells(i, 12).Value < 0
Then

```

```

        ' Following if statement covers slope issue for the
first point

```

```

        If i = 1 Then

```

```

            j = 2

```

```

        Else

```

```

            j = i

```

```

        End If

```

```

        If (.Range("NewCrackData").Cells(j - 1, 8).Value
- .Range("NewCrackData").Cells(i + 1, 8).Value) > 0 Then

```

```

            .Range("NewCrackData").Cells(i, 13).Value = -1
* (1 / (Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

            .Range("NewCrackData").Cells(i, 14).Value = -1
* ((.Range("newCrackData").Cells(i, 12)) /
(Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

            .Range("newXY").Cells(i, 1).Value =
.Range("NewCrackData").Cells(i, 8).Value +
(.Range("NewCrackData").Cells(i, 13).Value *
.Range("NewCrackData").Cells(i, 11).Value)

```

```

            .Range("newXY").Cells(i, 2).Value =
.Range("NewCrackData").Cells(i, 9).Value +
(.Range("NewCrackData").Cells(i, 14).Value *
.Range("NewCrackData").Cells(i, 11).Value)

```

```

        Else

```

```

            .Range("NewCrackData").Cells(i, 13).Value = (1
/ (Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

            .Range("NewCrackData").Cells(i, 14).Value =
((.Range("newCrackData").Cells(i, 12)) /
(Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

            .Range("newXY").Cells(i, 1).Value =
.Range("NewCrackData").Cells(i, 8).Value +
(.Range("NewCrackData").Cells(i, 13).Value *
.Range("NewCrackData").Cells(i, 11).Value)

```

```

            .Range("newXY").Cells(i, 2).Value =
.Range("NewCrackData").Cells(i, 9).Value +
(.Range("NewCrackData").Cells(i, 14).Value *
.Range("NewCrackData").Cells(i, 11).Value)

```

```

        End If

```

```

        Else

```

```

            'For a type 2 crack at the first couple points growng
down and out

```

```

            If (s = 2) And (i < rowCount / 4) Then

```

```

                .Range("NewCrackData").Cells(i, 13).Value = -1
* (1 / (Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

                .Range("NewCrackData").Cells(i, 14).Value = -1
* ((.Range("newCrackData").Cells(i, 12)) /
(Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

                .Range("newXY").Cells(i, 1).Value =
.Range("NewCrackData").Cells(i, 8).Value +
(.Range("NewCrackData").Cells(i, 13).Value *
.Range("NewCrackData").Cells(i, 11).Value)

```

```

                .Range("newXY").Cells(i, 2).Value =
.Range("NewCrackData").Cells(i, 9).Value +
(.Range("NewCrackData").Cells(i, 14).Value *
.Range("NewCrackData").Cells(i, 11).Value)

```

```

            Else

```

```

                .Range("NewCrackData").Cells(i, 13).Value = (1
/ (Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

                .Range("NewCrackData").Cells(i, 14).Value =
((.Range("newCrackData").Cells(i, 12)) /
(Sqr(.Range("newCrackData").Cells(i, 12) ^ 2 + 1)))

```

```

                .Range("newXY").Cells(i, 1).Value =
.Range("NewCrackData").Cells(i, 8).Value +

```

```

.Range("NewCrackData").Cells(i, 13).Value *
.Range("NewCrackData").Cells(i, 11).Value)
    .Range("newXY").Cells(i, 2).Value =
.Range("NewCrackData").Cells(i, 9).Value +
(.Range("NewCrackData").Cells(i, 14).Value *
.Range("NewCrackData").Cells(i, 11).Value)
    End If
    End If
    Next
    End With

'define crack type

    Dim xVal() As Double
    Dim yL() As Double
    Dim xL() As Double
    Dim Eq() As Variant
    ReDim Eq(3)
    '* making the first new xy point at the edge (different than
at the bore because x and y points are reversed in the plot)
    ReDim yL(3, 0)
    ReDim xVal(3, 0)
    ReDim xL(3, 1)

    With ActiveSheet

        For i = 1 To rowCount - 1
            If .Range("NewCrackData").Cells(i, 6) = "KEEP"
Then
                j = j + 1
            Else
                j = 0
            End If

            If j = 3 Then
                rowIndex = i - 3
                Exit For
            End If
        Next

'test to see what type of crack location
If s = 1 Or s = 3 Then
    For i = 1 To 4
        xL(i - 1, 0) = .Range("newXY").Cells(rowIndex + i,
2).Value
        xL(i - 1, 1) = (.Range("newXY").Cells(rowIndex + i,
2).Value) ^ 2
        xVal(i - 1, 0) = .Range("newXY").Cells(rowIndex + i,
2).Value
        yL(i - 1, 0) = .Range("newXY").Cells(rowIndex + i,
1).Value
    Next

Else
    If s = 2 Then
        For i = 1 To 4
            xL(i - 1, 0) = .Range("newXY").Cells(rowIndex + i,
1).Value
            xL(i - 1, 1) = (.Range("newXY").Cells(rowIndex + i,
1).Value) ^ 2
            xVal(i - 1, 0) = .Range("newXY").Cells(rowIndex + i,
1).Value

```

```

        yL(i - 1, 0) = .Range("newXY").Cells(rowIndex + i,
2).Value
        Next
    Else
    End If
End If
End With

Range("S6").Select
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter

' With ActiveChart
' .SeriesCollection.NewSeries
' .SeriesCollection(1).XValues = xVal
' .SeriesCollection(1).Values = yL
' .SeriesCollection(1).Select
' .SeriesCollection(1).Trendlines.Add xlPolynomial, 2
' ActiveSheet.ChartObjects(2).Activate
' .SeriesCollection(1).Trendlines(1).Select
' Selection.DisplayEquation = True
' .SeriesCollection(1).Trendlines(1).DataLabel.Select
' Selection.Left = 120
' Selection.Top = 120
' ActiveSheet.Shapes(9).Left = ActiveSheet.Cells(37,
38).Left
' ActiveSheet.Shapes(9).Top = ActiveSheet.Cells(37,
38).Top
' ActiveSheet.Shapes(9).Height =
ActiveSheet.Range("AS37:AS50").Height
' ActiveSheet.Shapes(9).Width =
ActiveSheet.Range("AL50:AS50").Width
' End With

With ActiveChart
    .SeriesCollection.NewSeries
    .SeriesCollection(1).XValues = xVal
    .SeriesCollection(1).Values = yL
    .SeriesCollection(1).Select
    .SeriesCollection(1).Trendlines.Add xlLinear
    ActiveSheet.ChartObjects(2).Activate
    .SeriesCollection(1).Trendlines(1).Select
    Selection.DisplayEquation = True
    .SeriesCollection(1).Trendlines(1).DataLabel.Select
    Selection.Left = 120
    Selection.Top = 120
    ActiveSheet.Shapes(8).Left = ActiveSheet.Cells(54,
38).Left
    ActiveSheet.Shapes(8).Top = ActiveSheet.Cells(54,
38).Top
    ActiveSheet.Shapes(8).Height =
ActiveSheet.Range("AS54:AS67").Height
    ActiveSheet.Shapes(8).Width =
ActiveSheet.Range("AL67:AS67").Width
End With
Eq() = Application.WorksheetFunction.LinEst(yL, xVal)
ActiveSheet.Range("Linear3").Value = Eq(1)
ActiveSheet.Range("Constant3").Value = Eq(2)

```

```

Eq() = Application.WorksheetFunction.LinEst(yL,
xVal)
ActiveSheet.Range("Linear3").Value = Eq(1)
ActiveSheet.Range("Constant3").Value = Eq(2)

If s = 1 Or s = 3 Then
    If ActiveSheet.Range("newXY").Cells(4, 2).Value <= 0
    Then
        ActiveSheet.Range("newXY").Cells(4, 2).Value = 0
        ActiveSheet.Range("newXY").Cells(4, 1).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(4, 2).Value
+ (Eq(2))
        ActiveSheet.Range("newXY").Cells(3, 2).Value =
ActiveSheet.Range("newXY").Cells(4, 2).Value
        ActiveSheet.Range("newXY").Cells(3, 1).Value =
ActiveSheet.Range("newXY").Cells(4, 1).Value
        End If

        If ActiveSheet.Range("newXY").Cells(3, 2).Value <= 0
        Then
            ActiveSheet.Range("newXY").Cells(3, 2).Value = 0
            ActiveSheet.Range("newXY").Cells(3, 1).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(3, 2).Value
+ (Eq(2))
            ActiveSheet.Range("newXY").Cells(2, 2).Value =
ActiveSheet.Range("newXY").Cells(3, 2).Value
            ActiveSheet.Range("newXY").Cells(2, 1).Value =
ActiveSheet.Range("newXY").Cells(3, 1).Value
            End If

            If ActiveSheet.Range("newXY").Cells(2, 2).Value <= 0
            Then
                ActiveSheet.Range("newXY").Cells(2, 2).Value = 0
                ActiveSheet.Range("newXY").Cells(2, 1).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(2, 2).Value
+ (Eq(2))
                ActiveSheet.Range("newXY").Cells(1, 2).Value =
ActiveSheet.Range("newXY").Cells(2, 2).Value
                ActiveSheet.Range("newXY").Cells(1, 1).Value =
ActiveSheet.Range("newXY").Cells(2, 1).Value
                Else
                    ActiveSheet.Range("newXY").Cells(1, 2).Value = 0
                    ActiveSheet.Range("newXY").Cells(1, 1).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(1, 2).Value
+ (Eq(2))
                End If

                For i = 1 To rowIndex
                    ActiveSheet.Range("newXY").Cells(i, 1).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(i, 2).Value +
(Eq(2))
                Next
            Else
                If s = 2 Then
                    If ActiveSheet.Range("newXY").Cells(4, 1).Value <= 0
                    Then
                        ActiveSheet.Range("newXY").Cells(4, 1).Value = 0
                        ActiveSheet.Range("newXY").Cells(4, 2).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(4, 1).Value
+ (Eq(2))

```

```

ActiveSheet.Range("newXY").Cells(3, 2).Value =
ActiveSheet.Range("newXY").Cells(4, 2).Value
        ActiveSheet.Range("newXY").Cells(3, 1).Value =
ActiveSheet.Range("newXY").Cells(4, 1).Value
        End If

        If ActiveSheet.Range("newXY").Cells(3, 1).Value <= 0
        Then
            ActiveSheet.Range("newXY").Cells(3, 1).Value = 0
            ActiveSheet.Range("newXY").Cells(3, 2).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(3, 1).Value
+ (Eq(2))
            ActiveSheet.Range("newXY").Cells(2, 2).Value =
ActiveSheet.Range("newXY").Cells(3, 2).Value
            ActiveSheet.Range("newXY").Cells(2, 1).Value =
ActiveSheet.Range("newXY").Cells(3, 1).Value
            End If

            If ActiveSheet.Range("newXY").Cells(2, 1).Value <= 0
            Then
                ActiveSheet.Range("newXY").Cells(2, 1).Value = 0
                ActiveSheet.Range("newXY").Cells(2, 2).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(2, 1).Value
+ (Eq(2))
                ActiveSheet.Range("newXY").Cells(1, 2).Value =
ActiveSheet.Range("newXY").Cells(2, 2).Value
                ActiveSheet.Range("newXY").Cells(1, 1).Value =
ActiveSheet.Range("newXY").Cells(2, 1).Value
                Else
                    ActiveSheet.Range("newXY").Cells(1, 1).Value = 0
                    ActiveSheet.Range("newXY").Cells(1, 2).Value =
(Eq(1)) * ActiveSheet.Range("newXY").Cells(1, 1).Value
+ (Eq(2))
                End If

                For i = 1 To rowIndex
                    ActiveSheet.Range("newXY").Cells(i, 2).Value = Eq(1)
* ActiveSheet.Range("newXY").Cells(i, 1).Value + Eq(2)
                Next

                Else
                    End If
                End If

                'Verification
                ReDim yL(3, 0)
                ReDim xVal(3, 0)

                For i = 1 To 4
                    xVal(i - 1, 0) = ActiveSheet.Range("newXY").Cells(i,
1).Value
                    yL(i - 1, 0) = ActiveSheet.Range("newXY").Cells(i,
2).Value
                Next

                Range("S6").Select
                ActiveSheet.Shapes.AddChart.Select
                ActiveChart.ChartType = xlXYScatter

```

```

'Create a plot to verify that the linear extrapolation
worked correctly
With ActiveChart
    .SeriesCollection.NewSeries
    .SeriesCollection(1).XValues = xVal
    .SeriesCollection(1).Values = yL
    .SeriesCollection(1).Select
    .SeriesCollection(1).Trendlines.Add xlLinear
    ActiveSheet.ChartObjects(2).Activate
    .SeriesCollection(1).Trendlines(1).Select
    Selection.DisplayEquation = True
    .SeriesCollection(1).Trendlines(1).DataLabel.Select
    Selection.Left = 120
    Selection.Top = 120
    ActiveSheet.Shapes(8).Left = ActiveSheet.Cells(54,
38).Left
    ActiveSheet.Shapes(8).Top = ActiveSheet.Cells(54,
38).Top
    ActiveSheet.Shapes(8).Height =
ActiveSheet.Range("AS54:AS67").Height
    ActiveSheet.Shapes(8).Width =
ActiveSheet.Range("AL67:AS67").Width
    End With

*****8 making the last new xy point at the
bore
ReDim yL(3, 0)
ReDim xVal(3, 0)
ReDim xL(3, 1)
Dim ExtractNum As Double
'ExtractNum = ActiveSheet.Range("ExtractNum").Value

With ActiveSheet
For i = rowCount / 2 To rowCount
    If .Range("NewCrackData").Cells(i, 6) =
"EXCLUDE" Then
        rowIndex = i - 1
        Exit For
    Else
        End If
    Next

For i = 0 To 3
    xL(i, 0) = .Range("newXY").Cells(rowIndex - i,
1).Value
    xL(i, 1) = .Range("newXY").Cells(rowIndex - i,
1).Value
    xL(i, 1) = (.Range("newXY").Cells(rowIndex - i,
1).Value) ^ 2
    yL(i, 0) = .Range("newXY").Cells(rowIndex - i,
2).Value
    xVal(i, 0) = .Range("newXY").Cells(rowIndex - i,
1).Value
    Next

End With

Range("S6").Select
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter

With ActiveChart

```

```

.SeriesCollection.NewSeries
.SeriesCollection(1).XValues = xVal
.SeriesCollection(1).Values = yL
.SeriesCollection(1).Select
.SeriesCollection(1).Trendlines.Add xlPolynomial, 2
.SeriesCollection(1).Trendlines.Add xlLinear
ActiveSheet.ChartObjects(2).Activate
.SeriesCollection(1).Trendlines(1).Select
Selection.DisplayEquation = True
.SeriesCollection(1).Trendlines(1).DataLabel.Select
Selection.Left = 120
Selection.Top = 120
ActiveSheet.Shapes(11).Left = ActiveSheet.Cells(236,
38).Left
ActiveSheet.Shapes(11).Top = ActiveSheet.Cells(236,
38).Top
ActiveSheet.Shapes(11).Height =
ActiveSheet.Range("AS236:AS249").Height
ActiveSheet.Shapes(11).Width =
ActiveSheet.Range("AL249:AS249").Width
    End With

Eq() = Application.WorksheetFunction.LinEst(yL, xVal)
ActiveSheet.Range("Linear4").Value = Eq(1)
ActiveSheet.Range("Constant4").Value = Eq(2)

' Eq() = Application.WorksheetFunction.LinEst(yL, xL)
' ActiveSheet.Range("Squared11").Value = Eq(1)
' ActiveSheet.Range("Linear11").Value = Eq(2)
' ActiveSheet.Range("Constant11").Value = Eq(3)

*****8Dont need this if through crack
For i = rowIndex + 1 To rowCount
' ActiveSheet.Range("newXY").Cells(i, 2).Value = Eq(1) *
ActiveSheet.Range("newXY").Cells(i, 1).Value ^ 2 + Eq(2)
* ActiveSheet.Range("newXY").Cells(i, 1).Value + Eq(3)
ActiveSheet.Range("newXY").Cells(i, 2).Value = Eq(1) *
ActiveSheet.Range("newXY").Cells(i, 1).Value + Eq(2)
Next

If s = 1 Then
' If ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value <= 0 Then
' ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value = 0
' ActiveSheet.Range("newXY").Cells(rowCount - 1,
2).Value = Eq(1) *
ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value ^ 2 + Eq(2) *
ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value + Eq(3)
' ActiveSheet.Range("newXY").Cells(rowCount,
1).Value = ActiveSheet.Range("newXY").Cells(rowCount
- 1, 1).Value
' ActiveSheet.Range("newXY").Cells(rowCount,
2).Value = ActiveSheet.Range("newXY").Cells(rowCount
- 1, 2).Value
' Else
' ActiveSheet.Range("newXY").Cells(rowCount,
1).Value = 0

```

```

' ActiveSheet.Range("newXY").Cells(rowCount,
2).Value = Eq(1) *
ActiveSheet.Range("newXY").Cells(rowCount, 1).Value ^
2 + Eq(2) * ActiveSheet.Range("newXY").Cells(rowCount,
1).Value + Eq(3)
' End If
If ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value <= 0 Then
    ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value = 0
    ActiveSheet.Range("newXY").Cells(rowCount - 1,
2).Value = Eq(1) *
ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value + Eq(2)
    ActiveSheet.Range("newXY").Cells(rowCount, 1).Value
= ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value
    ActiveSheet.Range("newXY").Cells(rowCount, 2).Value
= ActiveSheet.Range("newXY").Cells(rowCount - 1,
2).Value
Else
    ActiveSheet.Range("newXY").Cells(rowCount, 1).Value
= 0
    ActiveSheet.Range("newXY").Cells(rowCount, 2).Value
= Eq(1) * ActiveSheet.Range("newXY").Cells(rowCount,
1).Value + Eq(2)
    End If
Else
If s = 2 Then
'ActiveSheet.Range("newXY").Cells(rowCount, 1).Value =
ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value - 0.003
'ActiveSheet.Range("newXY").Cells(rowCount, 2).Value =
Eq(1) * ActiveSheet.Range("newXY").Cells(rowCount,
1).Value ^ 2 + Eq(2) *
ActiveSheet.Range("newXY").Cells(rowCount, 1).Value +
Eq(3)
ActiveSheet.Range("newXY").Cells(rowCount, 1).Value =
ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value - 0.003
ActiveSheet.Range("newXY").Cells(rowCount, 2).Value =
Eq(1) * ActiveSheet.Range("newXY").Cells(rowCount,
1).Value + Eq(2)

Else
If s = 3 Then
'ActiveSheet.Range("newXY").Cells(rowCount, 1).Value =
ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value - 0.003
'ActiveSheet.Range("newXY").Cells(rowCount, 2).Value =
Eq(1) * ActiveSheet.Range("newXY").Cells(rowCount,
1).Value ^ 2 + Eq(2) *
ActiveSheet.Range("newXY").Cells(rowCount, 1).Value +
Eq(3)
ActiveSheet.Range("newXY").Cells(rowCount, 1).Value =
ActiveSheet.Range("newXY").Cells(rowCount - 1,
1).Value - 0.02
ActiveSheet.Range("newXY").Cells(rowCount, 2).Value =
Eq(1) * ActiveSheet.Range("newXY").Cells(rowCount,
1).Value + Eq(2)

End If

```

```

End If

End If

*****

'verification
    ReDim yL(3, 0)
    ReDim xVal(3, 0)

    For i = 0 To 3
        xVal(i, 0) =
ActiveSheet.Range("newXY").Cells(rowCount - i, 1).Value
        yL(i, 0) =
ActiveSheet.Range("newXY").Cells(rowCount - i, 2).Value
    Next

    Range("S6").Select
    ActiveSheet.Shapes.AddChart.Select
    ActiveChart.ChartType = xlXYScatter

'Create a plot to verify that the linear extrapolation
worked correctly
With ActiveChart
    .SeriesCollection.NewSeries
    .SeriesCollection(1).XValues = xVal
    .SeriesCollection(1).Values = yL
    .SeriesCollection(1).Select
    .SeriesCollection(1).Trendlines.Add xlLinear
    ActiveSheet.ChartObjects(2).Activate
    .SeriesCollection(1).Trendlines(1).Select
    Selection.DisplayEquation = True
    .SeriesCollection(1).Trendlines(1).DataLabel.Select
    Selection.Left = 120
    Selection.Top = 120
    ActiveSheet.Shapes(12).Left = ActiveSheet.Cells(253,
38).Left
    ActiveSheet.Shapes(12).Top = ActiveSheet.Cells(253,
38).Top
    ActiveSheet.Shapes(12).Height =
ActiveSheet.Range("AS253:AS266").Height
    ActiveSheet.Shapes(12).Width =
ActiveSheet.Range("AL266:AS266").Width
End With

With ActiveSheet
    For i = 1 To rowCount
        .Range("newXY").Cells(i, 3).Value =
.Range("newXY").Cells(i, 1).Value +
.Range("HRadius").Cells(1, 1).Value
        .Range("newXY").Cells(i, 4).Value =
.Range("newXY").Cells(i, 2).Value +
.Range("Crack0").Cells(1, 2).Value
    Next
End With

Range("S6").Select
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter

With ActiveChart
    .SeriesCollection.NewSeries

```



```

        .SeriesCollection(1).Name = "Original"
        .SeriesCollection(1).XValues =
ActiveSheet.Range("$I$25:$I$30000")
        .SeriesCollection(1).Values =
ActiveSheet.Range("$J$25:$J$30000")
        .SeriesCollection.NewSeries
        .SeriesCollection(2).Name = "New"
        .SeriesCollection(2).XValues =
ActiveSheet.Range("$Y$25:$Y$30000")
        .SeriesCollection(2).Values =
ActiveSheet.Range("$Z$25:$Z$30000")
        .Axes(xlCategory).HasTitle = True
        .Axes(xlCategory).AxisTitle.Caption = "Crack Length
along Surface (in)"
        .Axes(xlValue).MinimumScale = 0
        .Axes(xlValue).MaximumScale =
ActiveSheet.Range("y53") + 0.1
        .Axes(xlCategory).MinimumScale =
ActiveSheet.Range("Crack0").Value
        .Axes(xlCategory).MaximumScale =
ActiveSheet.Range("Crack0").Value +
ActiveSheet.Range("w25")
        .Axes(xlValue).HasTitle = True
        .Axes(xlValue).HasMajorGridlines = False
        .Axes(xlValue).AxisTitle.Caption = "Crack Length at
Hole (in)"
        ActiveSheet.Shapes(13).Left = ActiveSheet.Cells(22,
47).Left
        ActiveSheet.Shapes(13).Top = ActiveSheet.Cells(22,
47).Top
        ActiveSheet.Shapes(13).Height =
ActiveSheet.Range("BD22:BD48").Height
        ActiveSheet.Shapes(13).Width =
ActiveSheet.Range("AU35:BD35").Width
    End With

```

'Making array of points to copy for new geometry
'leaving out points that are close together

```

'k = crackpts(0) + crackpts(1) + crackpts(2)
'crackpts(0) = 15
'crackpts(1) = 35
'crackpts(2) = 15
j = 0
k = crackpts(0) + crackpts(1) + crackpts(2)
ReDim newcrack(k - 1, 1)

```

With ActiveSheet

```

    '* Makse sure the final slope is no less than -0.5
'
'    l = rowCount
'
'    If .Range("newXY").Cells(l - 1, 5).Value < -0.41
Then
'    .Range("newXY").Cells(l, 4) = -1 * (-0.41 *
(.Range("newXY").Cells(l - 1, 3) -
.Range("newXY").Cells(l, 3)) - .Range("newXY").Cells(l -
1, 4))
'    End If

```

'Number of points skipped in the first and last 10% of the
crack front

```

    If .Range("ExtractNum") > 100 Then
        l = 3
    Else
        l = 2
    End If

```

For i = 1 To k

```

    If i < crackpts(0) Then
        newcrack(j, 0) = Range("newXY").Cells(i, 3)
        newcrack(j, 1) = Range("newXY").Cells(i, 4)
        i = i + 1
        j = j + 1
    
```

If (Range("newXY").Cells(i, 3) = newcrack(j - 1, 0))
And (Range("newXY").Cells(i, 4) = newcrack(j - 1, 1))

```

Then
    i = i + 1
End If

```

```

Else
End If

```

If (i < (crackpts(0) + crackpts(1))) And (i >= crackpts(0))
Then

```

        newcrack(j, 0) = Range("newXY").Cells(i, 3)
        newcrack(j, 1) = Range("newXY").Cells(i, 4)
        j = j + 1
    Else
    End If

```

If (i < (crackpts(0) + crackpts(1) + crackpts(2) - 3)) And
(i >= (crackpts(0) + crackpts(1) + 1)) Then

```

        newcrack(j, 0) = Range("newXY").Cells(i, 3)
        newcrack(j, 1) = Range("newXY").Cells(i, 4)
        i = i + 1
        j = j + 1
    Else
    End If

```

Next

```

    newcrack(j, 0) = Range("newXY").Cells(rowCount, 3)
    newcrack(j, 1) = Range("newXY").Cells(rowCount, 4)
    i = 1

```

End With

```

j = 0

```

End Sub

Private Function GetSetInfo(ByVal setname As String) As
StressCheck.OBJECTCOORD()

'Function gets information (element #'s, faces, etc.) about
set

```

Dim objcoord() As StressCheck.OBJECTCOORD
Dim faces As Variant, objs As Variant, numobjs As
Integer
Dim i As Integer

numobjs = SCSets.Set(setname).ObjectCount
faces = SCSets.Set(setname).ElementFaceList
objs = SCSets.Set(setname).ObjectList

ReDim objcoord(numobjs - 1)
For i = 0 To numobjs - 1
    objcoord(i).EdgeFaceNum = faces(i)
    objcoord(i).ObjectNum = objs(i)
    objcoord(i).Eta = 0.000001
    objcoord(i).Pita = 0.000001
    objcoord(i).Xi = 0.000001
    objcoord(i).Zeta = 0.000001
Next i

GetSetInfo = objcoord

End Function

Private Function GetSetNumObj(ByVal setname As String)
As Integer

'Gets object count from set
GetSetNumObj =
SCModel.Sets.Set(setname).ObjectCount

End Function

```

Module 2

```

Option Explicit
'These are the routines to oversee the iterative nature of the
calculations

Dim CurrentSheet As Worksheet

Sub CrackGrowth()
    Dim steps As Integer
    Dim i As Integer, j As Integer, k As Integer
    Dim NewSheet As Worksheet

    'By assumption all sheets will be named using the name
of the second sheet in the workbook as a base
    'The first sheet can have anything on it such as a
description
    'The second sheet must define the problem and all named
cells must have worksheet scope, not workbook scope.
    'The scope can be found in the name manager under the
formulas tab.
    'The calculation will always start with the last sheet.

    steps = ActiveWorkbook.Worksheets.Count - 1
    i = 0

```

```

k = 0
Do
    Set CurrentSheet =
ActiveWorkbook.Worksheets(ActiveWorkbook.Worksheet
s.Count)
    CurrentSheet.Activate

    'Need to set some exit criterion
    If i >= ActiveSheet.Range("Iterations").Value Then
        Exit Do
    End If
    Dim path As String
    path = ActiveSheet.Range("path")

    'Do the calculation
    Err.Clear
    On Error Resume Next
    Module1.CrackInitiation i, path
    ' If Err.Number <> 0 Then
    ' Err = 0
    If Err.Number <> 0 Then
        If k < 1 Then
            MsgBox "The crack routine has encountered a
problem: " & VBA.Chr$(13) & Err.Description
            Module3.TerminateStressCheck
            k = k + 1
        Else
            MsgBox "The crack routine has encountered a
problem: " & VBA.Chr$(13) & Err.Description
            Exit Do
        End If
    '*End If
    Else

        'Create the new sheet
        CurrentSheet.Copy After:=CurrentSheet
        k = CurrentSheet.Index
        Set NewSheet = Worksheets.Item(k + 1)
        NewSheet.Name = Worksheets(2).Name & " " &
VBA.Str(steps)
        steps = steps + 1
        'transfer result

        NewSheet.Range("radius1").Value =
NewSheet.Range("newr1").Value
        NewSheet.Range("radius2").Value =
NewSheet.Range("newr2").Value
        NewSheet.Range("ellx").Value =
NewSheet.Range("newellx").Value
        NewSheet.Range("elly").Value =
NewSheet.Range("newelly").Value
        Dim rowCount As Double

        rowCount = Range(ActiveSheet.Range("W25"),
ActiveSheet.Range("W65535").End(xlUp)).Count
        If ActiveSheet.Range("CrackType").Value = 2 Or 3
Then

```

```

    NewSheet.Range("int").Value =
NewSheet.Range("csinky").Value -
NewSheet.Range("newXY").Cells(1, 2)
    NewSheet.Range("int").Value =
NewSheet.Range("newXY").Cells(rowCount, 1)
    NewSheet.Range("int").Value =
NewSheet.Range("newXY").Value / 2
    Else
        If ActiveSheet.Range("newXY").Cells(1, 1).Value <=
ActiveSheet.Range("newXY").Cells(rowCount, 2) Then
            NewSheet.Range("int").Value =
NewSheet.Range("newXY").Value
        Else
            NewSheet.Range("int").Value =
NewSheet.Range("newXY").Cells(rowCount, 4)
        End If
    End If
End If

```

```

NewSheet.Range("c25:c1000").ClearContents
NewSheet.Range("d25:d1000").ClearContents

```

```

NewSheet.Range("h25:h1000,i25:i1000,n25:n1000,m25:m
1000,l25:l1000,k25:k1000,j25:j1000").ClearContents

```

```

NewSheet.Range("o25:o1000,p25:p1000,q25:q1000,r25:r1
000,s25:s1000,t25:t1000,u25:u1000").ClearContents

```

```

NewSheet.Range("v25:v1000,w25:w1000,x25:x1000,y25:y
1000,z25:z1000").ClearContents

```

```

    For j = 0 To (UBound(newcrack))
        NewSheet.Range("CrackFront").Cells(j + 1,
1).Value = newcrack(j, 0)
        NewSheet.Range("CrackFront").Cells(j + 1,
2).Value = newcrack(j, 1)
    Next j

```

```

NewSheet.Range("MissXpts") = 0

```

```

NewSheet.Activate
ActiveWorkbook.Save
i = i + 1
k = 0

```

```

End If

```

```

Loop

```

```

Module3.TerminateStressCheck

```

```

End Sub

```

Module 3

```

Option Explicit

```

```

Sub TerminateStressCheck()

```

```

'-----TerminateStressCheck
Routine-----

```

```

'-----
'
' :Terminates a process. First checking to see
if it is running or not
' :Uses Windows Management
Instrumentation to query all running process
' :then terminates ALL instances of the
specified process held
' :in the variable strTerminateThis.
' :
' :
' :***WARNING: This will terminate a
specified running process, use with caution!
' :***Terminating certain processes can effect
the running of
' :***windows and/or running applications.
'-----

```

```

Dim strTerminateThis As String 'The variable to hold the
process to terminate

```

```

Dim objWMIcimv2 As Object

```

```

Dim objProcess As Object

```

```

Dim objList As Object

```

```

Dim intError As Integer

```

```

strTerminateThis = "StressCheck.exe" 'Process to
terminate,

```

```

'change notepad.exe to the process you want to
terminate

```

```

Set objWMIcimv2 = GetObject("winmgmts:" _
& "{impersonationLevel=impersonate}!\.\root\cimv2")
'Connect to CIMV2 Namespace

```

```

Set objList = objWMIcimv2.ExecQuery _
("select * from win32_process where name='" &
strTerminateThis & "'") 'Find the process to terminate

```

```

If objList.Count = 0 Then 'If 0 then process isn't running

```

```

' MsgBox strTerminateThis & " is NOT running." &
vbCr & vbCr _
' & "Exiting procedure.", vbCritical, "Unable to
continue"

```

```

' Set objWMIcimv2 = Nothing

```

```

' Set objList = Nothing

```

```

' Set objProcess = Nothing

```

```

Exit Sub

```

```

Else

```

```

'Ask if OK to continue

```

```

' Select Case MsgBox("Are you sure you want to
terminate this running process?:" _

```

```

' & vbCrLf & "" _

```

```

' & vbCrLf & "Process name: " & strTerminateThis

```

```

' & vbCrLf & "" _

```

```

' & vbCrLf & "Note:" _

```

```

' & vbCrLf & "Terminating certain processes can
effect the running of Windows" _

```

```

' & "and/or running applications. The process will
terminate if you OK it, WITHOUT " _

```

```
' & "giving you the chance to save any changes in
anything that is running in the specified process above." _
', vbOKCancel Or vbQuestion Or vbSystemModal
Or vbDefaultButton1, "WARNING:")
```

```
' Case vbOK
' OK to continue with terminating the process
For Each objProcess in objList

    intError = objProcess.Terminate 'Terminates a
process and all of its threads.
'Return value is 0 for success. Any other number
is an error.
    If intError <> 0 Then
        MsgBox "ERROR: Unable to terminate that
process.", vbCritical, "Aborting"
        Exit Sub
    End If
Next
'ALL instances of specified process
(strTerminateThis) has been terminated
' Call MsgBox("ALL instances of process " &
strTerminateThis & " has been successfully terminated.", _
vbInformation, "Process Terminated")
```

```
Set objWMicimv2 = Nothing
Set objList = Nothing
Set objProcess = Nothing
Exit Sub

' Case vbCancel
' NOT OK to continue with the termination, abort
Set objWMicimv2 = Nothing
Set objList = Nothing
Set objProcess = Nothing
Exit Sub
End Select
End If
```

End Sub

Module 4

Option Explicit

'These are the routines to oversee the iterative nature of the calculations

Dim CurrentSheet As Worksheet

'-----Determine Angle-----

Sub FindAngle()

Dim i As Integer, j As Integer, k As Double, l As Double

Dim Xm As Double, Ym As Double, a45 As Double, a20

As Double, a80 As Double

Dim K80(1) As Variant, K85(1) As Variant, K87(1) As Variant

Dim q As Integer, r As Integer, s As Integer, maxX As Double

Dim NewSheet As Worksheet

Application.ScreenUpdating = False

q = ActiveSheet.Range("Angle1").Value

r = ActiveSheet.Range("Angle2").Value

s = ActiveSheet.Range("Angle3").Value

For j = 1 To ActiveWorkbook.Worksheets.Count - 1

K80(0) = 0

K85(0) = 0

Worksheets(ActiveSheet.Index + j).Select 'Take required values from each sheet and put them into the first sheet

With ActiveSheet

Xm =

Application.WorksheetFunction.Max(Range("c25:c1000")) - .Range("HRadius").Value

Ym =

Application.WorksheetFunction.Max(Range("d25:d1000"))

maxX =

Application.WorksheetFunction.Max(Range("p25:p1000"))

'determining the angle closest to what is desired

For i = 2 To

Application.WorksheetFunction.Count(Range("z25:z1000"))

l = (180 / 3.14159265358979) *

(Atn(.Range("newXY").Cells(i - 1, 2).Value / (.Range("newXY").Cells(i - 1, 1).Value)))

k = (180 / 3.14159265358979) *

(Atn(.Range("newXY").Cells(i, 2).Value / (.Range("newXY").Cells(i, 1).Value + 0.00000001)))

If K80(0) = 0 Then

If (Abs(k - q) > Abs(l - q)) Then

K80(0) = .Range("newCrackData").Cells(i - 1, 7)

K80(1) = l

a20 = .Range("newCrackData").Cells(i - 1, 11)

End If

End If

If K85(0) = 0 Then

If (Abs(k - r) > Abs(l - r)) Then 'And (i <

Application.WorksheetFunction.Count(Range("aa25:aa100"))

))) Then

K85(0) = .Range("newCrackData").Cells(i - 1, 7)

K85(1) = l

a45 = .Range("newCrackData").Cells(i - 1, 11)

Else

If i =

Application.WorksheetFunction.Count(Range("z25:z1000"))

)) Then

K85(0) = .Range("newCrackData").Cells(i, 7)

K85(1) = k

a45 = .Range("newCrackData").Cells(i - 1, 11)

End If

End If

End If

```

        If (Abs(k - s) > Abs(l - s)) Then 'And (i <
Application.WorksheetFunction.Count(Range("aa25:aa100
")))) Then
        K87(0) = .Range("newCrackData").Cells(i - 1, 7)
        K87(1) = 1
        a80 = .Range("newCrackData").Cells(i - 1, 11)
        Exit For
    Else
        If i =
Application.WorksheetFunction.Count(Range("z25:z1000"
)) Then
        K87(0) = .Range("newCrackData").Cells(i, 7)
        K87(1) = k
        a80 = .Range("newCrackData").Cells(i - 1, 11)
        End If
    End If

    Next i
End With

Worksheets(ActiveSheet.Index - j).Select
With ActiveSheet
    .Range("Iteration").Cells(j + 1, 1).Value = j
    .Range("Xmax").Cells(j + 1, 1).Value = Xm
    .Range("Ymax").Cells(j + 1, 1).Value = Ym
    .Range("AspectRatio").Cells(j + 1, 1).Value = Ym /
(Xm)
    .Range("Kmin").Cells(j + 1, 1).Value = K80(0)
    .Range("Kmid").Cells(j + 1, 1).Value = K85(0)
    .Range("Kmax").Cells(j + 1, 1).Value = K87(0)
    .Range("Kmin").Cells(j + 1, 4).Value = K80(1)
    .Range("Kmid").Cells(j + 1, 4).Value = K85(1)
    .Range("Kmax").Cells(j + 1, 4).Value = K87(1)
    .Range("Atwozero").Cells(j + 1, 1).Value = a20
    .Range("Afourfive").Cells(j + 1, 1).Value = a45
    .Range("Aeightzero").Cells(j + 1, 1).Value = a80
    .Range("MaxX").Cells(j + 1, 1).Value = maxX
End With
Worksheets(ActiveSheet.Index + 1).Select

Next

With ActiveSheet
For j = 1 To ActiveWorkbook.Worksheets.Count - 1

    .Range("Cycles").Cells(2, 1).Formula =
"=(M18)/($D$7*((F18)^$D$6))"
    .Range("Cycles").Cells(2, 1).Copy
    .Range("Cycles").Cells(j + 2, 1).PasteSpecial
    .Range("Cycles").Cells(j + 1, 2).PasteSpecial
    .Range("Cycles").Cells(j + 1, 3).PasteSpecial

    .Range("Cycles").Cells(2, 5) =
.Range("InitialCycles").Value
    .Range("Cycles").Cells(2, 6) =
.Range("InitialCycles").Value
    .Range("Cycles").Cells(2, 7) =
.Range("InitialCycles").Value
    .Range("Cycles").Cells(3, 5).Formula = "=V18+R18"
    .Range("Cycles").Cells(3, 5).Copy
    .Range("Cycles").Cells(j + 2, 6).PasteSpecial
    .Range("Cycles").Cells(j + 2, 7).PasteSpecial
    .Range("Cycles").Cells(j + 2, 5).PasteSpecial
Next
End With

Application.ScreenUpdating = True

End Sub

```