Computational Framework for Fracture of Graphite Bricks in an AGR Core

Costy Kodsi

Submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

School of Engineering
University of Glasgow

April, 2017

Abstract

Life-extension of EDF Energy's existing nuclear fleet is based on an assumption of continued safe operation. Potential fracture of graphite bricks in the nuclear reactor core of a power station represents an unknown variable in the equation. An understanding of the nature of this phenomenon and the impact on operation of the power station is desired. This work prepares the way for the future study of fracture in graphite bricks in a reactor core subject to dynamic excitation. Methodology to couple a multi-body finite element contact code to a crack propagation code is thus developed. Three important scientific contributions have been made:

(i) An optimisation problem formulated on a smooth manifold to yield the rotation responsible for infinitesimal rigid body motion. This involves an iterative scheme in the form of Newton's method that takes into account the geometry of the underlying parameter space. There are no issues with singularities or additional computations in each iteration to scale the solution onto the manifold.

(ii) An energy consistent crack initiation criterion for brittle material where nucleation is treated as a sudden and discrete rupture event at the macroscopic level. At the heart of the criterion is the finite difference form of the energy release rate; an expression for the characteristic length is derived and the change in total potential energy is obtained from an asymptotic argument involving the topological derivative. The criterion can predict crack onset at a sharp or blunt notch. Fracture toughness and material strength are the only input requirements.

(iii) Algorithms related to the detection of sharp notches in a tetrahedral finite element mesh and a general computational procedure for evaluation of non-local crack initiation criteria. The only tool in the implementation of these algorithms is C++11. There is no need for a complex data structure storing all incidence information. Unordered associative containers in the standard library are exploited in the design of these rather efficient algorithms, which cover surface extraction and provide connectivity of the edges representing a sharp notch tip. A mesh re-generation routine for purposes of refinement at the sharp notch tips has also been developed.

Contents

List of Tables

List of Figures

List of Functions

Acknowledgements

I am extremely grateful to Prof. Chris Pearce for the opportunity to undertake a research doctorate. A great debt is owed to Dr. Tomasz Koziara. If it were not for his support and patience, this work would not have seen the light of day. Many thanks are also due to Ms. Larisa Viviana Hodea for help with the drawings in Chapter 3.

Financial support was received from EDF Energy Nuclear Generation Ltd and EPSRC (*grants: EP/P504937/1, EP/P505534/1, EP/L50497X/1, EP/J500434/1*).

Declaration

The research contained in this thesis is entirely the outcome of my own work and has not been submitted for a degree or diploma at any other academic or professional institution. All contributing sources have been fully acknowledged.

Nomenclature

General

$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of strictly positive real numbers
$\mathbb{Z}$	set of integers
$\mathbb{N}$	set of non-negative integers
$\mathbb{R}^d$	Euclidean space
$GL(n, \mathbb{R})$	set of real $n \times n$ invertible matrices
$GL^+(n)$	set of real $n \times n$ matrices with positive determinant
$O(n)$	set of real $n \times n$ orthogonal matrices
$SO(n)$	set of real $n \times n$ rotation matrices
$S(n)$	set of real $n \times n$ symmetric matrices
$\mathfrak{so}(n)$	set of real $n \times n$ skew-symmetric matrices
$SPD(n)$	set of real $n \times n$ symmetric, positive definite matrices
I	arbitrary index set
$\mathcal{O}$	error term / algorithm classification
T_r	triangular number
$\mathbf{e}_i$	unit basis vector; $\mathbf{e}_i \in \mathbb{R}^d$
$\mathbf{I}$	second-order identity tensor
$\mathbf{II}$	fourth-order identity tensor

Acronyms

AGR	advanced gas-cooled reactor
FEM	finite element method
LEFM	linear elastic fracture mechanics
SED	strain energy density

Chapter 2

a (scalar) cost function

$\mathcal{A}^+$ maximal atlas

$\mathbf{B}$ left Cauchy-Green tensor

$\mathcal{B}$ bounded and connected open subset of $\mathbb{R}^3$

$\partial\mathcal{B}$ boundary of $\mathcal{B}$

$\mathbf{C}$ right Cauchy-Green tensor

ds differential surface element

dy differential volume element

$\mathbf{E}$ infinitesimal strain

$\mathbf{F}$ deformation gradient

$\mathbf{g}$ gradient vector

$\mathbf{H}$ Hessian matrix

$\mathbf{J}^i$ basis tangent vector

$\mathbf{K}$ tangent vector

$\mathbf{L}$ skew-symmetric matrix

$\mathbf{m}$ outward unit normal to boundary in current configuration

M topological space

$\mathcal{M}$ smooth manifold

$\mathbf{n}$ outward unit normal to boundary in reference configuration

$\mathbf{p}$ material point

P open set; $P \subseteq M$

$\mathbf{q}$ material point

$\mathbf{r}$ material point

$\mathbf{R}$ rotation tensor

t constant; $t \in \mathbb{R}$

$T_e(SO(3))$ tangent space at the identity

$T_{\mathbf{R}}(SO(3))$ tangent space at $\mathbf{R}$

$\mathbf{u}$ displacement

$\mathbf{U}$ right stretch tensor

$\mathbf{V}$ left stretch tensor

$\mathbf{W}$ infinitesimal rotation

$\mathbf{x}$ spatial point

$\mathbf{Y}$ tangent vector

$\mathbf{Z}$ tangent vector

γ error tolerance

λ Newton decrement

ϕ map; $P \to \mathbb{R}^d$

$\boldsymbol{\varphi}$ mapping function (continuum mechanics)

θ constant; $\theta \in \mathbb{R}$

Chapter 3

a variable crack length

a_c characteristic length

$\boldsymbol{C}$ fourth-order isotropic elasticity tensor

ds differential surface element

dy differential area element

D_T first-order topological derivative

E Young's modulus

F, F_c load, subscript denotes critical value

$\mathbf{g}$ traction prescribed on the boundary

G, G_c energy release rate, subscript denotes critical value

$\mathbf{h}$ displacement prescribed on the boundary

H_I, H_{II} constants, Mode-I and Mode-II

J_I^{-1}, J_{II}^{-1} coefficients, Mode-I and Mode-II

K_I Mode-I stress intensity factor associated with a crack

K_I^{η}, K_{II}^{η} generalised stress intensity factors, Mode-I and Mode-II

K_{Ic} fracture toughness

l existing crack length in body

m_{ij}^{I}, m_{ij}^{II} angular functions, Mode-I and Mode-II

M_σ stress failure function

$\mathbf{n}$ outward unit normal vector to boundary

N, N_c normal force, subscript denotes critical value

p finite length

P, P_c tensile load, subscript denotes Mode-I fracture value

q horizontal distance from centre of U-notched specimen

Q change in total potential energy (domain perturbation)

$\mathscr{R}$ remainder function (asymptotic expansion)

S, S_c strain energy density factor, subscript denotes critical value

$\mathbf{t}$ unit tangential vector to boundary

T shear load

$\mathbf{u}$ displacement

U kinetic energy

$\mathbf{v}$ shape change velocity

V shape change speed

w_c modified McClintock criterion distance parameter

W strain energy

$\mathbf{x}$ material point

$\mathbf{x}^\tau$ spatial point

Y non-dimensional constant related to crack location

z_c Novozhilov-Seweryn criterion characteristic length parameter

α angle; $\alpha = \pi - \beta$

β wedge semi-angle

χ arbitrary shape functional

$\boldsymbol{\varepsilon}$ infinitesimal strain

η_I, η_{II} exponents, Mode-I and Mode-II

γ fracture surface energy density

ι_c shear strength

κ constant representing plane stress or plane strain

λ Lamé modulus

μ shear modulus

ν Poisson's ratio

ϕ_1, ϕ_2 coefficients; $\phi_1 = \frac{1}{2}(\sigma_1 + \sigma_2)$ and $\phi_2 = \frac{1}{2}(\sigma_1 - \sigma_2)$

$\boldsymbol{\varphi}$ mapping (motion) function

Φ scalar variable

Π total potential energy functional

ψ ratio of tensile to shear load

Ψ scalar variable

ρ notch radius

ϱ_c minimum strain energy density criterion radial distance parameter

$\boldsymbol{\sigma}$ Cauchy stress

σ_1, σ_2 principal stress values

σ_B atomic lattice strength

σ_c tensile strength

$\boldsymbol{\sigma}_D$ deviatoric stress tensor

$\sigma_{\max}$ maximum principal stress

σ_∞ remote tensile stress

ς constant representing plane stress or plane strain

$\boldsymbol{\Sigma}$ second-order Eshelby energy-momentum tensor

Υ fracture surface energy

$\boldsymbol{\zeta}$ convective rate of change of displacement

(r, θ) polar coordinate system

Geometry

Ω bounded and connected open subset of $\mathbb{R}^2$

$\partial\Omega$ boundary of Ω

Γ_D part of the boundary $\partial\Omega$ subject to a prescribed displacement; $\Gamma_D \subset \partial\Omega$

Γ_N part of the boundary $\partial\Omega$ subject to a prescribed traction; $\Gamma_N \subset \partial\Omega$

$\mathbf{x}_0$ centre point of domain perforation; $\mathbf{x}_0 \in \Omega$

ξ parameter associated with the size of domain perturbation

ω fixed bounded and connected open subset of $\mathbb{R}^2$

$\partial\omega$ boundary of ω

ω_ξ small domain; $\omega_\xi = \mathbf{x}_0 + \xi\omega$

$\partial\omega_\xi$ boundary of ω_ξ

Ω_ξ perturbed domain; $\Omega_\xi = \Omega \backslash \overline{\omega_\xi}$

$\partial\Omega_\xi$ boundary of Ω_ξ; $\partial\Omega_\xi = \partial\Omega \cup \partial\omega_\xi$

τ domain transformation parameter; $\tau \in \mathbb{R}^+$

Ω_τ transformed domain; $\Omega_\tau = \Omega_{\xi+\delta\xi}$

$\partial\Omega_\tau$ boundary of Ω_τ; $\partial\Omega_\tau = \partial\Omega_{\xi+\delta\xi}$

Chapter 4

c_i stress intensity factor (crack)

c_i^η, c_i^ν generalised stress intensity factors

$\mathbf{f}^i$ ith angular function (crack)

$\mathbf{g}^i$ ith angular function (V-notch)

$\mathbf{h}^i$ ith angular function (3-d)

H hyperplane

$\mathbf{n}$ outward unit normal to boundary

$\mathbf{n}_e$ outward unit edge normal

P d-dimensional convex polytope

$\mathbf{Q}$ orthogonal tensor

S_d d-simplex

∂S_d d-simplex boundary

$\mathbf{u}$ displacement

α angle; $\alpha = \pi - \beta$

β (V-notch) wedge semi-angle

χ user-set constant

η_i exponent (V-notch)

ν_i exponent (3-d)

$\boldsymbol{\sigma}$ Cauchy stress

θ_e dihedral angle

θ_{thresh} angle threshold (sharp notch tip detection)

ξ distance parameter

(r, θ) polar coordinate system

(r, θ, z) cylindrical coordinate system

(ρ, θ, ϕ) spherical coordinate system

Chapter 1

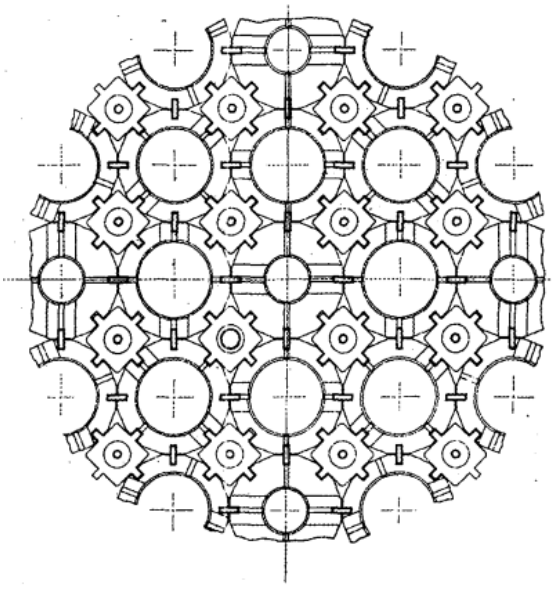
Introduction

1.1 Motivation

Hunterston-B nuclear power station made the national headlines on the 6th of October 2014. It was reported in [1] that cracks were found in the reactor core. Hunterston-B operates an Advanced Gas-Cooled Reactor (AGR) core that went online in 1976. The core is comprised of a multi-layer arrangement of graphite bricks (see Figure 1.1a) that are loosely connected. A steel plate system supports the bottom of the core. The sides of the core are also restrained in a steel structure that provides stability in case of dynamic loading, such as in a seismic event.

Graphite bricks are stacked (free-standing) in vertical columns. There are two types of graphite bricks in an AGR core differing in geometry and dimensions: (i) fuel bricks with a hollow circular inner profile, and (ii) interstitial bricks that are designed to receive control rods and coolant. Control rods regulate the fission rate in the core through neutron absorption. The fuel bricks in vertical columns provide a continuous channel for the slightly enriched uranium pellets that are stored in fuel stringers and direct the carbon dioxide coolant flow. Typically, there are 320 fuel channels in an AGR core. A system of loose and integral keys in keyways as shown in Figure 1.1b connects the graphite bricks. This design permits limited movement of the graphite bricks.

(a)



(b)

Figure 1.1. (a) Fuel and interstitial graphite bricks [1] and (b) inter-connectivity system [2].

Graphite bricks in an AGR core act as a moderator to slow the neutrons and increase the likelihood of fissionable reactions. Irradiation of the graphite bricks, across their service lifetime, induces dimensional change leading to internal stresses and distortion of the original shape. The fundamental operational safety requirement is that free movement of the control rods as well as fuel is unimpeded and that there is an uninterrupted adequate gas flow supply to cool both the fuel and core.

Hunterston-B was intended to generate electricity for a period of thirty years. However, there are plans to prolong the service lifetime of the power station. Therefore, cracks emanating from keyway corners that have not been observed before – even if only in two graphite bricks – are a cause for concern. Consequently, there is a need to develop safety arguments for the continued operation of the AGR core accounting for fractured graphite bricks. Computational modelling is seen as one way to meet this end.

1.2 Objective

The aim of this research project is to develop the theory and methodology for coupling SOLFEC [3], a dynamic multi-body finite element contact code, with the crack propagation capable finite element code MoFEM [4] to be able to study fracture in graphite bricks in an

AGR core subject to dynamic excitation. Results of this work are meant to represent a first step on the path leading towards a comprehensive thermodynamically justified computational framework that can augment safety studies for AGR core life-extension.

1.3 Assumptions

When engineering a solution to a problem, assumptions serve to constrain the design space. Ideally, assumptions reduce the complexity of the problem without sacrificing solution accuracy or jeopardising thermodynamic laws. The result is a more manageable problem in terms of mathematical representation, computational implementation and processing. In this research, the following simplifying assumptions are adopted:

i. Graphite is treated as a homogeneous and isotropic linear elastic material that is in a brittle state;

ii. Fracture of a graphite brick does not influence contact force evolution on the surface with time, i.e. crack propagation occurs faster than a change in the boundary conditions; and

iii. Cracks do not branch.

Nuclear-grade graphite is increasingly brittle with irradiation [5] justifying the first assumption. From the second assumption, failure as an instantaneous event is inferred. This is supported by the fact that crack propagation is unstable in graphite bricks [6]. The third assumption simply reduces the demands put on the computational modelling of crack evolution but is not based on any physical observation.

1.4 Computational modelling of fracture

A summary of SOLFEC and MoFEM is now given. The methodology for coupling is then discussed.

1.4.1 SOLFEC

There are two approaches to contact dynamics. In the first, friction is a function of the velocity. This is the regularised approach and is adopted in commercially available thermo-mechanical finite element software packages, where correcting springs are employed when constraints are violated. The use of springs can lead to numerical instabilities and inaccurate results. The non-smooth approach, on the other hand, can be more accurate but does not enjoy the same level of popularity. The mathematics is more complicated and the computational implementation more involved. Friction contact laws are expressed as set-valued force laws, which take the form of non-continuous functions. Nonetheless, tools from convex analysis can still be applied to these functions. SOLFEC implements an implicit formulation of contact dynamics as developed by Moreau [7] and Jean [8]. This methodology allows for a more accurate depiction of reality and larger time-steps than in the regularised approach.

SOLFEC was developed with high performance computing in mind from the outset, making it an ideal choice for modelling a large number of interacting bodies. Unfortunately, the parallelisation capability does not extend to the finite element analysis part of the code. Furthermore, only continuous piecewise linear finite elements of the tetrahedral and hexahedral type are provided. To date, graphite bricks in AGR core analyses conducted in SOLFEC have been modelled with a very coarse mesh as illustrated in Figure 1.2. While it is possible to increase the number of elements, this has to be based on a compromise between the required level of accuracy in the solution and practical simulation run-times.

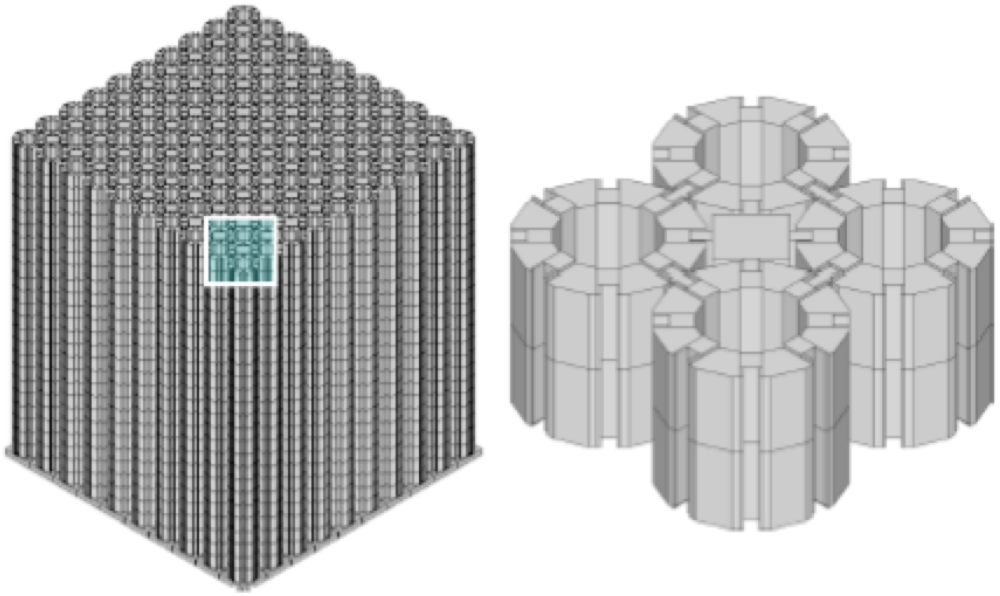


Figure 1.2. Simplified graphite core subject to seismic excitation [9].

1.4.2 MoFEM

MoFEM is a multi-physics finite element analysis code that efficiently deals with an arbitrary level of approximation and supports mesh refinement. Crack propagation is a prominent feature of MoFEM and is formulated for a homogeneous and linear elastic body. A global dissipation argument in the form of the Clausius-Duhem inequality is exploited to provide a consistent thermodynamic framework for brittle fracture. The elastic equilibrium response follows naturally from this. Then, the configurational force [10] that maximises the local energy dissipation at the crack front provides the crack propagation direction. Of course, this is subject to satisfaction of an energy condition. The crack is extended a small finite length, followed by a repeat of the energy analysis. This process of crack extension and energy interrogation continues until the condition is no longer fulfilled. The mesh surrounding the crack is modified to accommodate propagation. This involves movement of the nodes on the crack front in the propagation direction, whilst edge decimation/flipping and possibly mesh refinement maintains the validity of the mesh.

The theory underpinning this methodology is attributed to Miehe et al. [11]. Kaczmarczyk et al. [12] provide an advanced numerical treatment of this theory, which is realised in MoFEM. However, the splitting of elements suggested in [12] for purposes of mesh modification has been superseded in MoFEM as discussed in the previous paragraph. The improved implementation produces smooth crack fronts in contrast to split elements with varying dimensions. A smooth crack front is shown in Figure 1.3, which is from an example problem executed in MoFEM.

A limitation of the fracture methodology in MoFEM is that there is no automatic crack initiation capability. Currently, user interaction is required to identify the location and propagation direction.

It is worth noting that the mesh employed in a crack propagation analysis is significantly more dense than that used in SOLFEC. Solving for stress in a linear elastic body results in a singularity solution at the crack tip. Accordingly, a greater number of finite elements and/or finite elements with a higher polynomial degree are required to resolve the steep gradient in the local region. Additionally, MoFEM only supports tetrahedral elements for crack propagation analyses.

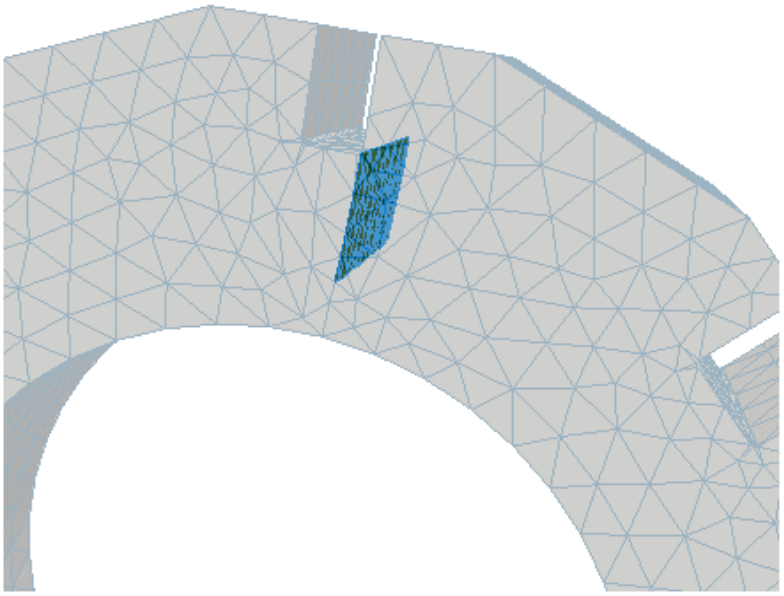


Figure 1.3. Crack propagation in graphite brick example [13].

1.4.3 Coupling methodology

Detection of fracture in a graphite brick is problematic as the finite element mesh density in a SOLFEC analysis is insufficient for resolving the solution with any level of accuracy. Consequently, there is no real measure for crack onset prediction. At most, the strain energy of a body can be monitored. However, no distinction can be made between impact and fracture. As it stands, the time-step for fracture of a body must be user-instigated.

Once fracture is deemed to occur in a graphite brick, the reference configuration of the geometry along with the surface displacement and contact force data is output at the identified time-step. There is a rigid component in the displacement that must be dealt with. This is the first research topic tackled in the project. The mesh representing the original geometry is then re-generated, comprised solely of tetrahedral elements, in preparation for MoFEM.

Surface displacement data at the contact force sites constrain the problem in MoFEM. This is justifiable, as the movement of graphite bricks is restricted in the core and because the given boundary conditions drive propagation until there is a separation of the body. If rigid body motion in the displacement is not mitigated, the direction of crack propagation will be artificially affected resulting in a physically non-realistic path. For crack propagation to take place, the location and propagation direction of a crack must be first determined. Accordingly, the second research topic is concerned with the development of a criterion for crack initiation from sharp as well as blunt notches. The latter requirement is to accommodate

crack propagation from the circular inner profile of a graphite fuel brick. In order to evaluate the criterion at or near a sharp notch tip, the relevant edge or edges in the mesh must be known computationally. This is the subject of the third research topic.

1.5 Thesis outline

Chapter 2 addresses the issue of rigid body motion of the graphite brick output by SOLFEC. Translation and rotation are treated separately. A cost function based on the skew-symmetric part of the displacement gradient is defined for the latter. An optimisation scheme is formulated on the manifold $SO(3)$ in order to determine the rigid body rotation. This methodology is fully documented in the chapter.

Chapter 3 develops the notion of crack nucleation based on a sudden and discrete rupture event at the macroscopic level. This takes place in a two-dimensional setting. A topological asymptotic expansion is employed to evaluate the total potential energy change associated with domain perturbation representative of finite-length crack extension. Additionally, the chapter reviews important existing fracture criteria. Numerous examples relating to the successful application of the developed criterion are presented.

Chapter 4 investigates the automatic application of two-dimensional non-local criteria in a three-dimensional context. Feature detection algorithms for sharp notch tip identification, and mesh re-generation form an important part of the chapter. Actual source code is disclosed in place of pseudo-code. This serves not only to simplify but also to elucidate algorithm design. A general computational procedure for application of non-local criteria is given. Operations relating to stress tensor transformation and optimisation to find the direction of propagation are explained. Examples of crack onset predictions made by a specific criterion are included.

Chapter 5 provides a summary of achievements in this project and critically reviews the strategy for the computational modelling of fracture in graphite bricks. Avenues for further research and project extension are presented.

1.6 Mathematical prerequisites and notation

In keeping with convention, the set of real numbers and integers are denoted $\mathbb{R}$ and $\mathbb{Z}$, respectively. $\mathbb{R}^+$ stands for the set of strictly positive real numbers. $\mathbb{N}$ is the set of natural numbers, corresponding to the non-negative elements of $\mathbb{Z}$:

$$\mathbb{N} = \{z \in \mathbb{Z} \,|\, z \geq 0\}. \tag{1.6.1}$$

Let $\mathbb{R}^d$ represent Euclidean space. An element of $\mathbb{R}^d$ is an ordered d-tuple of real numbers, $\mathbf{x} = (x_1, \ldots, x_d)$. Given vectors $\mathbf{a}$ and $\mathbf{b}$, their inner product is defined as $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + \cdots + a_d b_d$. The norm or length of any vector, say $\mathbf{a}$, is $\|\mathbf{a}\| = (\mathbf{a} \cdot \mathbf{a})^{1/2}$. If vectors $\mathbf{a}, \mathbf{b} \neq \mathbf{0}$, the angle between them is obtained from

$$\theta = \cos^{-1}\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|}\right). \tag{1.6.2}$$

Furthermore, the distance between points $\mathbf{a}$ and $\mathbf{b}$ is given by

$$\|\mathbf{a} - \mathbf{b}\| = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2}. \tag{1.6.3}$$

Euclidean space is not endowed naturally with an orientation. It is, in fact, the specification of a coordinate system that induces a preferred orientation. A Cartesian coordinate system consists of a fixed point $\mathbf{o}$ called the origin and basis (linearly independent) vectors — $\mathbf{e}_1$, $\mathbf{e}_2$ for $d = 2$ and $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ for $d = 3$ — that are mutually orthogonal, $\mathbf{e}_i \cdot \mathbf{e}_j = 0$ for $i \neq j$, and unitary, $\|\mathbf{e}_i\| = 1$. The basis is said to be orthonormal as the vectors satisfy

$$\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}, \tag{1.6.4}$$

where δ_{ij} is the Kronecker delta:

$$\delta_{ij} = \begin{cases} 0 & \text{for } i \neq j, \\ 1 & \text{for } i = j. \end{cases} \tag{1.6.5}$$

An axis of the system is a line emanating from the origin point containing a basis vector. Arrangement of the axes according to a "hand-rule", i.e. right-handed or left-handed, is where the orientation comes from. The right-hand Cartesian coordinate system is prevalent in the mathematical sciences and is assumed from here on in. For $d = 3$, this equates to the scalar triple product

$$(\mathbf{e}_1 \times \mathbf{e}_2) \cdot \mathbf{e}_3 = 1, \tag{1.6.6}$$

in which $\times$ is the cross product symbol (only between vectors). Geometrically, a scalar triple product represents the (signed) volume of the parallelepiped formed by three vectors.

Important concepts from vector and tensor algebra, as well as calculus, are fully covered in Appendix A. Only the basics of set theory is assumed in this work. An introduction to certain groups is required for the immediate chapter after this. A group under multiplication is a non-empty set G along with a map $G \times G \to G$, $(g, h) \mapsto gh$, satisfying the following axioms:

i. (Associativity) For all $a, b, c \in G$, $(ab)c = a(bc)$.

ii. (Existence of identity) There is an element $e \in G$, such that $eg = ge = g$ for every $g \in G$.

iii. (Existence of inverse) There is an element $h \in G$ for every element $g \in G$ with the property $gh = hg = e$.

Then, a subset H of a group G under the same multiplication is a subgroup of G.

The general linear group, $GL(n, \mathbb{R})$, consists of all the non-singular real matrices. $GL^+(n)$ is often written for the group of non-singular real matrices with positive determinant. A subgroup of $GL(n, \mathbb{R})$ formed by orthogonal matrices is unsurprisingly called the orthogonal group, $O(n)$. $SO(n)$ in turn is a subgroup of $O(n)$, where the determinant of the matrices is equal to $+1$. This is known as the rotation group. More comprehensively:

$$\begin{aligned} GL(n, \mathbb{R}) &= \{\mathbf{A} \in \mathbb{R}^{n \times n} \,|\, \det \mathbf{A} \neq 0\}; \\ GL^+(n) &= \{\mathbf{A} \in GL(n, \mathbb{R}) \,|\, \det \mathbf{A} > 0\}; \\ O(n) &= \{\mathbf{A} \in GL(n, \mathbb{R}) \,|\, \mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{I}\}; \\ SO(n) &= \{\mathbf{A} \in O(n) \,|\, \det \mathbf{A} = +1\}. \end{aligned}$$

The set of real symmetric matrices is identified by $S(n)$ and the set of real skew-symmetric matrices by $\mathfrak{so}(n)$. $SPD(n)$ refers to the set of real symmetric, positive definite matrices. More comprehensively:

$$\begin{aligned} S(n) &= \{\mathbf{A} \in \mathbb{R}^{n \times n} \,|\, \mathbf{A} - \mathbf{A}^{\mathrm{T}} = \mathbf{0}\}; \\ \mathfrak{so}(n) &= \{\mathbf{A} \in \mathbb{R}^{n \times n} \,|\, \mathbf{A} + \mathbf{A}^{\mathrm{T}} = \mathbf{0}\}; \\ SPD(n) &= \{\mathbf{A} \in \mathbb{R}^{n \times n} \,|\, \mathbf{A} - \mathbf{A}^{\mathrm{T}} = \mathbf{0}, \mathbf{v} \cdot \mathbf{A}\mathbf{v} > 0 \text{ for any } \mathbf{v} \neq \mathbf{0}\}. \end{aligned}$$

Matrix representation of a tensor is not invoked when dealing with sets or groups, e.g. $\mathbf{R} \in SO(n)$ is to be understood as $[\mathbf{R}] \in SO(n)$.

Throughout this work, bold lower-case Roman letters are reserved for vectors or points without exception. Bold upper-case Roman letters denote tensors or matrices and their non-bold equivalent (without subscripts) represent sets. However, this rule is not adhered to in Chapter 3, where bold lower-case and upper-case Greek letters can also be tensors and vectors if so defined. Also, an exception is made for the Cauchy stress in Chapter 4 for purposes of continuity. Additionally in Chapter 3, non-bold upper-case Roman letters signify scalars. Furthermore, there are a number of non-bold upper-case Roman letters with special meaning: I represents an arbitrary index set; O is used to describe the asymptotic behaviour of functions in Chapter 3 and to classify algorithms according to their performance with respect to processing time in Chapter 4; and T_r is the triangular number. Components of vectors, matrices and tensors are the non-bold counterparts with subscripts. It goes without saying that non-bold lower-case letters, unless otherwise specified, are scalars.

It is worth noting that a domain, in this work, is defined as a bounded and connected open set. A smooth boundary of such a domain is to be understood as piecewise smooth.

Finally, each chapter treats a separate topic and is, therefore, mathematically distinct from the other. However, there is a common thread based on continuum mechanics and linear elasticity.

Chapter 2

Rigid Body Motion Mitigation

2.1 Introduction

Rigid body displacement due to translation is easily negated. First, the nodal point displacements on the surface of a finite element mesh (refer to Section 4.2.3 for a complete definition) are averaged. Then, the resulting value is subtracted from all nodal point displacements of the mesh. Unfortunately, the task of eliminating rigid body displacement due to rotation is not so simple. De Veubeke [14] minimised the norm of relative displacements over the whole body to yield the rotation operator. The optimisation procedure was somewhat complex, and it involved a classical eigenvalue problem. Kaczmarczyk et al. [15] suggest a cost function based on the skew-symmetric part of the displacement gradient. However, the optimisation procedure employed did not take full advantage of the geometric structure of the problem. This is the reason why in many iterative schemes additional computations are required in each step to scale the solution back into the search space [16].

In recent years, geometric optimisation has come to the fore. This is where the geometry of the underlying parameter space is taken into account. Of particular interest are optimisation schemes on Riemannian manifolds, i.e. manifolds with a metric structure. Manifolds look locally like Euclidean space. Therefore, it stands to reason that there is a Riemannian equivalent to a straight line (that is a geodesic), a gradient and a Hessian of an objective function. Smith in [17] provided an accomplished treatment of Newton's method on general Riemannian manifolds. Newton's method specifically on compact Lie groups was also addressed by Mahony [18]. Edelman et al. [19] focused on the application of Newton's method to Stiefel and Grassmann manifolds. This work follows closely that of Ma et al.

in [20], where Newton's method on the essential manifold was employed to tackle the motion and structure problem in computer vision.

In this chapter, a novel and efficient optimisation scheme exploiting the geometric structure of the rotation group is proposed to evaluate the cost function in [15] and yield the best-fit matrix required to mitigate rigid body rotation.

This chapter is organised as follows. Section 2.2 introduces concepts from continuum mechanics, concisely but comprehensively. This serves to furnish the reader with the knowledge required to appreciate the derivation of the cost function in Section 2.3. A review of the geometric structure of the rotation group is offered in Section 2.4. Section 2.5 contains the Riemannian gradient and Riemannian Hessian of the cost function, along with the optimisation procedure. Experiments demonstrating the capability of the developed methodology applied to finite element mesh examples are included in Section 2.6. Section 2.7 concludes the chapter.

2.2 Problem background

Consider a body occupying a domain $\mathcal{B}$ in $\mathbb{R}^3$ with a smooth boundary denoted $\partial\mathcal{B}$. A body can occupy different regions in $\mathbb{R}^3$ but not at the same time. Therefore, there is a need to identify a reference configuration of the body, say $\mathcal{B}_0$. A point $\mathbf{p} \in \mathcal{B}_0$ is known as a material point and corresponds to a particle of the body. This is based on the notion that a continuum body consists of a continuous distribution of an infinite number of particles. Assume that there exists a smooth and invertible mapping function

$$\mathbf{x} = \boldsymbol{\varphi}(\mathbf{p}) \tag{2.2.1}$$

at a fixed time, which assigns every material point to a point $\mathbf{x}$ in the body $\mathcal{B}_1$. This mapping is the vehicle through which deformation of the body $\mathcal{B}_0$ is achieved. Since the mapping is injective, no material points overlap during deformation. The deformed body $\mathcal{B}_1$ is said to be the current configuration. Displacement of a point between the reference and current configurations of the body is the vector

$$\mathbf{u}(\mathbf{p}) = \boldsymbol{\varphi}(\mathbf{p}) - \mathbf{p}. \tag{2.2.2}$$

A fundamental concept in continuum mechanics is the deformation gradient $\mathbf{F}$, which is

defined as

$$\mathbf{F}(\mathbf{p}) = \nabla\boldsymbol{\varphi}(\mathbf{p}). \tag{2.2.3}$$

The determinant of the deformation gradient, $\det \mathbf{F}$, is representative of the post-deformation local volume per unit of original volume. It is reasonable then to assume that $\det \mathbf{F} \neq 0$. Furthermore, in the reference configuration, the deformation gradient is equal to one and it follows that

$$\det \mathbf{F} > 0. \tag{2.2.4}$$

Thus, $\mathbf{F} \in GL^{+}(3)$ and the deformation gradient is invertible. Re-arranging (2.2.2) as such

$$\boldsymbol{\varphi}(\mathbf{p}) = \mathbf{p} + \mathbf{u}, \tag{2.2.5}$$

and taking the gradient leads to another way of expressing the deformation gradient:

$$\mathbf{F} = \mathbf{I} + \nabla\mathbf{u}. \tag{2.2.6}$$

There are two algebraic decompositions that are prominent in the analysis of deformation. The first is multiplicative and is realised through the polar decomposition theorem:

$$\mathbf{F} = \mathbf{RU} = \mathbf{VR}. \tag{2.2.7}$$

Here, $\mathbf{U}$ and $\mathbf{V}$ are symmetric, positive definite tensors known as the right stretch tensor and left stretch tensor, respectively. Consequently, they belong to $SPD(3)$. The local rotation $\mathbf{R}$ features in both decompositions and is a proper orthogonal tensor, that is $\mathbf{R} \in SO(3)$, satisfying

$$\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I} \text{ and } \det \mathbf{R} = 1. \tag{2.2.8}$$

Note that the decompositions are uniquely determined.

According to the polar decomposition theorem, any deformation can be interpreted as two sequential transformations: stretching and rotation (order is irrelevant). If $\mathbf{U} = \mathbf{I}$, deformation is in fact a pure rotation. Similarly, if $\mathbf{R} = \mathbf{I}$, then stretching is the sole contributor to the deformation. Focusing on the right polar decomposition, it can be seen from

$$\mathbf{F}^{\mathrm{T}}\mathbf{F} = \mathbf{U}\mathbf{R}^{\mathrm{T}}\mathbf{R}\mathbf{U} \tag{2.2.9}$$

that

$$\mathbf{U} = \sqrt{\mathbf{F}^{\mathrm{T}}\mathbf{F}}. \tag{2.2.10}$$

This result holds if $\mathbf{F}^{\mathrm{T}}\mathbf{F}$ is symmetric and positive definite. Clearly, $(\mathbf{F}^{\mathrm{T}}\mathbf{F})^{\mathrm{T}} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$. Then, for any vector $\mathbf{v} \neq \mathbf{0}$, the positive-definite condition

$$\mathbf{v} \cdot \mathbf{F}^{\mathrm{T}}\mathbf{F}\mathbf{v} = \mathbf{F}\mathbf{v} \cdot \mathbf{F}\mathbf{v} > 0 \tag{2.2.11}$$

is satisfied. Thus, $\mathbf{F}^{\mathrm{T}}\mathbf{F} \in SPD(3)$. This analysis is not repeated for the left polar decomposition, the same logic leads to $\mathbf{V} = \sqrt{\mathbf{F}\mathbf{F}^{\mathrm{T}}}$. It is not very convenient to work with a measure that involves a square root operation. This is resolved by the right and left Cauchy-Green tensors:

$$\mathbf{C} = \mathbf{U}^2 = \mathbf{F}^{\mathrm{T}}\mathbf{F} \text{ and } \mathbf{B} = \mathbf{V}^2 = \mathbf{F}\mathbf{F}^{\mathrm{T}}. \tag{2.2.12}$$

Based on (2.2.6), the Cauchy-Green tensors can be expressed as

$$\mathbf{C} = \mathbf{I} + \boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}\boldsymbol{\nabla}\mathbf{u} \text{ and} \tag{2.2.13a}$$

$$\mathbf{B} = \mathbf{I} + \boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}} + \boldsymbol{\nabla}\mathbf{u}\boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}. \tag{2.2.13b}$$

If small deformation theory is assumed, the higher order terms can be ignored and

$$\mathbf{C} = \mathbf{B} = \mathbf{I} + 2\mathbf{E}, \tag{2.2.14}$$

in which the infinitesimal strain tensor $\mathbf{E}$ has the form

$$\mathbf{E} = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}). \tag{2.2.15}$$

The second decomposition is additive in nature and relies on the unique representation of a Cartesian second-order tensor as the sum of a symmetric tensor and a skew-symmetric tensor. It follows that

$$\mathbf{F} = (\mathbf{I} + \mathbf{E}) + \mathbf{W}. \tag{2.2.16}$$

Observe that $\mathbf{E} \in S(3)$ is the infinitesimal strain tensor. It can be thus concluded that the additive decomposition of the deformation gradient is only relevant for small deformations. The skew-symmetric tensor, $\mathbf{W} \in \mathfrak{so}(3)$, is called the infinitesimal rotation tensor with the form

$$\mathbf{W} = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} - \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}). \tag{2.2.17}$$

If $\mathbf{W} = \mathbf{0}$, deformation is due to pure strain. Similarly, if $\mathbf{E} = \mathbf{0}$, there is only rotation.

Finally, for any two points $\mathbf{q}, \mathbf{r} \in \mathcal{B}_0$, rigid deformation can be defined as

$$\|\boldsymbol{\varphi}(\mathbf{q}) - \boldsymbol{\varphi}(\mathbf{r})\| = \|\mathbf{q} - \mathbf{r}\|. \tag{2.2.18}$$

This states that the distance between any two points in the reference configuration $\mathcal{B}_0$ must be preserved in the current configuration $\mathcal{B}_1$.

2.3 Cost function

A function is now sought for minimisation purposes to yield the rotation $\mathbf{R}$ responsible for the rigid deformation of a body. Since only small deformations are of concern, the additive decomposition of the deformation gradient is most apt here. It has already been established that the infinitesimal rotation tensor must be absent for there to be a pure strain deformation. An assumption is, therefore, made that the condition

$$\frac{1}{2}\int_{\mathcal{B}_0}(\boldsymbol{\nabla}\mathbf{u}(\mathbf{p})-\boldsymbol{\nabla}\mathbf{u}(\mathbf{p})^{\mathrm{T}})\,dy_0=\mathbf{0} \tag{2.3.1}$$

is sufficient.

It is possible to express displacement with respect to the current configuration as

$$\mathbf{u}(\mathbf{x})=\mathbf{x}-\mathbf{p}(\mathbf{x}) \tag{2.3.2}$$

at a fixed time. Clearly,

$$\mathbf{u}(\mathbf{x})=\mathbf{u}(\mathbf{p}). \tag{2.3.3}$$

In small deformation theory, it can be assumed that

$$\boldsymbol{\nabla}\mathbf{u}(\mathbf{x})=\boldsymbol{\nabla}\mathbf{u}(\mathbf{p}). \tag{2.3.4}$$

Based on the infinitesimal rotation tensor taken with respect to the current configuration,

$$\frac{1}{2}\int_{\mathcal{B}_1}(\boldsymbol{\nabla}\mathbf{u}(\mathbf{x})-\boldsymbol{\nabla}\mathbf{u}(\mathbf{x})^{\mathrm{T}})\,dy_1, \tag{2.3.5}$$

a pseudo-vector

$$\mathbf{b}=\frac{1}{2}\int_{\mathcal{B}_1}\boldsymbol{\nabla}\times\mathbf{u}\,dy_1 \tag{2.3.6}$$

can be defined (refer to Appendix A). Applying the divergence theorem of Gauss to obtain the form

$$\mathbf{b}=-\frac{1}{2}\int_{\partial\mathcal{B}_1}\mathbf{u}\times\mathbf{m}\,ds_1, \tag{2.3.7}$$

where $\mathbf{m}$ is the outward unit normal vector field to $\partial\mathcal{B}_1$.

For any deformation, let

$$\mathbf{u}=(\mathbf{x}-\mathbf{p})-(\mathbf{R}\mathbf{p}-\mathbf{p})=\mathbf{x}-\mathbf{R}\mathbf{p} \tag{2.3.8}$$

and

$$\mathbf{b} = -\frac{1}{2}\int_{\partial\mathcal{B}_1} \mathbf{x}\times\mathbf{m}\,ds_1 + \frac{1}{2}\int_{\partial\mathcal{B}_1} \mathbf{R}\mathbf{p}\times\mathbf{m}\,ds_1. \tag{2.3.9}$$

Consider the case of a body subject to a pure rotation. An infinitesimal surface element in the current configuration is related to an infinitesimal surface element in the reference configuration by

$$\mathbf{m}\,ds_1 = \mathbf{R}\mathbf{n}\,ds_0, \tag{2.3.10}$$

in which $\mathbf{n}$ is the outward unit normal vector field to $\partial\mathcal{B}_0$. It is then possible to write

$$\mathbf{b} = -\frac{1}{2}\int_{\partial\mathcal{B}_1} \mathbf{x}\times\mathbf{R}\mathbf{n}\,ds_1 + \frac{1}{2}\int_{\partial\mathcal{B}_0} \mathbf{R}\mathbf{p}\times\mathbf{R}\mathbf{n}\,ds_0, \tag{2.3.11}$$

as dealing only with small deformations. Taking advantage of the identity $\mathbf{R}(\mathbf{a}\times\mathbf{b}) = \mathbf{R}\mathbf{a}\times\mathbf{R}\mathbf{b}$ for vectors $\mathbf{a},\mathbf{b}\in\mathbb{R}^3$, see Appendix B for the proof, the second-term on the right-hand side disappears as

$$\frac{1}{2}\int_{\mathcal{B}_0} (\boldsymbol{\nabla}\mathbf{p} - \boldsymbol{\nabla}\mathbf{p}^{\mathrm{T}})\,dy_0 = \mathbf{0}. \tag{2.3.12}$$

Thus, (2.3.11) reduces to

$$\mathbf{b} = \int_{\partial\mathcal{B}_1} \mathbf{R}\mathbf{n}\times\mathbf{x}\,ds_1 \tag{2.3.13}$$

with the fraction dropped. Further manipulation leads to

$$\begin{aligned}\mathbf{b} &= \int_{\partial\mathcal{B}_1} \mathbf{R}\mathbf{R}^{\mathrm{T}}(\mathbf{R}\mathbf{n}\times\mathbf{x})\,ds_1\\ &= \int_{\partial\mathcal{B}_1} \mathbf{R}(\mathbf{n}\times\mathbf{R}^{\mathrm{T}}\mathbf{x})\,ds_1.\end{aligned} \tag{2.3.14}$$

Given that $\mathbf{R}$ is constant over the body,

$$\mathbf{b} = \mathbf{R}\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times\mathbf{R}^{\mathrm{T}}\mathbf{x}\,ds_1. \tag{2.3.15}$$

A cost function can now be defined as [15]

$$a(\mathbf{R}) = \frac{1}{2}[\mathbf{c}]^{\mathrm{T}}[\mathbf{c}], \tag{2.3.16}$$

where

$$\mathbf{c} = \int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times\mathbf{R}^{\mathrm{T}}\mathbf{x}\,ds_1. \tag{2.3.17}$$

Minimisation of the cost function returns the best-fit rotation. It is worth noting that the non-rigid deformation is akin to noise in the optimisation.

2.4 Geometry of $SO(3)$

In addition to being a group, $SO(3)$ is also a C^∞ (smooth) manifold. Consequently, the multiplication, $G \times G \to G$, and inverse, $G \to G$, group operations are C^∞. This is the very definition of a Lie group. A manifold is a set M together with a maximal atlas $\mathcal{A}^+$, i.e. $\mathcal{M} = (M, \mathcal{A}^+)$. Basically, a C^∞ atlas is a finite collection of open sets P_i covering the set M, that is $M = \cup_{i \in I} P_i$. There is a homeomorphism ϕ_i from P_i into an open subset of $\mathbb{R}^d$. Each pair (P_i, ϕ_i) forms an individual chart. If $P_i \cap P_j \neq \emptyset$, then $\phi_i \circ \phi_j^{-1} : \phi_j(P_i \cap P_j) \to \phi_i(P_i \cap P_j)$ is C^∞. An atlas is maximal when there is no larger atlas containing M. It is worth noting that a maximal atlas is called a differential structure on M.

Since $SO(3)$ is a C^∞ manifold, there is a tangent space $T_{\mathbf{R}}(SO(3))$ encompassing all tangent vectors at every $\mathbf{R}$. Of particular interest is the tangent space at the identity element, which forms the corresponding Lie algebra:

$$T_e(SO(3)) = \mathfrak{so}(3). \tag{2.4.1}$$

It is clear that the Lie algebra consists of all 3×3 skew-symmetric matrices. $SO(3)$ is endowed with a Riemannian metric, which is the inner product

$$g(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{2}\mathrm{tr}\,(\mathbf{S}_1^{\mathrm{T}}\mathbf{S}_2), \tag{2.4.2}$$

where $\mathbf{S}_1, \mathbf{S}_2 \in T_{\mathbf{R}}(SO(3))$. This introduces the notion of length in specific relation to tangent vectors. Based on this, the length of a curve on the manifold can be determined. A geodesic is the curve of shortest length between $\mathbf{R}_1 \in SO(3)$ and $\mathbf{R}_2 \in SO(3)$ with $\mathbf{R}_1 \neq \mathbf{R}_2$. Furthermore, a push-forward map yields the tangent space at any $\mathbf{R} \in SO(3)$:

$$T_{\mathbf{R}}(SO(3)) = \{\mathbf{RS} \,|\, \mathbf{S} \in \mathfrak{so}(3)\}. \tag{2.4.3}$$

There exists a well-defined and surjective exponential map [21] that connects the Lie algebra to the Lie group through

$$\exp([\mathbf{v}]_\times \theta) = \mathbf{I} + [\mathbf{v}]_\times \sin\theta + [\mathbf{v}]_\times^2(1 - \cos\theta), \tag{2.4.4}$$

where $[\mathbf{v}]_\times \in \mathfrak{so}(3)$, $\|\mathbf{v}\| = 1$ and $\theta \in \mathbb{R}$. This is known as Rodrigues' formula. Geometrically, this can be interpreted as a circular movement around an axis $\mathbf{v}$ by an amount θ. A geodesic on $SO(3)$ at $\mathbf{R}$ in the direction $\mathbf{K} \in T_{\mathbf{R}}(SO(3))$ is written as

$$\begin{aligned} \mathbf{R}(t) &= \exp(\mathbf{R}, \mathbf{K}t) \\ &= \mathbf{R}\exp(\mathbf{L}t) \\ &= \mathbf{R}(\mathbf{I} + \mathbf{L}\sin t + \mathbf{L}^2(1 - \cos t)), \end{aligned} \tag{2.4.5}$$

in which $\mathbf{L} = \mathbf{R}^{\mathrm{T}}\mathbf{K} \in \mathfrak{so}(3)$ and $t \in \mathbb{R}$. Observe that at $t = 0$, $\mathbf{R}(0) = \mathbf{R}$. Then, the first and second derivatives at $t = 0$ are

$$\begin{aligned}\left.\frac{d\mathbf{R}(t)}{dt}\right|_{t=0} &= \left.(\mathbf{RL}\cos t + \mathbf{RL}^2 \sin t)\right|_{t=0} \\ &= \mathbf{RL} \\ &= \mathbf{R}(\mathbf{R}^{\mathrm{T}}\mathbf{K}) \\ &= \mathbf{K}\end{aligned} \tag{2.4.6}$$

and

$$\begin{aligned}\left.\frac{d^2\mathbf{R}(t)}{dt^2}\right|_{t=0} &= \left.(-\mathbf{RL}\sin t + \mathbf{RL}^2 \cos t)\right|_{t=0} \\ &= \mathbf{RL}^2 \\ &= \mathbf{K}(\mathbf{R}^{\mathrm{T}}\mathbf{K}) \\ &= -\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{R},\end{aligned} \tag{2.4.7}$$

respectively. The final line in the second derivative follows from the fact that $\mathbf{L}$ is skew-symmetric.

2.5 Optimisation on the manifold $SO(3)$

Consider the unconstrained minimisation problem of a function $f(z)$, where z belongs to $\mathbb{R}$. Newton's method constructs a sequence of iterates beginning with an initial guess z^0 and ending with a value z^* that is a local minimiser, i.e. $f'(z^*) = 0$. This approach is predicated on the function f being twice-continuously differentiable. Each iteration is of the form

$$z^{k+1} = z^k + h^k. \tag{2.5.1}$$

The update h^k to the solution is the Newton direction. It can be found by minimising the quadratic approximation of f in the neighbourhood of z^k. The first three terms of the Taylor series are

$$f(z^k + h) \approx f(z^k) + f'(z^k)h + \frac{1}{2}f''(z^k)h^2, \tag{2.5.2}$$

where $h \in \mathbb{R}$. Then, setting the gradient to zero,

$$\frac{d}{dh}\left(f(z^k) + f'(z^k)h + \frac{1}{2}f''(z^k)h^2\right) = f'(z^k) + f''(z^k)h = 0, \tag{2.5.3}$$

provides

$$h = -\frac{f'(z^k)}{f''(z^k)}. \tag{2.5.4}$$

Clearly, if $f(z)$ is a quadratic function, convergence is achieved in a single step.

Newton's method can be generalised to optimisation problems on the manifold $SO(3)$. The first and second derivative of the cost function in the direction $\mathbf{K} \in T_{\mathbf{R}}(SO(3))$ evaluated at $t = 0$ are

$$\begin{aligned} da(\mathbf{K}) &= \frac{d}{dt} a(\mathbf{R}(t))\Big|_{t=0} \\ &= \left([\mathbf{c}]^{\mathrm{T}}[\dot{\mathbf{c}}]\right)\Big|_{t=0} \\ &= \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \dot{\mathbf{R}}^{\mathrm{T}}\mathbf{x}\, ds_1\right] \\ &= \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{K}^{\mathrm{T}}\mathbf{x}\, ds_1\right] \end{aligned} \tag{2.5.5}$$

and

$$\begin{aligned} \operatorname{Hess} a(\mathbf{K},\mathbf{K}) &= \frac{d^2}{dt^2} a(\mathbf{R}(t))\Big|_{t=0} \\ &= \left([\dot{\mathbf{c}}]^{\mathrm{T}}[\dot{\mathbf{c}}] + [\mathbf{c}]^{\mathrm{T}}[\ddot{\mathbf{c}}]\right)\Big|_{t=0} \\ &= \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \dot{\mathbf{R}}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \dot{\mathbf{R}}^{\mathrm{T}}\mathbf{x}\, ds_1\right] \\ &\quad + \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \ddot{\mathbf{R}}^{\mathrm{T}}\mathbf{x}\, ds_1\right] \\ &= \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{K}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{K}^{\mathrm{T}}\mathbf{x}\, ds_1\right] \\ &\quad - \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{K}\mathbf{K}^{\mathrm{T}}\mathbf{x}\, ds_1\right], \end{aligned} \tag{2.5.6}$$

respectively. A polarisation of $\operatorname{Hess} a(\mathbf{K},\mathbf{K})$ is required as in [20] to aid the computation. For $\mathbf{Y},\mathbf{Z} \in T_{\mathbf{R}}SO(3)$, this equates to

$$\begin{aligned} \operatorname{Hess} a(\mathbf{Y},\mathbf{Z}) &= \frac{1}{4}\left[\operatorname{Hess} a(\mathbf{Y}+\mathbf{Z},\mathbf{Y}+\mathbf{Z}) - \operatorname{Hess} a(\mathbf{Y}-\mathbf{Z},\mathbf{Y}-\mathbf{Z})\right] \\ &= \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{Y}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{Z}^{\mathrm{T}}\mathbf{x}\, ds_1\right] \\ &\quad -\frac{1}{2}\left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{Z}\mathbf{Y}^{\mathrm{T}}\mathbf{x}\, ds_1\right] \\ &\quad -\frac{1}{2}\left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{x}\, ds_1\right]^{\mathrm{T}} \left[\int_{\partial\mathcal{B}_1} [\mathbf{n}]_\times \mathbf{R}^{\mathrm{T}}\mathbf{Y}\mathbf{Z}^{\mathrm{T}}\mathbf{x}\, ds_1\right]. \end{aligned} \tag{2.5.7}$$

Note the incredibly simple form of the derivatives. There is no need for approximation or even modification of the second-derivative in this context.

Selecting $\mathbf{J}^i = \mathbf{R}[\mathbf{e}_i]_\times$ for $i = 1, 2, 3$ as the basis. Then, the gradient vector $\mathbf{g}$ and Hessian matrix $\mathbf{H}$ can be defined as

$$g_k = da(\mathbf{J}^k) \tag{2.5.8}$$

and

$$H_{kl} = \text{Hess}\, a(\mathbf{J}^k, \mathbf{J}^l) \tag{2.5.9}$$

for $k = l = 1, 2, 3$. Assuming a non-degenerate and invertible Hessian, it is possible to solve for the vector $\mathbf{s} \in \mathbb{R}^3$:

$$\mathbf{s} = -\mathbf{H}^{-1}\mathbf{g}. \tag{2.5.10}$$

Thus, the Newton direction is

$$\mathbf{K} = \mathbf{R}[\mathbf{s}]_\times. \tag{2.5.11}$$

Once the Newton direction is available in each iteration, an update of the rotation along the geodesic in the direction $\mathbf{K}$ is performed. Convergence is determined by the Newton decrement

$$\lambda := \sqrt{[\mathbf{g}]^{\mathrm{T}}[\mathbf{H}^{-1}][\mathbf{g}]} \tag{2.5.12}$$

meeting the requirement $\lambda \leq \gamma$, where $\gamma > 0$ is a user-set error tolerance. Smith in [17] showed that Newton's method on Riemannian manifolds has a local quadratic rate of convergence. A summary of Newton method's as formulated here follows:

Step 0. Initialise $\mathbf{R}$

Step 1. (Newton direction)

- Compute basis $\mathbf{J}^i = \mathbf{R}[\mathbf{e}_i]_\times$ for $i = 1, 2, 3$
- Compute gradient $(\mathbf{g})_k = da(\mathbf{J}^k)$
- Compute Hessian matrix $(\mathbf{H})_{kl} = \text{Hess}\, a(\mathbf{J}^k, \mathbf{J}^l)$
- Solve for $\mathbf{s} = -\mathbf{H}^{-1}\mathbf{g}$

- Determine $\mathbf{K} = \mathbf{R}[\mathbf{s}]_\times$

Step 2. (Rotation update)

- Calculate $t = \sqrt{\frac{1}{2}\text{tr}\,(\mathbf{K}^\text{T}\mathbf{K})}$ and $\mathbf{L} = \mathbf{R}^\text{T}\mathbf{K}/t$
- Compute $\mathbf{R}(t) = \mathbf{R}(\mathbf{I} + \mathbf{L}\sin t + \mathbf{L}^2(1 - \cos t))$

Step 3. If $\lambda > \gamma$, return to Step 1.

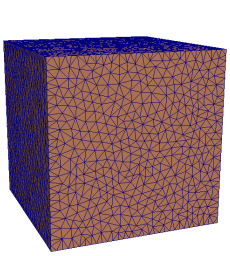
2.6 Validation

In order to demonstrate the effectiveness of the proposed methodology in a finite element context, a series of tests on geometrical shapes ranging from the simple to the complex were performed. Since only the surface of a finite element mesh is required for the optimisation, the input file examples (representing a cube, a quarter of a gear, and a brain; see Figure 2.1) available from [22] served the purpose. Each example consists of a file containing nodal point coordinates and another with the connectivity information related to triangles forming the surface of an assumed finite element mesh.

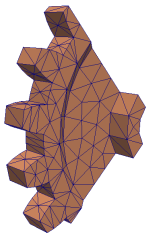
A MATLAB [23] function was written to read the input files. Then, deformation was simulated through the operation $\mathbf{F}\mathbf{p}^{(i)}$, where $\mathbf{p}^{(i)}$ represents the position of a nodal point. The form of the deformation gradient in (**2.2.7**) was used, in which $\mathbf{R}$ was supplied by (**2.4.4**). In the case of pure rotation, $\mathbf{U} = \mathbf{I}$. For non-rigid deformation, $\mathbf{U}$ was selected so that a symmetric deformation gradient (i.e. no rotation) resulted given $\mathbf{R} = \mathbf{I}$. Accordingly, pure stretch and pure shear were studied having the forms

$$[\mathbf{U}] = \begin{bmatrix} 1+b & 0 & 0 \\ 0 & 1-b & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } [\mathbf{U}] = \begin{bmatrix} 1 & c & 0 \\ c & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.6.1}$$

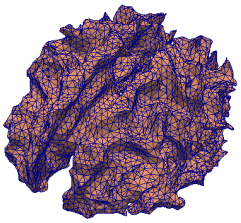
where b and c are constants. Figure 2.2 illustrates pure rotation, pure stretch and pure shear of the cube model. Refer to Section 4.2.3 for information on how the outward unit normal vector in the reference configuration was determined. The integral of $\mathbf{x}$ over each finite element in the current configuration was taken as the average of the corresponding positions of the nodes. Newton's method as formulated in the previous section, also implemented in MATLAB, was applied to all test cases.



(a) 5,212 triangles

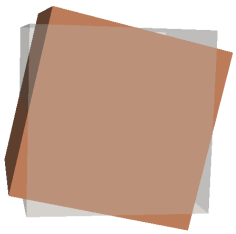


(b) 430 triangles

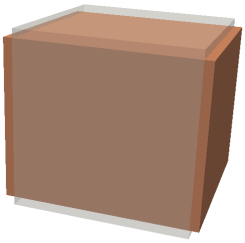


(c) 21,026 triangles

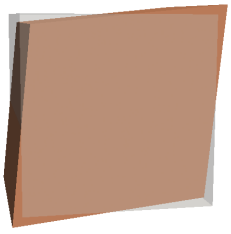
Figure 2.1. Test models: (a) cube; (b) quarter part of a gear; and (c) brain.



(a)



(b)



(c)

Figure 2.2. Example of a cube subject to (a) pure rotation with $\mathbf{v} = (0, 0, 1)^{\mathrm{T}}$ and $\theta = 0.2094$, (b) pure stretch with $b = 0.1$, and (c) pure shear with $c = 0.1$.

Introducing the Frobenius norm

$$\|\mathbf{A}\|_{\mathrm{F}} = \left(\sum_{i,j} |A_{ij}|^2 \right)^{\frac{1}{2}} \tag{2.6.2}$$

for a matrix $\mathbf{A}$. Then, the mean error can be defined as the Frobenius norm of the difference between the estimated and actual rotation. The rate of local convergence for the proposed Newton algorithm to a fixed point $\mathbf{R}^*$ is shown in Figure 2.3. This was the average of one hundred pure rotation tests carried out on the cube model, where $\mathbf{v}$ was selected randomly, as was $\theta \in [-0.2793, 0.2793]$ in each test. Initialisation was accomplished using $\mathbf{v} = (0.57735, 0.57735, 0.57735)^{\mathrm{T}}$ and $\theta = 0.017321$ for all tests. Observe that convergence was achieved in four iterations. Tests on the brain and quarter of a gear models produced very similar results. Clearly, the closer the initial rotation to the actual one, the better the rate of convergence. In general, the initialisation used or $\mathbf{R} = \mathbf{I}$ represent sensible starting points. Both have comparable rates of convergence.

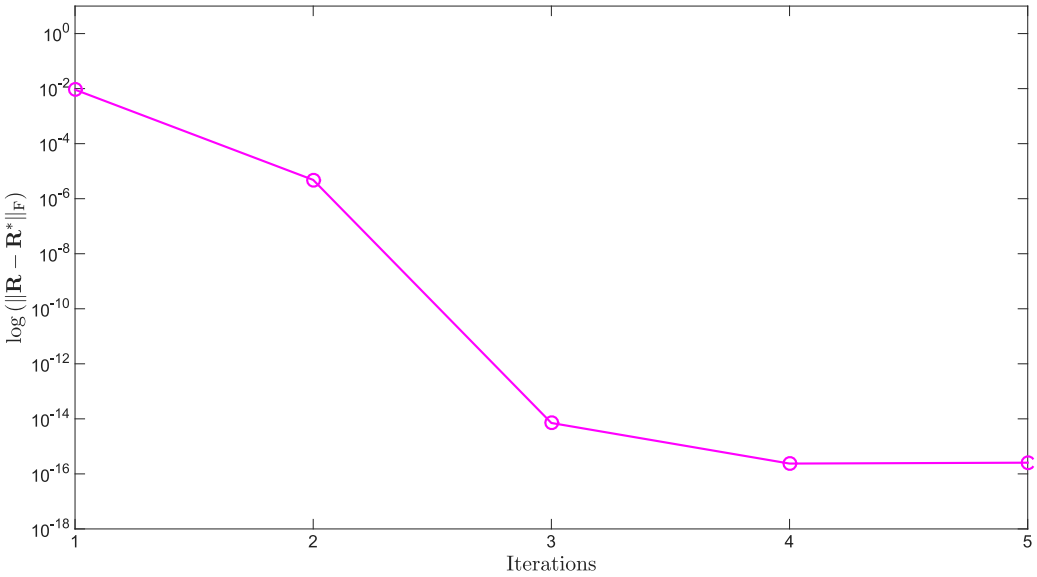
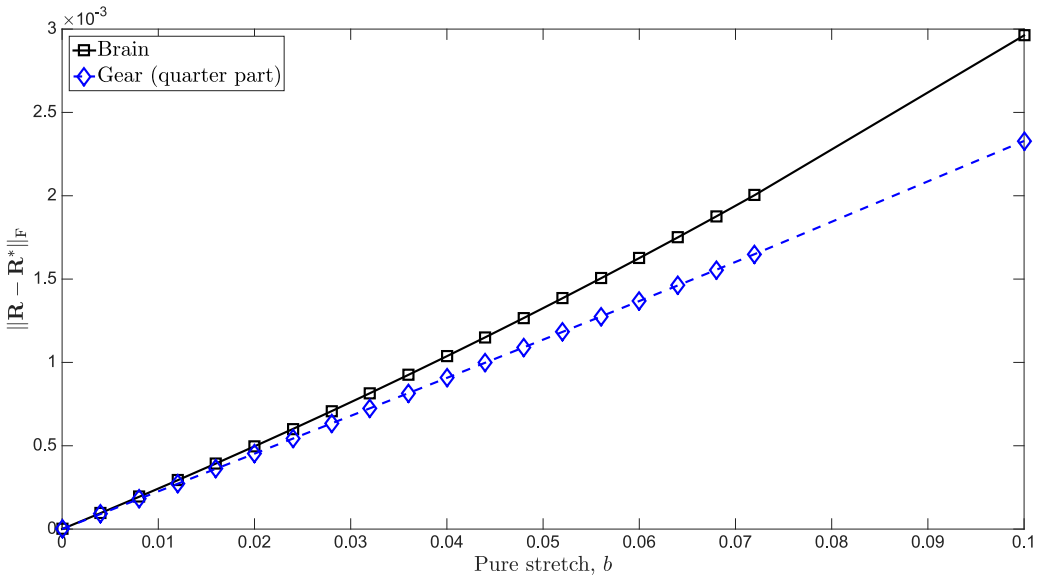


Figure 2.3. Rate of local convergence.

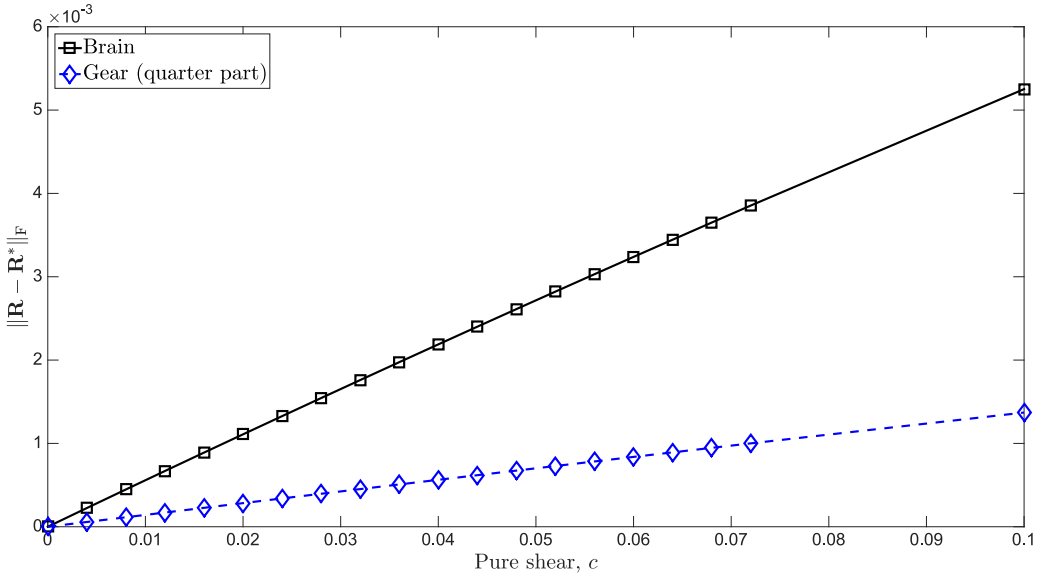
For non-rigid deformation, the mean error associated with the pure stretching or pure shear of the brain and quarter of a gear models can be seen in Figure 2.4a and Figure 2.4b, respectively. Two thousand tests were carried out to establish the impact of rotation on the mean error within the range of $[-0.2793, 0.2793]$ for θ and for random $\mathbf{v}$. It was concluded that the mean error is unaffected by the rotation. Additionally, tests on the cube model did not yield an error. There appears to be a correlation between the complexity of the geometry and mean error. However, further tests are required to confirm this. On average, convergence was achieved in the test cases within three or four iterations (for $\gamma = 1.0\mathrm{e} - 8$). Initialisation was as before in the pure rotation tests.

2.7 Conclusions

A cost function based on the skew-symmetric part of the displacement gradient was derived for the purpose of minimisation. Newton's method formulated on the manifold was put forward as a strategy to obtain the rotation in order to mitigate rigid body displacement due to rotation. This approach accounts for the geometry of the underlying parameter space to produce an efficient optimisation procedure. A mathematically elegant scheme results with very simple expressions for the gradient and Hessian. It would be interesting to investigate the non-Riemannian approach proposed in [16] for comparison purposes. This work has potential not only in a finite element context but also in object tracking and computer vision.



(a)



(b)

Figure 2.4. Mean error for (a) pure stretch and (b) pure shear.

Chapter 3

Crack Initiation: A Non-Local Energy Approach

3.1 Introduction

Currently, as it stands, there is no thermodynamically justified energy-based fracture condition that is geometry insensitive. In other words, there is no criterion available that deals with sharp or blunt notches in the correct physical manner. This chapter is, therefore, concerned with the development of a crack initiation criterion applicable to isotropic linear elastic media that is in a brittle state containing sharp and blunt notches, which yields the usual information: (i) fracture load; (ii) crack location; and (iii) propagation direction. Crack nucleation in the criterion is modelled as a sudden and discrete rupture event, i.e. a finite-length crack appearing abruptly, at the macroscopic level. The treatment of crack propagation as a discrete process is attributed to Novozhilov [24].

Sokołowski and Żochowski in [25] introduced the topological derivative to provide the sensitivity of an arbitrary shape functional subject to an infinitesimal domain perturbation. This concept was duly applied by Van Goethem and Novotny in their study of crack nucleation [26]. Allaire et al. in [27] developed a damage model where the notion of the topological derivative was used to determine whether to nucleate damage in a healthy domain. The resulting criterion possesses attractive features, namely the simplicity of the analytical expression and ease of implementation in a numerical framework. However, the criterion is not appropriate in fracture mechanics. In this work, the topological derivative is part of an asymptotic expansion that is employed in the approximation of the total potential

energy change associated with finite crack extension.

Fracture is to be understood as the total separation of an initially intact body. The criterion is designed to operate within the bounds of Linear Elastic Fracture Mechanics (LEFM). Consequently, the stress field surrounding a notch tip is governed by linear elasticity theory and local nonlinear or dissipative behaviour is considered negligible.

This chapter is organised as follows. Section 3.2 furnishes the reader with the requisite mathematics and fracture mechanics knowledge. In Section 3.3, the topological derivative pertaining to the total potential energy functional is derived from first principles. This is followed with a review of common fracture criteria in Section 3.4. Section 3.5 presents the theoretical underpinnings of the proposed criterion. In Section 3.6, predictions made by the criterion are scrutinised. Finally, concluding remarks are offered in Section 3.7.

3.2 Problem background

3.2.1 Linear elasticity

Let $\Omega \subset \mathbb{R}^2$ be a domain with a smooth boundary $\partial\Omega$, which is occupied by a homogeneous and linear elastic body. The constitutive equation relating stress $\boldsymbol{\sigma}$ and strain $\boldsymbol{\varepsilon}$ reads

$$\boldsymbol{\sigma}(\mathbf{u}) = \boldsymbol{C}\boldsymbol{\varepsilon}(\mathbf{u}), \tag{3.2.1}$$

in which $\boldsymbol{C}$ is the symmetric and positive-definite elasticity tensor. Strain is a function of the displacement $\mathbf{u}$ and here takes the infinitesimal form

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}). \tag{3.2.2}$$

For an isotropic material, the elasticity tensor $\boldsymbol{C}$ can be written in terms of the Lamé constants, λ and μ, as

$$\boldsymbol{C} = \lambda(\mathbf{I} \otimes \mathbf{I}) + 2\mu\mathbf{II}. \tag{3.2.3}$$

The Lamé constants are related to Young's modulus, E, and Poisson's ratio, ν, by

$$\mu = \frac{E}{2(1+\nu)} \text{ and } \lambda = \begin{cases} \frac{\nu E}{(1+\nu)(1-2\nu)} & \text{for plane strain.} \\ \frac{\nu E}{1-\nu^2} & \text{for plane stress.} \end{cases} \tag{3.2.4}$$

The boundary $\partial\Omega$ is comprised of two disjoint parts, $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$, where the displacement $\mathbf{h}$ is specified on Γ_D and the traction $\mathbf{g}$ on Γ_N.

The boundary value problem of elastostatics can be stated as follows: find the displacement field $\mathbf{u}$ that satisfies

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}) = \mathbf{0} & \text{in } \Omega, \\ \mathbf{u} = \mathbf{h} & \text{on } \Gamma_D, \\ \boldsymbol{\sigma}(\mathbf{u})\mathbf{n} = \mathbf{g} & \text{on } \Gamma_N, \end{cases} \tag{3.2.5}$$

where $\mathbf{n}$ is the outward unit normal vector to $\partial\Omega$. A minimum total potential energy problem, i.e. $\min(\Pi(\mathbf{u}))$, can be posed where the functional

$$\Pi(\mathbf{u}) = \frac{1}{2}\int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u})\, dy - \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{u}\, ds \tag{3.2.6}$$

is minimised by the admissible displacement field. The first term provides the strain energy stored in the body, while the second term accounts for the work done by surface forces.

3.2.2 Perturbed problem

Consider the case where the domain Ω is perturbed by a small hole that is introduced at an arbitrary point $\mathbf{x}_0 \in \Omega$. In order to keep the discussion general for the time being, the hole can assume any shape with size $\xi > 0$ as described by $\omega_\xi = \mathbf{x}_0 + \xi\omega$, where $\omega \subset \mathbb{R}^2$ is a small domain with a smooth boundary $\partial\omega$. It is important that the size of the hole is sufficiently small, such that $\overline{\omega_\xi} \subset \Omega$. The perturbed domain is then given by $\Omega_\xi = \Omega \backslash \overline{\omega_\xi}$ with the boundary $\partial\Omega_\xi = \partial\Omega \cup \partial\omega_\xi$ (see Figure 3.1).

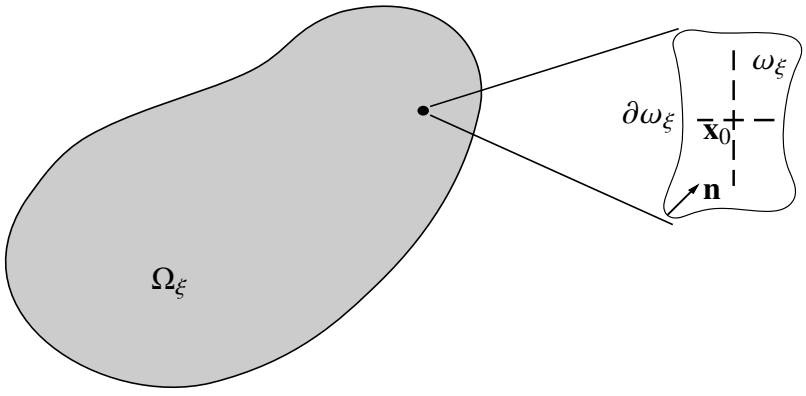


Figure 3.1. Perturbed domain.

The boundary value problem of elastostatics is re-stated for a perturbed domain with an additional homogeneous Neumann condition imposed on $\partial\omega_\xi$: find the displacement field

$\mathbf{u}^\xi$ that satisfies

$$\begin{cases} \operatorname{div} \boldsymbol{\sigma}(\mathbf{u}^\xi) = \mathbf{0} & \text{in } \Omega_\xi, \\ \mathbf{u}^\xi = \mathbf{h} & \text{on } \Gamma_D, \\ \boldsymbol{\sigma}(\mathbf{u}^\xi)\mathbf{n} = \mathbf{g} & \text{on } \Gamma_N, \\ \boldsymbol{\sigma}(\mathbf{u}^\xi)\mathbf{n} = \mathbf{0} & \text{on } \partial\omega_\xi, \end{cases} \tag{3.2.7}$$

where $\mathbf{n}$ is the outward unit normal vector to $\partial\Omega_\xi$. The functional in the minimum total potential energy problem is now written as

$$\Pi(\mathbf{u}^\xi) = \frac{1}{2}\int_{\Omega_\xi} \boldsymbol{\sigma}(\mathbf{u}^\xi) : \boldsymbol{\varepsilon}(\mathbf{u}^\xi)\, dy - \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{u}^\xi\, ds. \tag{3.2.8}$$

3.2.3 V-notch tip stress distribution

Moving on to study stress in an unperturbed domain with a non-smooth boundary. Notches act as stress concentration sites and, therefore, have a strong influence on the load-bearing capacity of a structure. Accordingly, knowledge of the stress field surrounding a V-notch tip in particular is fundamental to fracture mechanics and criterion development.

A V-notch contained in an isotropic linear elastic plate with a wedge angle of 2β and a polar coordinate system (r, θ) centred at the tip is depicted in Figure 3.2. Note that a crack is a specific instance of a V-notch with a wedge angle of $2\beta = 0°$.

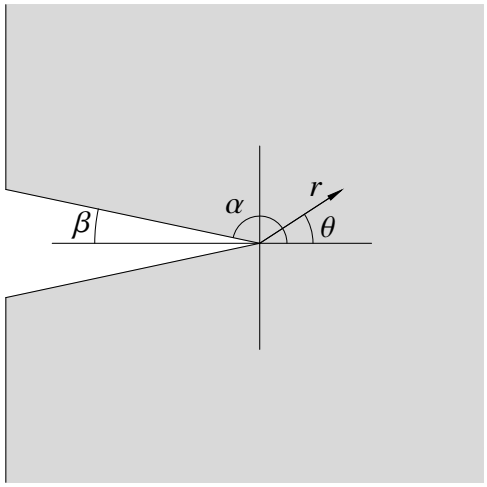


Figure 3.2. Polar coordinate system centred at the tip of a sharp notch.

It is appropriate, at this point, to distinguish three modes related to notch surface deformation:

- *Mode-I*. Normal separation.

- *Mode-II*. Shear sliding normal to the notch front.
- *Mode-III*. Shear sliding parallel to the notch front.

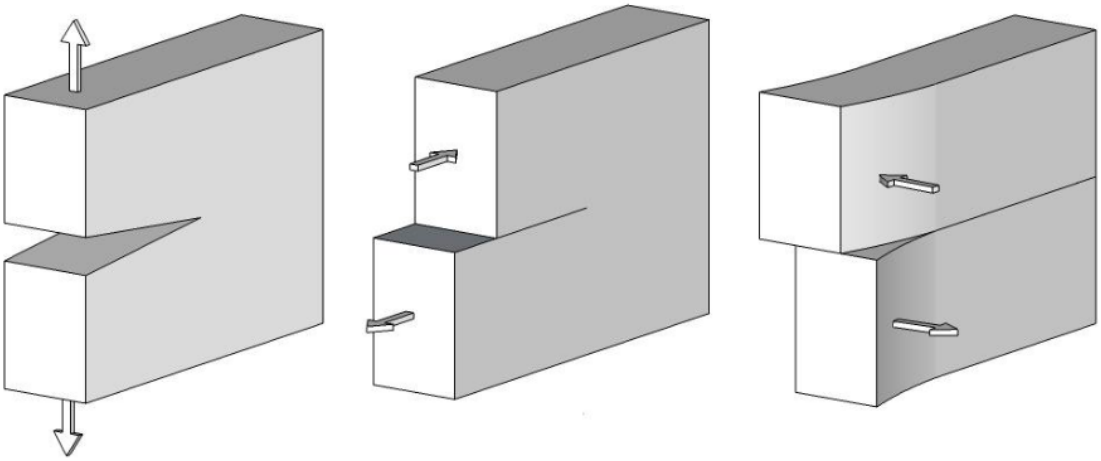


Figure 3.3. Modes of deformation (from left to right): (i) Mode-I; (ii) Mode-II; and (iii) Mode-III.

Any deformation can be obtained from the superposition of these modes. The deformation modes are illustrated pictorially in Figure 3.3. Only Mode-I and Mode-II are of concern in this work.

The stress distribution in the neighbourhood of a V-notch tip with traction-free boundary conditions, $\sigma_{\theta\theta} = \sigma_{r\theta} = 0$ at $\theta = \pm\alpha$, is given in indicial notation ($ij = rr, \theta\theta, r\theta$) by

$$\sigma_{ij} = \frac{K_I^\eta}{r^{1-\eta_I}} m_{ij}^I(\theta) + \frac{K_{II}^\eta}{r^{1-\eta_{II}}} m_{ij}^{II}(\theta). \tag{3.2.9}$$

Each individual term in the equation corresponds to the stress contribution of a deformation mode. Components of the functions $m_{ij}^I(\theta)$ and $m_{ij}^{II}(\theta)$ are documented in Appendix C. The amplitude of the stress field is characterised by the generalised stress intensity factors, K_I^η for Mode-I and K_{II}^η for Mode-II, which are defined as

$$K_I^\eta = \lim_{r\to 0} \left[\sqrt{2\pi} r^{1-\eta_I} \sigma_{\theta\theta}(r, 0)\right] \text{ and} \tag{3.2.10a}$$

$$K_{II}^\eta = \lim_{r\to 0} \left[\sqrt{2\pi} r^{1-\eta_{II}} \sigma_{r\theta}(r, 0)\right]. \tag{3.2.10b}$$

Mode-I and Mode-II exponents, η_I and η_{II}, are the real roots of the following equations:

$$\sin 2\eta_I\alpha + \eta_I \sin 2\alpha = 0 \text{ and} \tag{3.2.11a}$$

$$\sin 2\eta_{II}\alpha - \eta_{II} \sin 2\alpha = 0, \tag{3.2.11b}$$

where $\alpha = \pi - \beta$. The variation of exponents η_I and η_{II} with the wedge angle is presented in Figure 3.4.

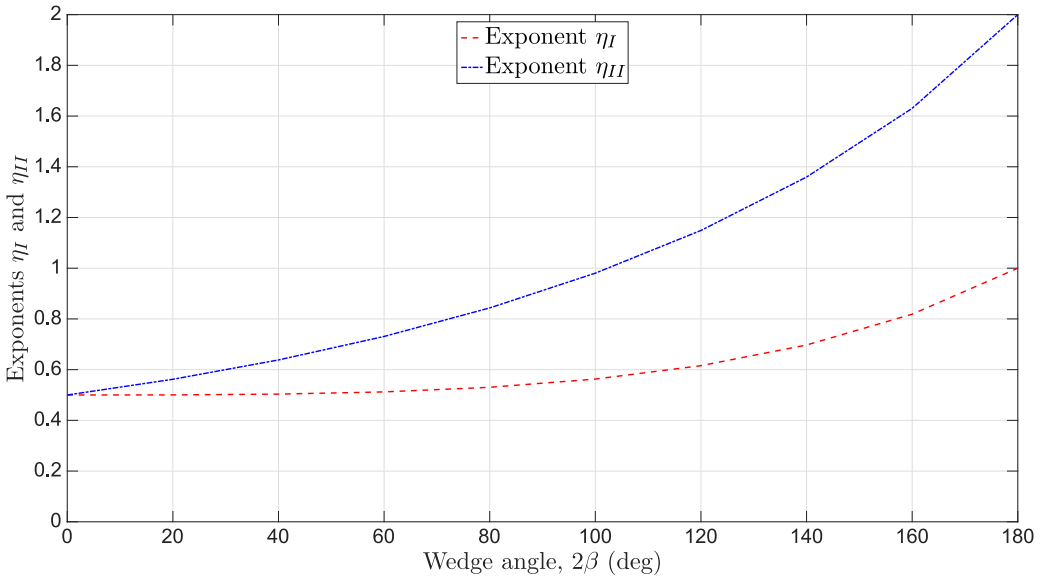


Figure 3.4. Variation of exponents η_I and η_{II} with the wedge angle.

In a pure Mode-I scenario, the circumferential component of stress in the neighbourhood of a crack tip is given by

$$\sigma_{\theta\theta} = \frac{K_I}{\sqrt{2\pi r}}, \tag{3.2.12}$$

where $\eta_I = 0.5$ and the superscript is dropped from the generalised stress intensity factor. A solution tending to infinity is obtained as $r \to 0$. It is worth noting that this singularity is only a mathematical phenomenon. Physically, for example in ceramics, a large number of micro-cracks nucleate or stress-induced phase transformation takes place in the immediate vicinity of a crack tip [28]. The affected region is known as the process zone. The stress is, therefore, re-distributed and the singularity has no material presence. The assumption of LEFM requires the size of the process zone to be very small relative to the macroscopic crack length and encompassing body dimensions.

3.3 Topological sensitivity analysis of the total potential energy problem

3.3.1 Asymptotic expansion

Let a functional defined in a perturbed domain admit the following asymptotic expansion:

$$\chi(\Omega_\xi) = \chi(\Omega) + \sum_{i=1}^{k} f_i(\xi) D_T^{(i)}(\mathbf{x}_0) + \mathscr{R}(f_k(\xi)), \tag{3.3.1}$$

where $D_T^{(i)}(\mathbf{x}_0)$ is known as the topological derivative of the ith order that is evaluated at the point $\mathbf{x}_0 \in \Omega$ representing the centre of perforation. The correction functions $f_k(\xi)$ monotonically tend to zero as $\xi \to 0$ satisfying

$$\lim_{\xi\to 0} \frac{f_j(\xi)}{f_i(\xi)} = 0, \ j > i \text{ and } \lim_{\xi\to 0} \frac{\mathscr{R}(f_k(\xi))}{f_k(\xi)} = 0, \tag{3.3.2}$$

where $\mathscr{R}$ is the remainder function. A formal definition of the first-order topological derivative results from dividing the asymptotic expansion in (3.3.1) by $f(\xi)$ and taking the limit $\xi \to 0$:

$$D_T(\mathbf{x}_0) = \lim_{\xi\to 0} \frac{\chi(\Omega_\xi) - \chi(\Omega)}{f(\xi)}. \tag{3.3.3}$$

Traditional calculus techniques do not serve the topological derivative, as there is no homeomorphism between the original and perturbed domains. However, it can be shown that the topological derivative associated with the extension of the hole $\overline{\omega_\xi}$, when $\xi \to 0$, is in fact equal to that of creating the hole in the first instance. The proof is included in Appendix D.1. Therefore, the case of hole extension is investigated with the aim of developing an expression for the topological derivative that can be readily evaluated. This work is interested in the total potential energy functional, where the domain implicitly features in the solution of (3.2.5), $\chi(\Omega) := \Pi(\mathbf{u})$, and (3.2.7), $\chi(\Omega_\xi) := \Pi(\mathbf{u}^\xi)$.

3.3.2 Topological-shape sensitivity analysis

Let Ω_ξ represent the initial (reference) configuration in the continuum mechanics sense at $\tau = 0$. Assume that there exists a smooth and invertible mapping function $\boldsymbol{\varphi}(\mathbf{x}, \tau)$, where

$\tau \in \mathbb{R}^+$, defining the motion involved in transforming the domain Ω_ξ. For sufficiently small τ, the transformed domain $\Omega_\tau = \Omega_{\xi+\delta\xi}$ and boundary $\partial\Omega_\tau = \partial\Omega_{\xi+\delta\xi}$ can be described by

$$\Omega_\tau := \left\{ \mathbf{x}^\tau \in \mathbb{R}^2 \,|\, \mathbf{x}^\tau = \mathbf{x} + \tau\mathbf{v},\ \mathbf{x} \in \Omega_\xi \right\} \text{ and} \tag{3.3.4a}$$

$$\partial\Omega_\tau := \left\{ \mathbf{x}^\tau \in \mathbb{R}^2 \,|\, \mathbf{x}^\tau = \mathbf{x} + \tau\mathbf{v},\ \mathbf{x} \in \partial\Omega_\xi \right\}, \tag{3.3.4b}$$

where $\mathbf{v}$ is the velocity field. It is the normal component of the velocity prescribed on the boundary $\partial\omega_\xi$ that is effectively responsible for shape change in the form of hole extension in Ω_ξ. For uniform hole expansion, it can be observed that

$$\mathbf{x}^\tau = \mathbf{x} - \tau V \mathbf{n} \quad \forall \mathbf{x} \in \partial\omega_\xi, \tag{3.3.5}$$

where V denotes an arbitrary speed of shape change (scalar) and is taken to be unity. Based on the preceding, it is possible to arrive at a definition for the topological derivative that is in the form of a limit:

$$D_T(\mathbf{x}_0) = \lim_{\xi\to 0} \frac{1}{f'(\xi)} \frac{d}{d\tau}\Pi(\mathbf{u}^\tau)\bigg|_{\tau=0}, \tag{3.3.6}$$

where $f(\xi)$ is a function selected so that $0 < |D_T(\mathbf{x}_0)| < \infty$. The proof is contained in Appendix D.2.

3.3.3 Shape sensitivity

Treating Ω_τ as the current configuration, the time derivative of the displacement can be expressed as

$$\begin{aligned} \dot{\mathbf{u}}^\tau(\mathbf{x}^\tau, \tau) &= \tfrac{\partial}{\partial\tau}\mathbf{u}^\tau(\boldsymbol{\varphi}(\mathbf{x}, \tau), \tau)\big|_{\mathbf{x}} \\ &= \boldsymbol{\zeta}^\tau + (\mathbf{u}^\tau)', \end{aligned} \tag{3.3.7}$$

where $\boldsymbol{\zeta}^\tau = (\nabla\mathbf{u}^\tau)\mathbf{v}$. The shape derivative of the total potential energy functional $\Pi(\mathbf{u}^\tau)$, making use of Reynold's Transport Theorem [29], at $\tau = 0$ is

$$\begin{aligned} \frac{d}{d\tau}\Pi(\mathbf{u}^\tau)\bigg|_{\tau=0} &= \frac{1}{2}\int_{\partial\Omega_\xi} (\boldsymbol{\sigma}(\mathbf{u}^\tau) : \boldsymbol{\varepsilon}(\mathbf{u}^\tau))|_{\tau=0}\,(\mathbf{v}\cdot\mathbf{n})\,ds \\ &\quad + \frac{1}{2}\int_{\Omega_\xi} \frac{\partial}{\partial\tau}(\boldsymbol{\sigma}(\mathbf{u}^\tau) : \boldsymbol{\varepsilon}(\mathbf{u}^\tau))\bigg|_{\tau=0} dy - \int_{\Gamma_N} \mathbf{g}\cdot\dot{\mathbf{u}}^\xi\,ds. \end{aligned} \tag{3.3.8}$$

Then,

$$\begin{aligned} \frac{d}{d\tau}\Pi(\mathbf{u}^\tau)\bigg|_{\tau=0} &= \frac{1}{2}\int_{\partial\Omega_\xi} (\boldsymbol{\sigma}(\mathbf{u}^\xi) : \boldsymbol{\varepsilon}(\mathbf{u}^\xi))\,(\mathbf{v}\cdot\mathbf{n})\,ds - \int_{\Omega_\xi} \boldsymbol{\sigma}(\mathbf{u}^\xi) : \boldsymbol{\varepsilon}(\boldsymbol{\zeta}^\xi)\,dy \\ &\quad + \int_{\Omega_\xi} \boldsymbol{\sigma}(\mathbf{u}^\xi) : \boldsymbol{\varepsilon}(\dot{\mathbf{u}}^\xi)\,dy - \int_{\Gamma_N} \mathbf{g}\cdot\dot{\mathbf{u}}^\xi\,ds, \end{aligned} \tag{3.3.9}$$

where $\boldsymbol{\varepsilon}(\boldsymbol{\zeta}) = \frac{1}{2}(\nabla\boldsymbol{\zeta} + \nabla\boldsymbol{\zeta}^{\mathrm{T}})$. Given that $\dot{\mathbf{u}}^{\xi}$ represents a variation of the displacement with respect to the direction of shape change, the last two terms vanish and the shape derivative becomes

$$\frac{d}{d\tau}\Pi(\mathbf{u}^{\tau})\bigg|_{\tau=0} = \frac{1}{2}\int_{\partial\Omega_{\xi}} (\boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\mathbf{u}^{\xi}))(\mathbf{v}\cdot\mathbf{n})\, ds - \int_{\Omega_{\xi}} \boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\boldsymbol{\zeta}^{\xi})\, dy. \quad (3.3.10)$$

Noting that $\boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\boldsymbol{\zeta}^{\xi}) = \boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \nabla\boldsymbol{\zeta}^{\xi}$, the domain integral term can be written as

$$\int_{\Omega_{\xi}} \boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\boldsymbol{\zeta}^{\xi})\, dy = \int_{\Omega_{\xi}} \operatorname{div}(\boldsymbol{\sigma}(\mathbf{u}^{\xi})\boldsymbol{\zeta}^{\xi})\, dy - \int_{\Omega_{\xi}} \operatorname{div}(\boldsymbol{\sigma}(\mathbf{u}^{\xi}))\cdot\boldsymbol{\zeta}^{\xi}\, dy. \quad (3.3.11)$$

From (3.2.7), the second term on the right-hand side can be disregarded. Applying the divergence theorem of Gauss to the remaining term yields

$$\int_{\Omega_{\xi}} \boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\boldsymbol{\zeta}^{\xi})\, dy = \int_{\partial\Omega_{\xi}} \boldsymbol{\sigma}(\mathbf{u}^{\xi})\boldsymbol{\zeta}^{\xi}\cdot\mathbf{n}\, ds. \quad (3.3.12)$$

Subsequently, the shape derivative can be defined solely on the boundary, as such

$$\frac{d}{d\tau}\Pi(\mathbf{u}^{\tau})\bigg|_{\tau=0} = \frac{1}{2}\int_{\partial\Omega_{\xi}} (\boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\mathbf{u}^{\xi}))(\mathbf{v}\cdot\mathbf{n})\, ds - \int_{\partial\Omega_{\xi}} \boldsymbol{\sigma}(\mathbf{u}^{\xi})\boldsymbol{\zeta}^{\xi}\cdot\mathbf{n}\, ds. \quad (3.3.13)$$

Introducing the Eshelby energy-momentum tensor [10], $\boldsymbol{\Sigma}^{\xi} = \frac{1}{2}(\boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\mathbf{u}^{\xi}))\mathbf{I} - (\nabla\mathbf{u}^{\xi})^{\mathrm{T}}\boldsymbol{\sigma}(\mathbf{u}^{\xi})$ into the equation, leads to

$$\frac{d}{d\tau}\Pi(\mathbf{u}^{\tau})\bigg|_{\tau=0} = \int_{\partial\Omega_{\xi}} \boldsymbol{\Sigma}^{\xi}\mathbf{n}\cdot\mathbf{v}\, ds. \quad (3.3.14)$$

Since only the component of velocity that is normal to the hole boundary is of importance, it follows that

$$\frac{d}{d\tau}\Pi(\mathbf{u}^{\tau})\bigg|_{\tau=0} = -\int_{\partial\omega_{\xi}} \boldsymbol{\Sigma}^{\xi}\mathbf{n}\cdot\mathbf{n}\, ds. \quad (3.3.15)$$

Taking into account the homogeneous Neumann condition imposed on the hole boundary results in

$$\frac{d}{d\tau}\Pi(\mathbf{u}^{\tau})\bigg|_{\tau=0} = -\frac{1}{2}\int_{\partial\omega_{\xi}} \boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\varepsilon}(\mathbf{u}^{\xi})\, ds. \quad (3.3.16)$$

In order to obtain a definition in terms of stress, we must revisit (3.2.1) and observe that

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \boldsymbol{C}^{-1}\boldsymbol{\sigma}(\mathbf{u}), \quad (3.3.17)$$

where

$$\boldsymbol{C}^{-1} = \frac{1+\nu}{E}\mathbf{II} - \frac{\nu}{E}\mathbf{I}\otimes\mathbf{I}. \quad (3.3.18)$$

All that remains is to substitute (3.3.17) into (3.3.16), and finally

$$\frac{d}{d\tau}\Pi(\mathbf{u}^{\tau})\bigg|_{\tau=0} = -\frac{1}{2E}\int_{\partial\omega_{\xi}} \left[(1+\nu)\boldsymbol{\sigma}(\mathbf{u}^{\xi}) : \boldsymbol{\sigma}(\mathbf{u}^{\xi}) - \nu\operatorname{tr}^{2}(\boldsymbol{\sigma}(\mathbf{u}^{\xi}))\right] ds. \quad (3.3.19)$$

3.3.4 Topological derivative

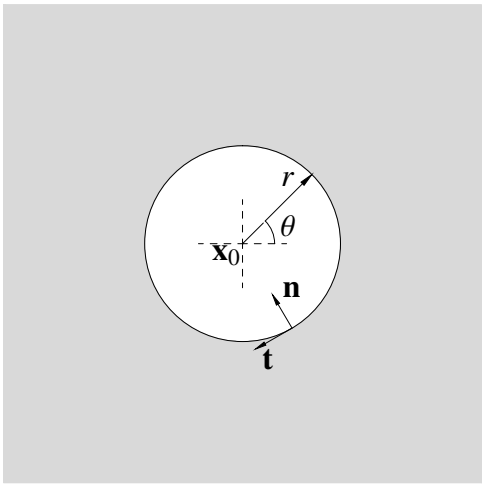


Figure 3.5. Orthonormal coordinate system $(\mathbf{t}, \mathbf{n})$ defined on the hole boundary $\partial\omega_\xi$.

For the sake of mathematical simplicity, let the hole take the form of a ball with radius $\xi > 0$. Introducing an orthonormal coordinate system $(\mathbf{t}, \mathbf{n})$ defined on the boundary $\partial\omega_\xi$, see Figure 3.5, where $\mathbf{t}$ is the unit tangential vector. The stress tensor can then be decomposed thus

$$\boldsymbol{\sigma}(\mathbf{u}^\xi)_{|\partial\omega_\xi} = \sigma_{tt}(\mathbf{t}\otimes\mathbf{t}) + \sigma_{tn}(\mathbf{t}\otimes\mathbf{n}) + \sigma_{nt}(\mathbf{n}\otimes\mathbf{t}) + \sigma_{nn}(\mathbf{n}\otimes\mathbf{n}). \tag{3.3.20}$$

The following are inferred given the homogeneous Neumann condition on the hole boundary:

$$\sigma_{nn}(\mathbf{u}^\xi) = \sigma_{tn}(\mathbf{u}^\xi) = \sigma_{nt}(\mathbf{u}^\xi) = 0 \text{ on } \partial\omega_\xi. \tag{3.3.21}$$

Only the tangential component of the stress remains, which upon substitution into (**3.3.19**) yields

$$\frac{d}{d\tau}\Pi(\mathbf{u}^\tau)\bigg|_{\tau=0} = -\frac{1}{2E}\int_{\partial\omega_\xi}(\sigma_{tt}(\mathbf{u}^\xi))^2\,ds \tag{3.3.22}$$

in the plane stress case. Note that the tangential stress coincides with the circumferential component of stress in a polar coordinate system centred at $\mathbf{x}_0 \in \Omega$, i.e. centre of the perforation.

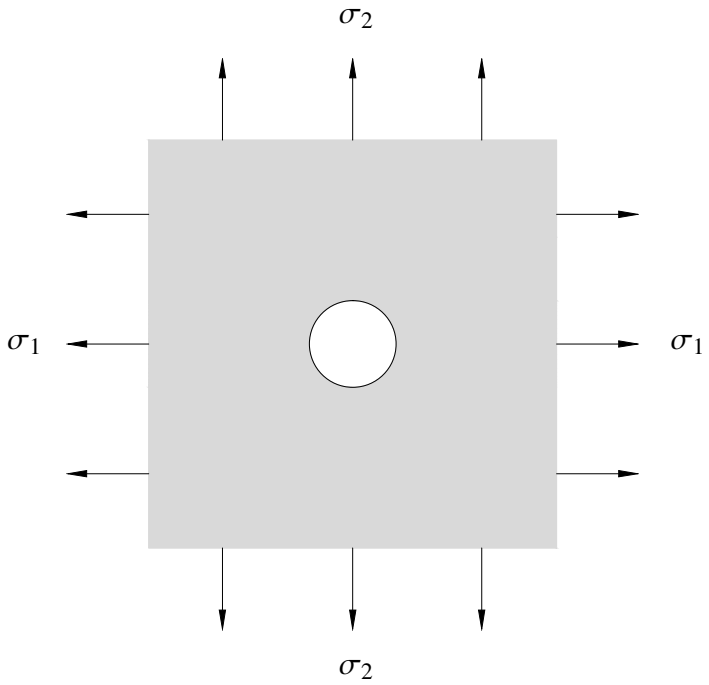


Figure 3.6. A circular hole in an infinite plate subject to remote stresses.

The stress distribution around a hole with a traction-free boundary, in a polar coordinate system centred at $\mathbf{x}_0 \in \Omega$, subject to the remote stresses σ_1 and σ_2 (refer to [30]) as shown in Figure 3.6 is given by

$$\sigma_{rr}(\mathbf{u}^\xi) = \phi_1 \left(1 - \frac{\xi^2}{r^2}\right) + \phi_2 \left(1 - 4\frac{\xi^2}{r^2} + 3\frac{\xi^4}{r^4}\right) \cos 2\theta + O(\xi), \tag{3.3.23a}$$

$$\sigma_{\theta\theta}(\mathbf{u}^\xi) = \phi_1 \left(1 + \frac{\xi^2}{r^2}\right) - \phi_2 \left(1 + 3\frac{\xi^4}{r^4}\right) \cos 2\theta + O(\xi), \tag{3.3.23b}$$

$$\sigma_{r\theta}(\mathbf{u}^\xi) = -\phi_2 \left(1 + 2\frac{\xi^2}{r^2} - 3\frac{\xi^4}{r^4}\right) \sin 2\theta + O(\xi), \tag{3.3.23c}$$

where $\phi_1 = \frac{1}{2}(\sigma_1 + \sigma_2)$ and $\phi_2 = \frac{1}{2}(\sigma_1 - \sigma_2)$. The remote stresses σ_1 and σ_2 correspond to principal stresses that are eigenvalues of the stress tensor $\boldsymbol{\sigma}(\mathbf{u})$ at $\mathbf{x}_0 \in \Omega$:

$$\sigma_{1,2}(\mathbf{u}) = \frac{1}{2}\left(\operatorname{tr}\boldsymbol{\sigma}(\mathbf{u}) \pm \sqrt{2\boldsymbol{\sigma}_D(\mathbf{u}) : \boldsymbol{\sigma}_D(\mathbf{u})}\right), \tag{3.3.24}$$

in which

$$\boldsymbol{\sigma}_D(\mathbf{u}) = \boldsymbol{\sigma}(\mathbf{u}) - \frac{1}{2}\operatorname{tr}\boldsymbol{\sigma}(\mathbf{u})\mathbf{I}. \tag{3.3.25}$$

At the hole boundary, $r = \xi$, the circumferential stress is

$$\sigma_{\theta\theta}(\mathbf{u}^\xi) = 2\phi_1 - 4\phi_2 \cos 2\theta + O(\xi). \tag{3.3.26}$$

Substituting this result into the shape derivative in (3.3.22),

$$\frac{d}{d\tau}\Pi(\mathbf{u}^\tau)\bigg|_{\tau=0} = -\frac{1}{2E}\int_0^{2\pi} (2\phi_1 - 4\phi_2\cos 2\theta)^2\,\xi\,d\theta + O(\xi^2), \tag{3.3.27}$$

and integrating yields

$$\frac{d}{d\tau}\Pi(\mathbf{u}^\tau)\bigg|_{\tau=0} = -\frac{4\pi\xi}{E}\left(\phi_1^2 + 2\phi_2^2\right) + O(\xi^2). \tag{3.3.28}$$

Let $f(\xi) = \pi\xi^2$ and taking the limit $\xi \to 0$ to obtain a closed-form expression for the topological derivative that can be evaluated at a point $\mathbf{x}_0$ in the virgin domain Ω:

$$D_T(\mathbf{x}_0) = -\frac{1}{2E}\left[(\sigma_1(\mathbf{u}) + \sigma_2(\mathbf{u}))^2 + 2(\sigma_1(\mathbf{u}) - \sigma_2(\mathbf{u}))^2\right]. \tag{3.3.29}$$

Eshelby's inclusion method [31] provides an alternative route to this result. The topological derivative in stress tenor form is

$$D_T(\mathbf{x}_0) = -\frac{1}{2E}\left[4\boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\sigma}(\mathbf{u}) - \mathrm{tr}^2(\boldsymbol{\sigma}(\mathbf{u}))\right]. \tag{3.3.30}$$

In terms of stress and strain, the topological derivative for plane stress has the form

$$D_T(\mathbf{x}_0) = -\frac{2}{(1+\nu)}\boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u}) + \frac{1-3\nu}{2(1-\nu^2)}\mathrm{tr}(\boldsymbol{\sigma}(\mathbf{u}))\mathrm{tr}(\boldsymbol{\varepsilon}(\mathbf{u})). \tag{3.3.31}$$

The plane strain version of the topological derivative can be written as

$$D_T(\mathbf{x}_0) = -2(1-\nu)\boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u}) + \frac{(1-4\nu)(1-\nu)}{2(1-2\nu)}\mathrm{tr}(\boldsymbol{\sigma}(\mathbf{u}))\mathrm{tr}(\boldsymbol{\varepsilon}(\mathbf{u})). \tag{3.3.32}$$

The topological derivative as formulated above returns a scalar value, which quantifies the sensitivity of the total potential energy to the introduction of a small circular hole of radius ξ centred at an arbitrary point $\mathbf{x}_0 \in \Omega$.

3.4 Fracture criteria

Fracture in materials is fundamentally a multi-scale process. In polycrystalline materials, crack propagation in the microstructure can evolve across grains or their boundaries. Grains have anisotropic properties and their neighbours do not necessarily share the same orientation of crystallographic planes. Furthermore, defects such as dislocations influence crack evolution. The task of modelling crack propagation at the micro-structural level remains an enormous challenge, whether it is the coupled multi-physics representation or the intensive computational effort. This has resulted in the continuing development of fracture criteria

relevant to the macroscopic continuum. These criteria are generally classified according to material failure type (i.e. brittle, quasi-brittle or ductile), rather than taking a material-specific form.

The following criteria for continua in a brittle state are reviewed in this section: (i) the strength criterion; (ii) the Griffith energy criterion; (iii) the minimum strain energy density (SED) criterion; (iv) the modified McClintock criterion; and (v) the Novozhilov-Seweryn criterion. This list is by no means exhaustive; it does, however, represent the most relevant and historically significant theories in the field of fracture mechanics. The context for the V-notch references made here on in can be found in Section 3.2.3.

3.4.1 Strength criterion

Fracture is assumed to occur when the tensile stress acting on the surface of a structural element reaches the value of the material strength, that is

$$\sigma = \sigma_c. \tag{3.4.1}$$

This condition is ineffective at a notch tip where stress tends to infinity, as a positive fracture outcome will always be returned.

3.4.2 Griffith energy criterion

Griffith approached the subject of fracture from a thermodynamic perspective, postulating that the energy necessary for creating new surfaces must be supplied by a release in the elastic body [32]. The energy associated with surface creation in a body with unit thickness is $\Upsilon = 2\gamma a$, where γ is the surface energy density and a is the crack length. The critical energy release rate is defined as

$$G_c = 2\gamma, \tag{3.4.2}$$

which is regarded as a material property.

Consider a homogeneous and isotropic linear elastic body that is subject to loading in an equilibrium state. Investigating the change in energy following the propagation of a finite-length crack in the body:

$$\delta\Pi + \delta U + G_c \delta a = 0, \tag{3.4.3}$$

where U is the kinetic energy. Only the static case will be pursued further, $\delta U = 0$, therefore

$$G_c = -\frac{\delta \Pi}{\delta a}. \tag{3.4.4}$$

Assuming that the crack evolves in a continuous manner and taking the limit $\delta a \to 0$, leads to

$$G = -\frac{d\Pi}{da}, \tag{3.4.5}$$

where G is the energy release rate. Thus, the necessary condition for crack propagation is $G = G_c$.

The energy release rate is intimately related to the stress intensity by the following [33]:

$$\sqrt{G\bar{E}} = K_I + K_{II}, \tag{3.4.6}$$

where $\bar{E} = E$ for plane stress and $\bar{E} = E/(1-\nu^2)$ for plane strain. The critical energy release rate has an equivalent in a stress intensity form, known as the fracture toughness, which for Mode-I is

$$K_{Ic} = \sqrt{G_c \bar{E}}. \tag{3.4.7}$$

Clearly, the necessary condition for crack propagation can also be stated in terms of stress intensity, that is $K = K_{Ic}$.

This criterion can only predict crack extension. The differential form of the energy release rate otherwise vanishes and the necessary condition for crack propagation would be unfulfilled.

3.4.3 Minimum SED criterion

The fundamental quantity assessed in this criterion is the strain energy density factor, which is defined as the product of a distance and strain energy density:

$$S = \varrho_c \frac{dW}{dy}. \tag{3.4.8}$$

The strain energy density dW/dy is evaluated at points on an arc located at a radial distance ϱ_c from the V-notch tip as shown in Figure 3.7.

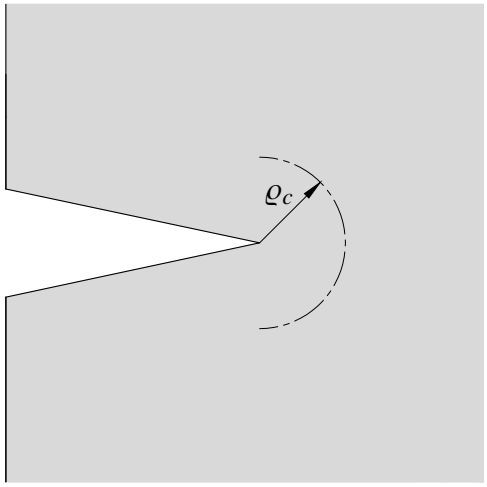


Figure 3.7. Radial distance ϱ_c from the V-notch tip.

The strain energy stored in the element $dy = rd\theta dr$ can be written in the form

$$dW = \frac{1}{8\mu}[\kappa(\sigma_{rr} + \sigma_{\theta\theta})^2 + (\sigma_{rr} - \sigma_{\theta\theta})^2 + 4\sigma_{r\theta}^2]dy, \tag{3.4.9}$$

where $\kappa = (1 - \nu)/(1 + \nu)$ for plane stress and $\kappa = 1 - 2\nu$ for plane strain.

Sih in [34] assumed that crack propagation occurs when a critical value is reached in the direction θ_c that minimises the strain energy density factor. This can be expressed mathematically as

$$S(\varrho_c, \theta) = S_c, \tag{3.4.10a}$$

where

$$\frac{\partial S}{\partial \theta} = 0 \text{ and } \frac{\partial^2 S}{\partial \theta^2} > 0. \tag{3.4.10b}$$

The critical value signifying fracture is related to the fracture toughness [34] by

$$S_c = \frac{\varsigma - 1}{8\pi\mu} K_{Ic}^2, \tag{3.4.11}$$

where $\varsigma = (3 - \nu)/(1 + \nu)$ for plane stress and $\varsigma = 3 - 4\nu$ for plane strain.

Consider a plate loaded in tension with symmetrical semi-circular notches that is in a state of plane stress. Assuming that the stress gradient is approximately uniform near the notch root. Then, applying the strength criterion to diagnose local failure, the critical strain energy density factor is given by

$$S_c = \frac{1}{2}\frac{\sigma_c^2}{E}\varrho_c. \tag{3.4.12}$$

Equating (3.4.12) with (3.4.11) yields

$$\varrho_c = \frac{1-\nu}{\pi}\left(\frac{K_{Ic}}{\sigma_c}\right)^2. \tag{3.4.13}$$

A criticism of this criterion is that it is not part of a physical or mathematical theory justifying the calculation of strain energy density at a distance from the notch tip. Additionally, the criterion as formulated can only predict crack onset at a V-notch tip.

3.4.4 Modified McClintock criterion

McClintock studying crack propagation in ductile materials [35] proposed that crack propagation occurs when the normal component of strain at a small distance ahead of the crack tip reaches a critical value. The criterion was subsequently adapted to stress and applied to brittle materials. Crack propagation is assumed to occur in the direction where the maximum circumferential stress at a radial distance w_c from the V-notch tip reaches the value of the material strength. Mathematically, this reads

$$\sigma_{\theta\theta}(w_c, \theta) = \sigma_c, \tag{3.4.14a}$$

where

$$\frac{\partial \sigma_{\theta\theta}}{\partial \theta} = 0 \text{ and } \frac{\partial^2 \sigma_{\theta\theta}}{\partial \theta^2} < 0. \tag{3.4.14b}$$

An expression for w_c can be obtained by considering the pure Mode-I crack tip stress field. Setting the circumferential component in (3.2.12) to the material strength yields

$$\sigma_c = \frac{K_I}{\sqrt{2\pi w_c}}. \tag{3.4.15}$$

Re-arranging and noting that the maximum the process zone size can assume is at $K_I = K_{Ic}$ provides

$$w_c = \frac{1}{2\pi}\left(\frac{K_{Ic}}{\sigma_c}\right)^2. \tag{3.4.16}$$

This is recognisable as the plane stress version of the plastic zone length in Irwin's model [36].

A single stress parameter may well be insufficient to describe crack propagation. If this is the case, the strength condition in the criterion can be replaced by a general function:

$$M_\sigma\left(\frac{\sigma_{\theta\theta}}{\sigma_c}, \frac{\sigma_{r\theta}}{\iota_c}\right) = 1, \tag{3.4.17}$$

where ι_c is the shear strength. For example, $M_\sigma(\Phi, \Psi)$ could take the form of an elliptic condition in the tensile regime,

$$M_\sigma = \left(\Phi^2 + \Psi^2\right)^{\frac{1}{2}}, \tag{3.4.18}$$

and a Mohr-Coulomb condition in the compressive regime (refer to [37]).

Additionally, the modified McClintock criterion has been applied to predict failure in plates with a central hole subject to uniaxial tensile loading [38]. Crack propagation, in this case, is assumed to occur in the direction normal to the hole boundary. Fracture is then predicted when the stress normal to the plane of crack extension achieves the value of the material strength at a distance w_c from the boundary, $\sigma_{nn} = \sigma_c$. An alternative formula for w_c is given in [38]:

$$w_c = \frac{1}{\pi}\left(\frac{K_{Ic}}{1.122\sigma_c}\right)^2. \tag{3.4.19}$$

3.4.5 Novozhilov-Seweryn criterion

Novozhilov argued that fracture in solids is a discrete process and not continuous as described by (**3.4.5**). The supporting example provided in [24] highlights the fact that it is not possible to separate two atoms and terminate halfway whilst effectively retaining the bond between the non-separated halves. Novozhilov sought a criterion where fracture is assumed to occur when the average stress taken over an effective length ahead of a crack tip attains a critical value:

$$\int_0^{z_c} \sigma_{nn}\, dr = z_c \sigma_B, \tag{3.4.20}$$

where σ_{nn} is the stress normal to the plane of crack extension, σ_B is the strength parameter and z_c represents a characteristic length. In the aforementioned example, Novozhilov took z_c to be twice the atomic radius and σ_B the (defect-free) atomic lattice strength. Seweryn et al. re-expressed the criterion for sharp notches in macroscopic media with $\sigma_B = \sigma_c$ [39].

Substituting the stress in the immediate vicinity of a crack tip in a pure Mode-I scenario, as given by (**3.2.12**), into (**3.4.20**) and setting $K_I = K_{Ic}$ yields

$$z_c = \frac{2}{\pi}\left(\frac{K_{Ic}}{\sigma_c}\right)^2. \tag{3.4.21}$$

Similarly to the modified McClintock criterion, a general function can be employed in place of the single stress parameter if the condition is insufficient:

$$\frac{1}{z_c}\int_0^{z_c} M_\sigma\, dr = 1. \tag{3.4.22}$$

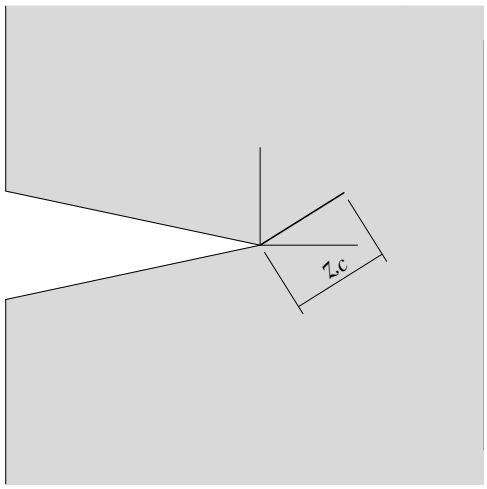


Figure 3.8. Characteristic length taken from V-notch tip.

The fracture condition can be written in terms of force with the characteristic length emanating from the V-notch tip (see Figure 3.8). This is mathematically expressed as

$$N(\theta) = N_c, \tag{3.4.23a}$$

where

$$\frac{dN}{d\theta} = 0 \text{ and } \frac{d^2N}{d\theta^2} < 0. \tag{3.4.23b}$$

The Novozhilov-Seweryn criterion has also been applied to predict failure in plates with a central hole subject to uniaxial tensile loading in [38]. Crack propagation, as before, is assumed to occur in the direction that is normal to the boundary. Accordingly, only condition (**3.4.23a**) must be fulfilled. Once again, an alternative formula for z_c is provided in [38]:

$$z_c = \frac{2}{\pi}\left(\frac{K_{Ic}}{1.122\sigma_c}\right)^2. \tag{3.4.24}$$

3.5 Proposed theory

3.5.1 Discrete crack propagation

Crack nucleation in this criterion is treated as a sudden and discrete rupture event at the macroscopic level. This negates the need to consider continuity arguments both mathematical

and structural (in the material sense with respect to the linking of scales). Applying the finite difference method to the energy release rate defined in (**3.4.5**) and setting $G = G_c$ leads to

$$G_c = -\frac{\Pi(l + a_c) - \Pi(l)}{a_c}, \tag{3.5.1}$$

where l is the existing crack length and a_c represents the characteristic length.

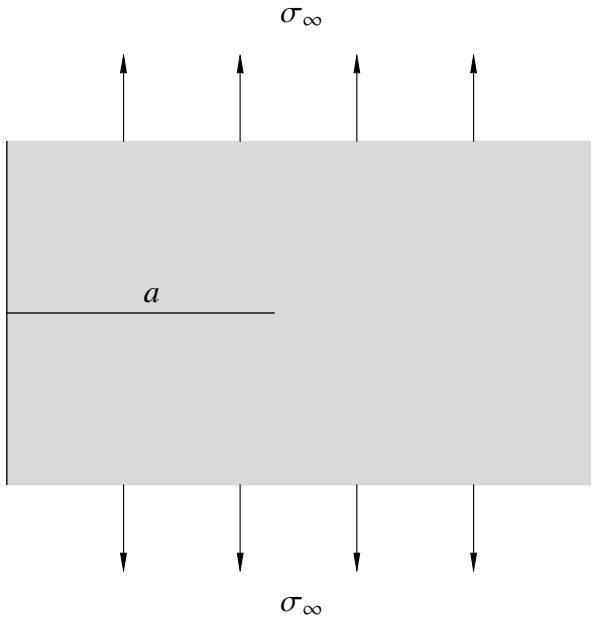


Figure 3.9. Small edge crack in an infinite plate subject to tensile loading.

3.5.2 Characteristic length

The first step in obtaining a definition of the characteristic length involves the integration of (**3.4.5**). The energy release associated with the extension of a crack by a small finite length p is written as

$$\int_l^{l+p} G\, da = \Pi(l) - \Pi(l + p). \tag{3.5.2}$$

Considering a Mode-I scenario and employing the relation in (**3.4.6**) results in

$$\frac{1}{\bar{E}} \int_l^{l+p} K_I^2\, da = \Pi(l) - \Pi(l + p). \tag{3.5.3}$$

The Mode-I stress intensity factor of a small edge-crack contained in an infinite plate that is subject to uniaxial tension (refer to [36]) as depicted in Figure 3.9 is

$$K_I = Y\sigma_\infty\sqrt{\pi a}, \tag{3.5.4}$$

where σ_∞ is the remote stress and Y is a non-dimensional constant accounting for crack location; $Y = 1$ for a centre crack and $Y = 1.122$ for an edge crack. Substituting (3.5.4) into (3.5.3) yields

$$\frac{1}{\bar{E}} \int_l^{l+p} Y^2 \sigma_\infty^2 \pi a \, da = \Pi(l) - \Pi(l+p). \tag{3.5.5}$$

Let $p = a_c$ and from (3.5.1), we have

$$\frac{1}{\bar{E}} \int_l^{l+a_c} Y^2 \sigma_\infty^2 \pi a \, da = a_c G_c. \tag{3.5.6}$$

Introducing the fracture toughness as defined in (3.4.7), produces

$$\frac{1}{a_c} \int_l^{l+a_c} Y^2 \sigma_\infty^2 \pi a \, da = K_{Ic}^2. \tag{3.5.7}$$

Finally, evaluating for a vanishing crack, $l = 0$, and noting the bounding stress $\sigma_\infty = \sigma_c$ provides

$$a_c = \frac{2}{\pi} \left(\frac{K_{Ic}}{Y \sigma_c} \right)^2. \tag{3.5.8}$$

3.5.3 Energy change

It is necessary to re-express the potential energy states in (3.5.1) as functions of the displacement field, as such

$$a_c G_c = \Pi(\mathbf{u}) - \Pi(\mathbf{u}^\xi). \tag{3.5.9}$$

The unperturbed domain containing the initial crack is implicitly featured in the solution $\mathbf{u}$ of the boundary value problem of linear elastostatics (just as applicable to domains with a non-smooth boundary). This formulation is not restricted to representing cracks only but is more general in nature and can describe any notch geometry. The superscript ξ denotes the introduction of a small circular hole in the virgin domain, which involves an equal amount of energy dissipation to the finite crack extension. As such, the total potential energy of the body is conserved, i.e. $\Pi(l + a_c) \equiv \Pi(\mathbf{u}^\xi)$.

The first-order asymptotic expansion in (3.3.1) with the remainder term ignored can then be employed to approximate the change in energy associated with the introduction of a small hole, that is

$$\Pi(\mathbf{u}) - \Pi(\mathbf{u}^\xi) = -f(\xi) D_T(\mathbf{x}_0). \tag{3.5.10}$$

An expression for the topological derivative can be selected from (3.3.29)-(3.3.32) as required and $f(\xi)$ is the area of the circular hole with radius $\xi > 0$. Evaluation of the topological derivative occurs at points (perforation centres) in the unperturbed domain, which are located at a distance ξ from the boundary (that may or may not be smooth). The key assumption here is that the topological derivative remains meaningful even if the hole boundary $\partial\omega_\xi$ impinges on but not exceeds the domain boundary.

3.5.4 Virtual hole radius

Consider a crack in a pure Mode-I fracture scenario. The propagation direction is known and an expression for the stress ahead of the crack tip, prior to fracture, is available. Therefore, the distance from the tip along the crack propagation direction, illustrated in Figure 3.10, where the change in total potential energy associated with hole creation satisfies $a_c G_c$ can be taken as the radius ξ_c. Substituting the topological derivative as given by (3.3.29) into (3.5.9) yields

$$a_c G_c = \frac{\pi \xi_c^2}{2\bar{E}} \left[(\sigma_1 + \sigma_2)^2 + 2(\sigma_1 - \sigma_2)^2 \right]. \tag{3.5.11}$$

Acknowledging the necessary condition for crack propagation, that is $K_I = K_{Ic}$, the principal stress ahead of a crack tip is $\sigma_1 = \sigma_2 = K_{Ic}/\sqrt{2\pi\xi}$ and

$$a_c G_c = \frac{K_{Ic}^2}{\bar{E}} \xi_c. \tag{3.5.12}$$

Taking (3.5.8) into account, the following is deduced:

$$\xi_c = \frac{2}{\pi} \frac{G_c \bar{E}}{(Y\sigma_c)^2}. \tag{3.5.13}$$

The radius is equal to the characteristic length when the two-dimensional plane assumption in the (plane-strain) fracture toughness calculation matches that of the body containing the crack.

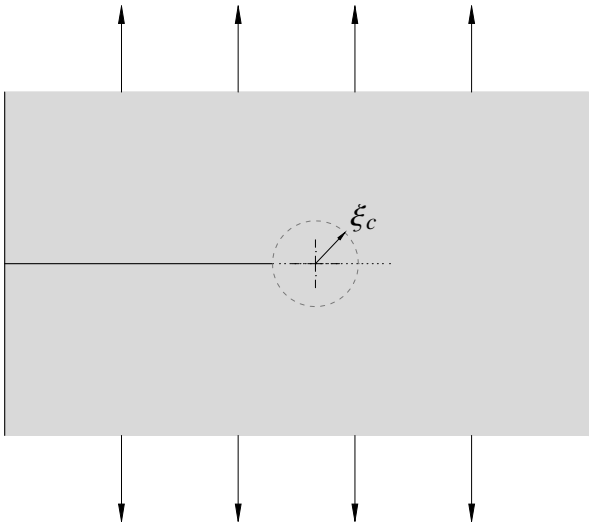


Figure 3.10. Virtual hole radius calibration.

The hole radius as defined in (3.5.13) is not suitable for predicting crack onset at a blunt notch. It will be assumed that crack propagation from blunt notches can be predicted locally by the (tensile) strength condition. Setting $\sigma_1 = \sigma_c$ and $\sigma_2 = 0$ in (3.5.11) results in

$$a_c G_c = \frac{3}{2}\frac{\pi\sigma_c^2}{\bar{E}}\xi_c^2 \text{ and } \xi_c = \sqrt{\frac{2a_c G_c \bar{E}}{3\pi\sigma_c^2}}. \tag{3.5.14}$$

3.5.5 Fracture condition

Crack propagation from a V-notch tip is assumed to occur in the direction where the minimum change in total potential energy associated with the introduction of a small circular hole in an unperturbed domain, representative of finite crack extension, satisfies the critical factor $a_c G_c$. The centre of the hole must be located at a distance ξ_c from the V-notch tip. Introducing $Q = -f(\xi)D_T(\mathbf{x}_0)$, the fracture condition in a polar coordinate system (r, θ) centred at the V-notch tip can be stated as

$$Q(\xi_c, \theta) = a_c G_c, \tag{3.5.15a}$$

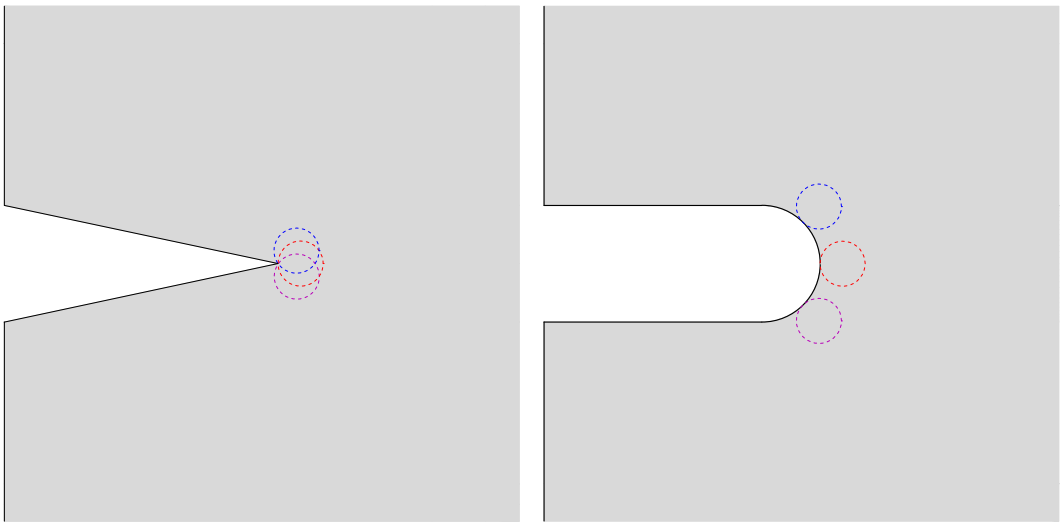
where

$$\frac{\partial Q}{\partial \theta} = 0 \text{ and } \frac{\partial^2 Q}{\partial \theta^2} > 0. \tag{3.5.15b}$$

In the vicinity of a blunt notch, the condition $Q = a_c G_c$ must be fulfilled with the centre of the hole located at a distance ξ_c from the boundary. Interestingly, the proposed criterion

involves an area calculation (although evaluated at a point), while the reviewed criteria are either point-wise in nature or involve a length.

Numerically, the criterion can be implemented as a post-processing procedure in a boundary or finite element framework. Once a solution for the displacement field is available in an unperturbed domain, the topological derivative can be evaluated at points located at a distance ξ_c from the boundary as shown in Figure 3.11. The virtual hole representing the energy change that satisfies $a_c G_c$ provides the fracture load, location of crack onset and propagation direction.



(a) (b)

Figure 3.11. Potential crack nucleation sites: (a) V-notch and (b) U-notch.

The generality of the proposed criterion is guaranteed as long as the circular hole does not exceed the domain boundary. However, the radius of the hole will differ based on the notch geometry as discussed. This should not detract from the methodology or hamper any serious effort at computational automation.

3.6 Validation

3.6.1 Experiments

In this section, predictions made by the proposed and reviewed criteria are compared to experimental data. The experiments considered are those available in literature:

- Double edge V-notched specimens subject to biaxial loading [40];
- Circular notched specimens subject to tensile loading [38]; and
- U-notched specimens subject to three-point bend tests [41].

Polymethyl metacrylate (PMMA) is the material of choice in the aforementioned experiments as it exhibits brittle characteristics and behaves in a near-linear elastic manner. The mechanical properties of the specimens in each experiment are reported in Table 3.1.

Table 3.1. Specimen (PMMA) material properties.

Property	V-notched [40]	Circular notched [38]	U-notched (@ −60°C) [41]
E (GPa)	3.3	3	5.05
ν	0.35	0.36	0.4
σ_c (MPa)	102.8	72	128.4
K_{Ic} (MPa$\sqrt{\text{m}}$)	1.202	0.9998	1.7

In the ensuing analyses, specimens along with their imposed boundary conditions are modelled as two-dimensional linear elasticity problems; the double edge V-notched specimen experiment as plane stress and the U-notched as well as circular notched specimen experiments as plane strain. Analytical formulae can then be relied on to calculate the stress field surrounding the V-notch tip and around the circular notch. Finite element analysis is, however, required to calculate the stress in the U-notched specimens.

The calculated values of the non-local parameters in the proposed and reviewed criteria are presented in Table 3.2. The minimum SED criterion is only valid for double edge V-notched specimens (see Section 3.4.3). Only the proposed criterion is applied to predict fracture in the U-notched specimens. This is due to the expensive computational overhead, with respect to both time and memory.

Table 3.2. Non-local fracture criteria parameters.

Criterion	V-notched	Circular notched	U-notched
Minimum SED, ϱ_c (mm)	0.02829	-	-
Modified McClintock, w_c (mm)	0.0218	0.04875	-
Novozhilov-Seweryn, z_c (mm)	0.087	0.09751	-
Proposed:			
a_c (mm)	0.06914	0.09751	0.08865
ξ_c (mm)	0.06067	0.06316	0.05742

3.6.2 V-notched specimens

Seweryn et al. [37] developed a device where specimens with double edge V-notches are subjected to biaxial loading. The device is designed to operate in a tensile machine, where a load F is applied to the ends as shown in Figure 3.12. It is the specimen orientation, ψ,

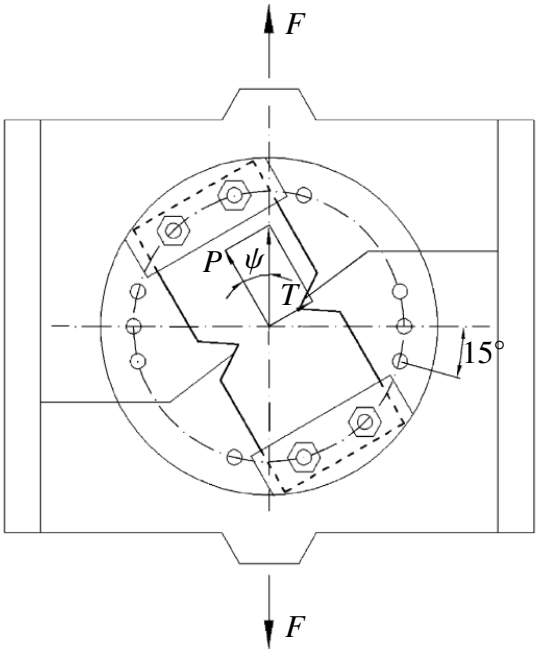


Figure 3.12. Bi-axial loading device (reproduced from [37]).

within the device that governs the loading regime; at the extremes of $\psi = 0°$ and $\psi = 90°$ the double edge V-notched specimen is subject to pure tension and pure shear, respectively. The specimen can be varied by 15° angle increments in the device. The load F can be resolved

into tensile, P, and shear, T, components (see Figure 3.13):

$$P = F\cos\psi \text{ and} \tag{3.6.1a}$$

$$T = F\sin\psi. \tag{3.6.1b}$$

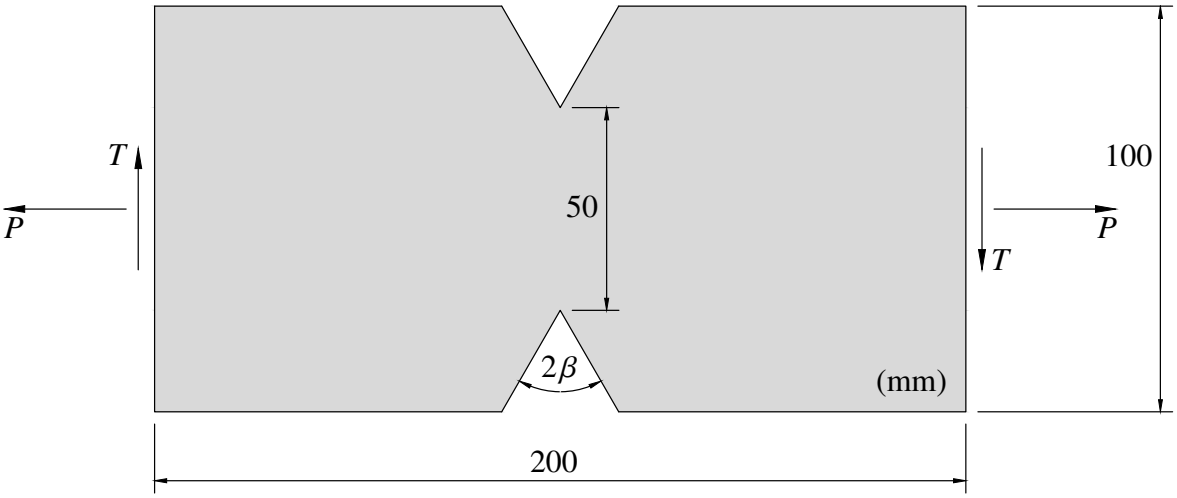


Figure 3.13. Double edge V-notched specimen geometry and loading scheme; Thickness = 5 mm.

Ratios of the generalised Mode-I and Mode-II stress intensity factors to tensile and shear loads, respectively, were included in [40] for specific wedge angles. These were adapted for use in this work (see Table 3.3). The stress in the immediate vicinity of a V-notch tip can, therefore, be readily calculated given a load F. Crack initiation criteria can then be evaluated and the resulting predictions compared with the experimental data available in [37]. MATLAB [23] was selected for the task of the single-variable optimisation. In the case of the stress-based criteria, the maximum was determined by Newton's method as an analytical form of the differentials is easily obtained. For energy-based criteria, the minimum was first bracketed and the **fminbnd** function (optimisation without a derivative in an interval) available in MATLAB was used.

Table 3.3. Ratios of generalised stress intensity factors to tensile and shear loads.

β (deg)	η_I	K_I^η/P (MPam$^{1-\eta_I}$/kN)	η_{II}	K_{II}^η/T (MPam$^{1-\eta_{II}}$/kN)
0	0.5	0.6133	0.5	-
10	0.5004	0.6171	0.5620	0.7550
20	0.5035	0.6364	0.6382	1.2867
30	0.5122	0.6824	0.7309	2.2581
40	0.5304	0.7772	0.8434	4.3444

The pure Mode-I fracture loads, P_c, predicted by the proposed and reviewed criteria for $\beta = 0°$, $10°$, $20°$, $30°$ and $40°$ are recorded in Table 3.4.

Table 3.4. Predicted pure Mode-I fracture loads.

β (deg)	P_c (kN)			
	Modified McClintock	Novozhilov-Seweryn	SED	Proposed
0	1.9599	1.9599	1.9599	1.9599
10	1.9563	1.9567	1.9710	1.9704
20	1.9611	1.9653	2.0199	2.0146
30	2.0080	2.0225	2.1411	2.1217
40	2.1435	2.1800	2.3840	2.3312

The fracture loads normalised by the relevant P_c, predicted as well as experimental, and the crack propagation directions are plotted against ψ in Figures 3.14-3.17 for $\beta = 10°$, $20°$, $30°$ and $40°$. Discounting $\beta = 40°$, the propagation direction predictions made by the proposed criterion are generally in good agreement with the experimental data. However, the proposed criterion consistently predicted the maximum propagation angle that can be assumed before the circular hole breaches the domain boundary, i.e. $\theta = -(90° - \beta)$, at $\psi = 90°$. In fact, it was observed that no unique minimum exists at $\psi = 90°$!

The normalised fracture loads predicted by the proposed criterion correlate fairly well with their experimental counterparts, except at $\psi = 90°$ for $\beta > 10°$. It is noteworthy that the modified McClintock and the Novozhilov-Seweryn criteria both yield predictions close to or in-line with experimental data at $\psi = 90°$ with a single exception being the normalised fracture load at $\beta = 40°$. While the minimum SED criterion poorly predicts the crack propagation direction at $\psi = 90°$, the corresponding normalised fracture loads are accurately captured. The predictive capability of the proposed criterion for pure Mode-II fracture warrants further study, but will not form part of this work.

3.6.3 Circular notched specimens

Li and Zhang [38] investigated crack initiation from a non-singular stress concentration site. They conducted an experimental study involving plates with a centrally located circular hole subject to tensile loading, as shown in Figure 3.18. The following notch diameters were

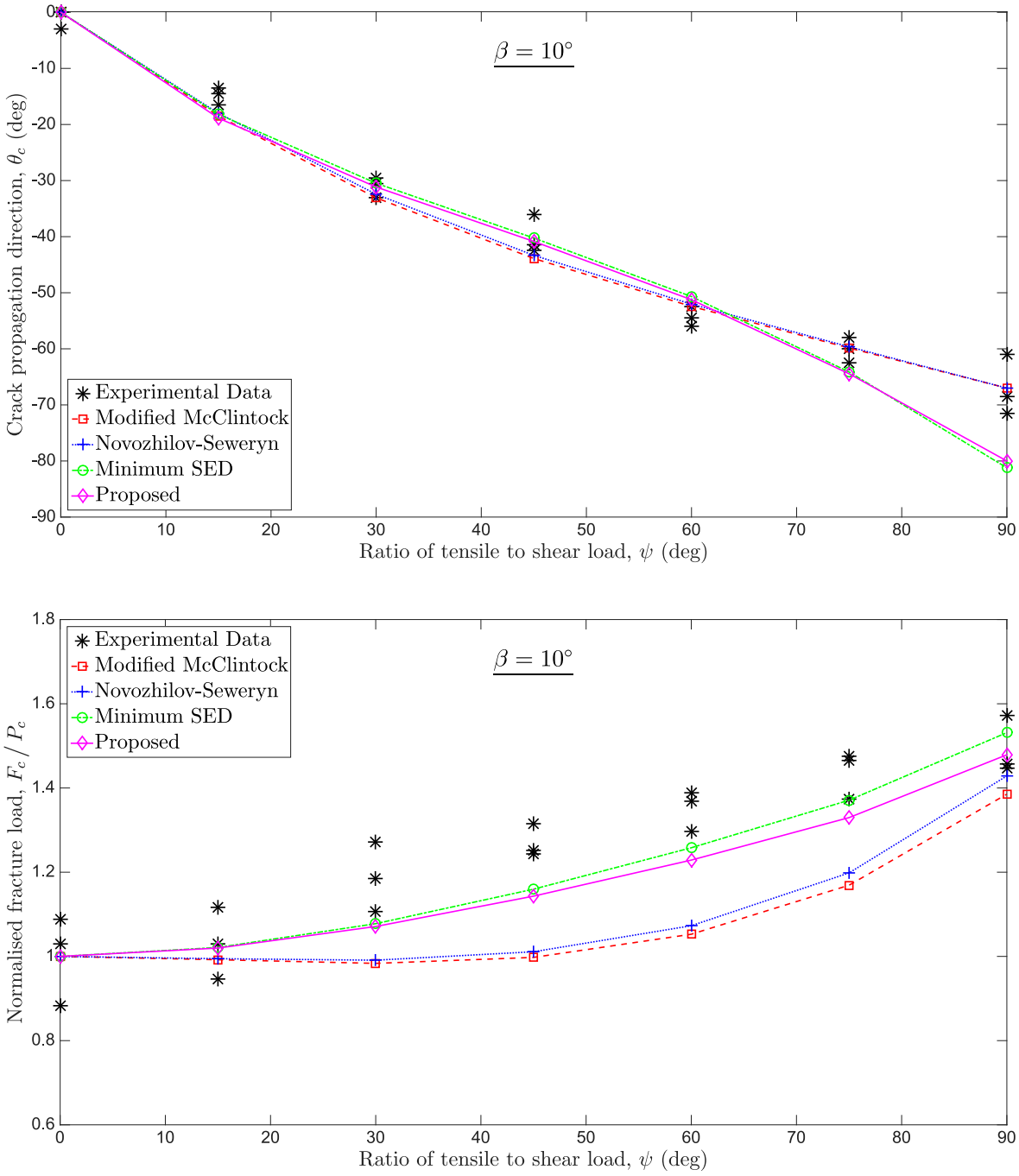


Figure 3.14. Crack propagation directions and normalised fracture loads for specimens with wedge angle $2\beta = 20°$.

considered: $2\rho = 0.6$, 1.2, 2 and 3 mm. The experimental data was reported in plot form and therefore required extraction, which was accomplished using WebPlotDigitizer [42], for reference purposes.

The stress distribution around a hole with a traction-free boundary subject to a remote tensile stress is given by (**3.3.23**) with $\sigma_1 = 0$ and $\sigma_2 = \sigma_\infty$. As before, the criteria

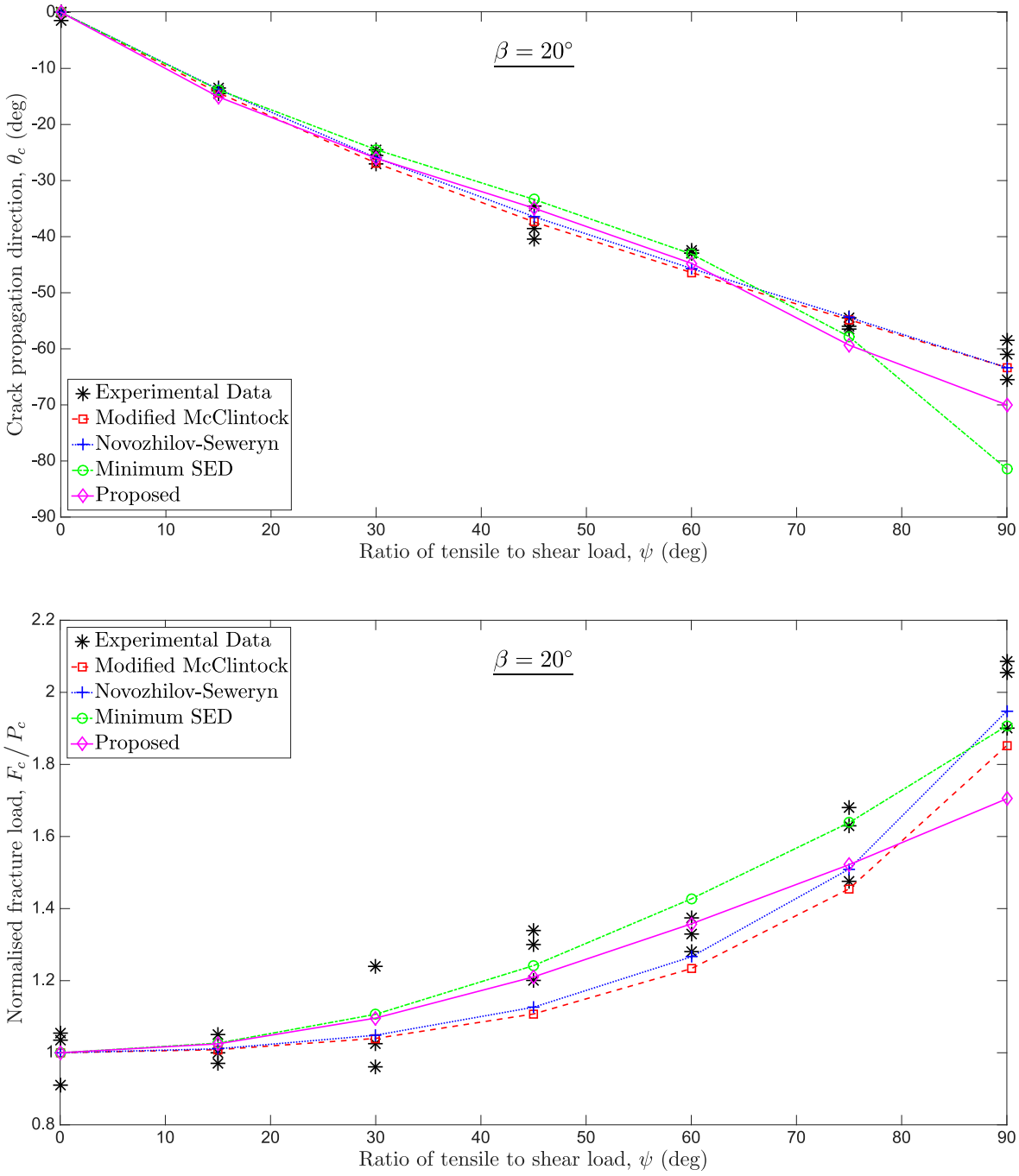


Figure 3.15. Crack propagation directions and normalised fracture loads for specimens with wedge angle $2\beta = 40°$.

were evaluated in MATLAB. Fracture stress predictions relating to the modified McClintock, Novozhilov-Seweryn and proposed criteria are normalised by the material strength and plotted against the notch radius in Figure 3.19. It can be observed that the results are comparable for $\rho = 1.5$ mm. However, as the radius decreases, the proposed criterion yields less conservative predictions. Nonetheless, these are still somewhat shy of the experimental data. This is not an altogether surprising result given that the remit of the proposed criterion does not extend

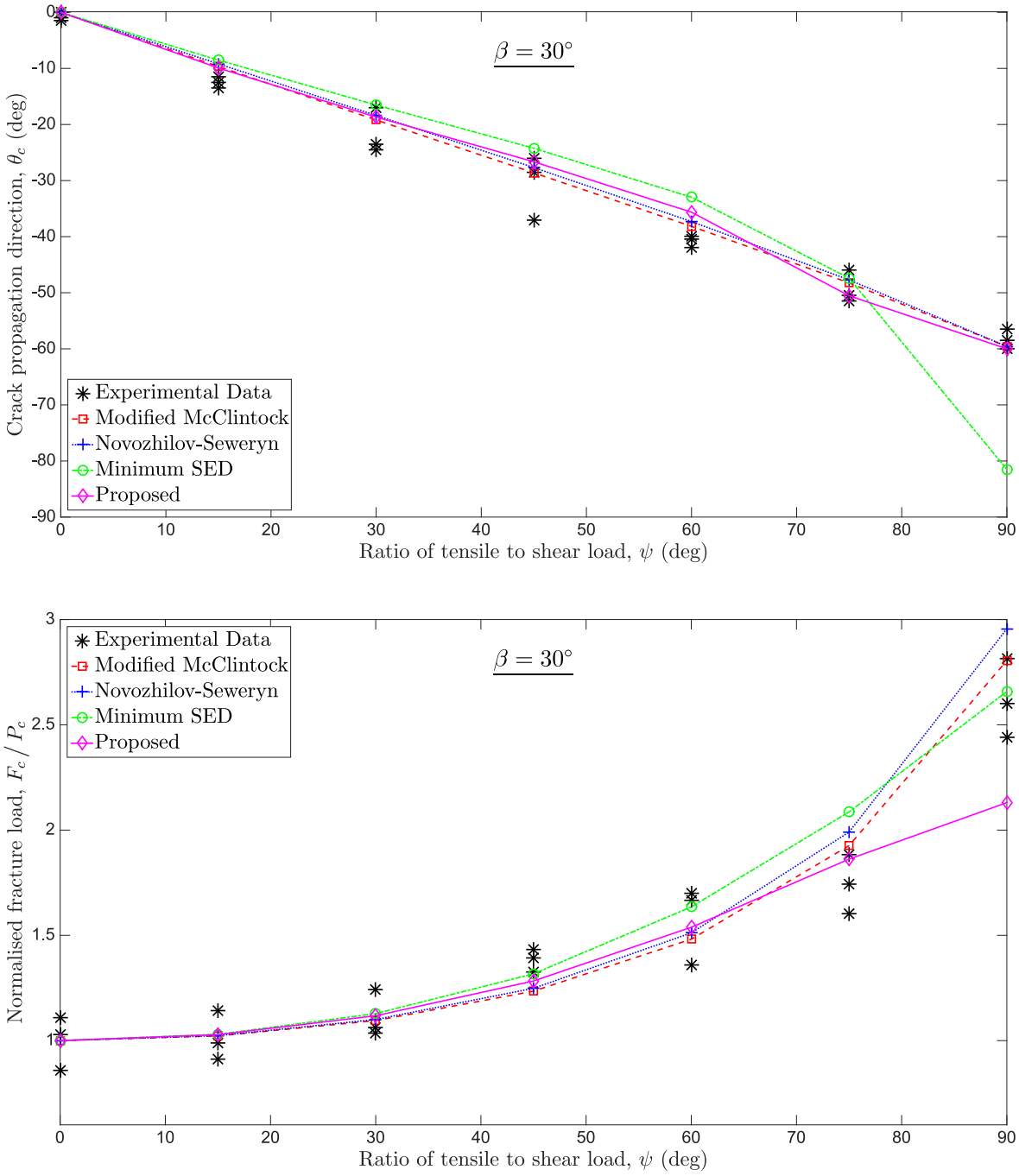


Figure 3.16. Crack propagation directions and normalised fracture loads for specimens with wedge angle $2\beta = 60°$.

to the prediction of crack propagation from non-macroscopic notches. The value of this experiment is, firstly, to demonstrate application of the criterion to specimens with circular notches, and secondly, to highlight the limits of application.

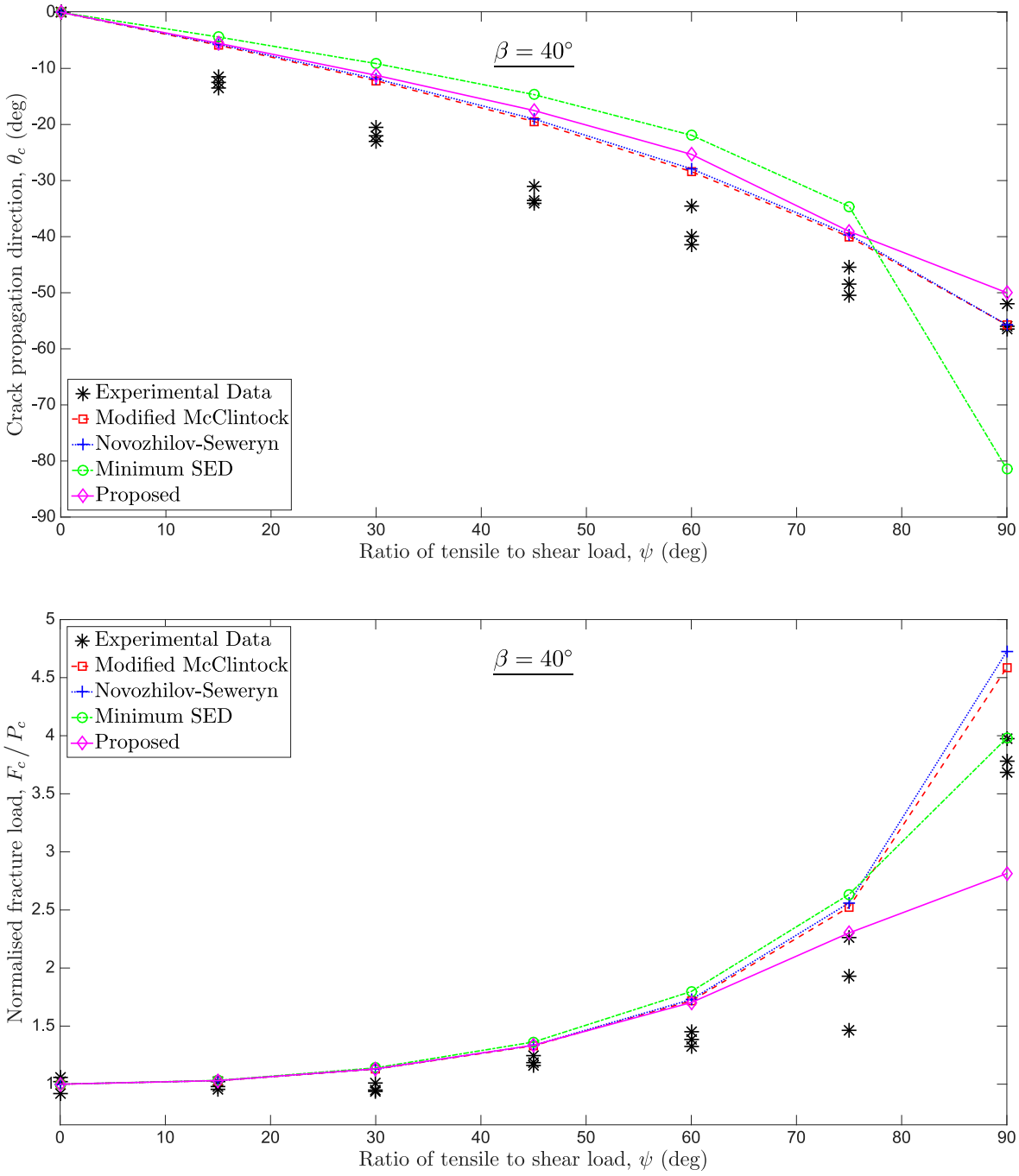


Figure 3.17. Crack propagation directions and normalised fracture loads for specimens with wedge angle $2\beta = 80°$.

3.6.4 U-notched specimens

The three-point bend test in [41] was employed to study crack propagation in U-notched specimens at a temperature of −60°C. Figure 3.20 shows the experimental set-up consisting of two supports and a point-load. Eccentric loading was used to induce mixed-mode fracture

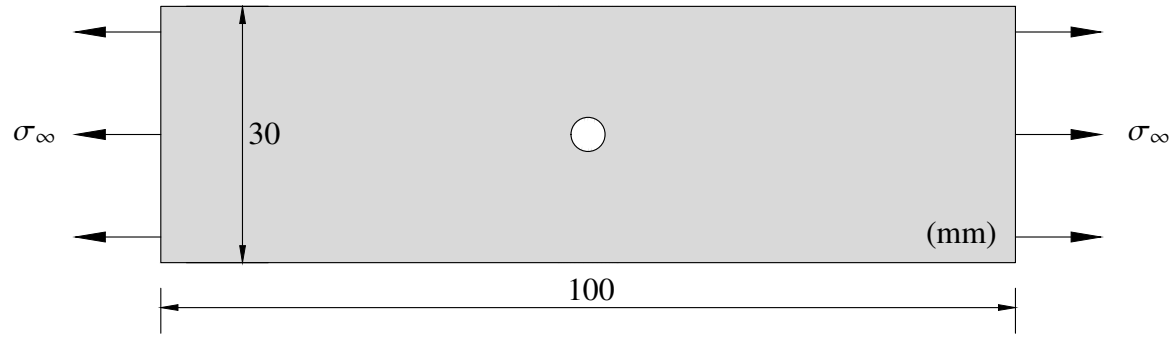


Figure 3.18. Circular notched specimen geometry and loading scheme; Thickness = 10 mm.

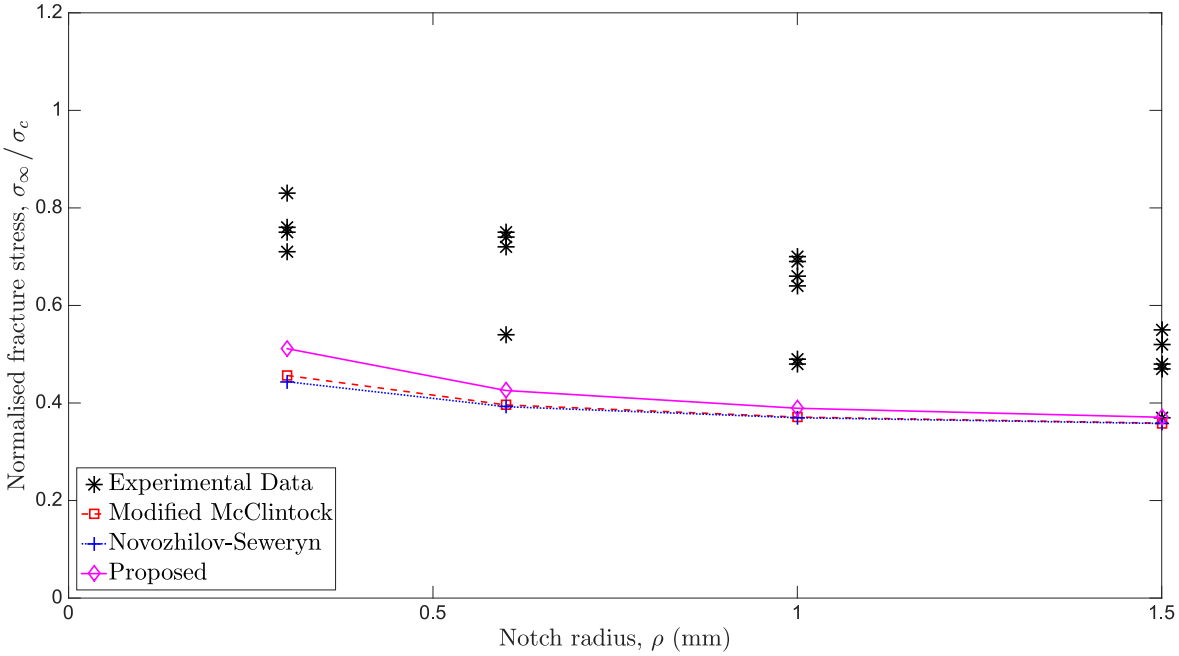


Figure 3.19. Normalised fracture stress for circular notched specimens.

in the U-notched specimens. Four loading positions were considered in the experimental study: q = 9, 18, 27 and 36 mm. Additionally, the influence of the notch root radius at each loading position was investigated; the selected radii were: ρ = 0, 0.2, 0.3, 0.5, 1, 2, and 4 mm. The radius corresponding to that of a crack does not form part of the following analysis.

The three-point bend test was modelled numerically in FreeFem++ [43], which is a partial differential equation solver based on the finite element method (FEM). FreeFem++ was supplied with the U-notched specimen geometry and variational formulation of the problem in (**3.2.5**) as required. Unfortunately, FreeFem++ does not support point boundary conditions. These were, therefore, defined over a very small length, namely 0.04 mm for $\rho < 2$ mm and 0.06 mm for ρ = 2 and 4 mm. Mesh generation based on a Delaunay-Voronoi algorithm is an automatic feature of FreeFem++. A highly refined mesh was specified in the vicinity of the U-notch for improved stress resolution. In fact, a typical mesh consisted of over three

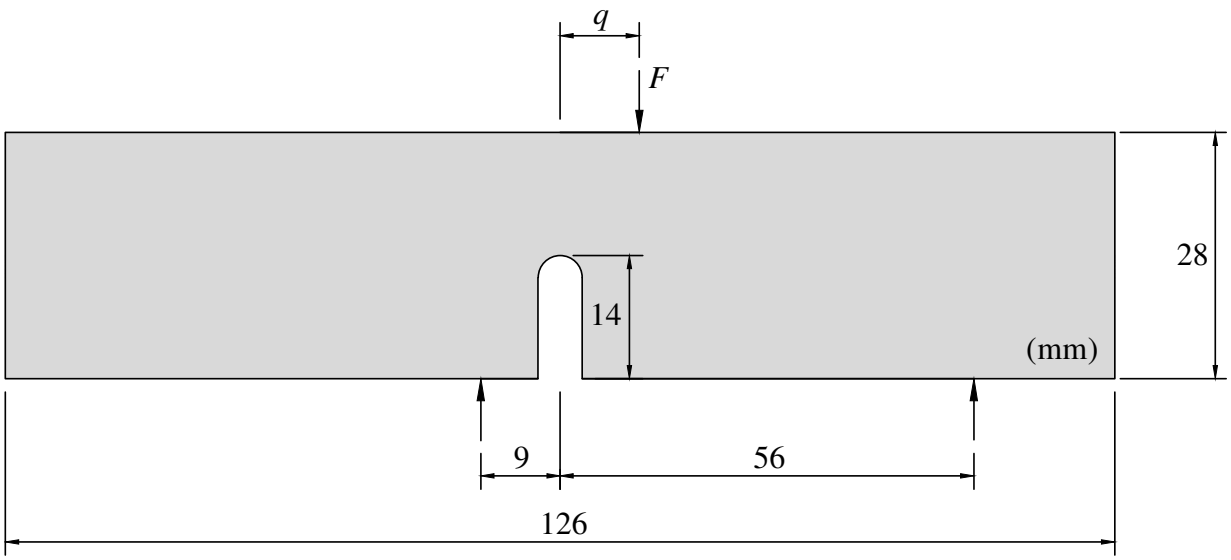


Figure 3.20. U-notched specimen geometry and loading; Thickness = 14 mm.

hundred thousand triangular elements. Continuous piecewise quartic finite elements were employed to approximate the displacement and stress solutions. If the boundary conditions were specified over a smaller length, then there would have been far too many elements in the mesh and the problem would not have been computationally tractable. FreeFem++ permits the interrogation of the solution at any given point in the mesh. This made it possible to evaluate the criterion at points a distance ξ_c from the boundary.

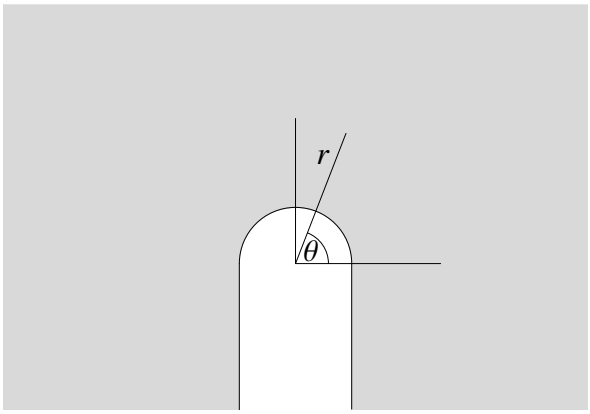


Figure 3.21. Polar coordinate system at centre of notch root radius.

In order to obtain confidence in the modelling strategy, the maximum principal stress $\sigma_{\max}$ on the notch root boundary and corresponding angle θ (see Figure 3.21) were obtained for each load position and notch root radius. The results were then compared with those calculated in [41]. It can be seen from Table 3.5 that as ρ decreases, the difference between the data increases. This difference is not overly significant and can be attributed to the point boundary condition definition, mesh density or finite element employed.

Table 3.5. Comparison of numerical results.

		Exp.	Numerical model in [41]		FreeFem++	
ρ (mm)	q (mm)	F_c (N)	$\sigma_{\max}$ (MPa)	θ (deg)	$\sigma_{\max}$ (MPa)	θ (deg)
4	36	9714	152.13	55.0	152.55	54.8
	27	6430	147.33	55.0	147.71	54.8
	18	5107	155.05	55.2	155.44	55.2
	9	4182	149.36	62.0	149.77	62.3
2	36	7786	152.94	53.5	153.73	53.5
	27	5052	145.14	53.0	145.87	53.2
	18	4081	155.30	53.5	156.12	52.6
	9	3589	161.87	59.0	162.70	59.2
1	36	6064	156.85	52.7	158.27	52.8
	27	4391	166.06	51.7	167.57	51.7
	18	3511	175.96	52.2	177.49	52.8
	9	2729	162.67	56.9	164.12	57.3
0.5	36	4716	165.27	52.0	168.47	52.3
	27	3458	177.44	51.9	180.61	52.3
	18	2561	174.17	52.0	177.21	51.2
	9	2188	177.27	55.8	180.37	55.7
0.3	36	4636	206.19	52.0	212.26	50.8
	27	3172	206.28	52.0	212.36	50.8
	18	2349	202.44	52.0	208.20	50.8
	9	1949	200.19	55.0	205.67	52.7
0.2	36	3289	177.27	52.0	185.72	51.3
	27	2127	167.62	51.0	175.63	51.3
	18	1548	161.73	52.2	169.25	54.3
	9	1308	162.99	52.2	170.73	54.3

The initial crack angle and fracture loads, predicted as well as experimental, are plotted against the notch root radii in Figures 3.22-3.25 for q = 9, 18, 27 and 36 mm. Apart from the notch root radii ρ = 0.2 and 0.3 mm, the predicted propagation directions are in excellent agreement with the experimental data. Furthermore, the proposed criterion provides sound predictions for the fracture loads. However, there are a few exceptions at ρ = 0.2 and 4 mm.

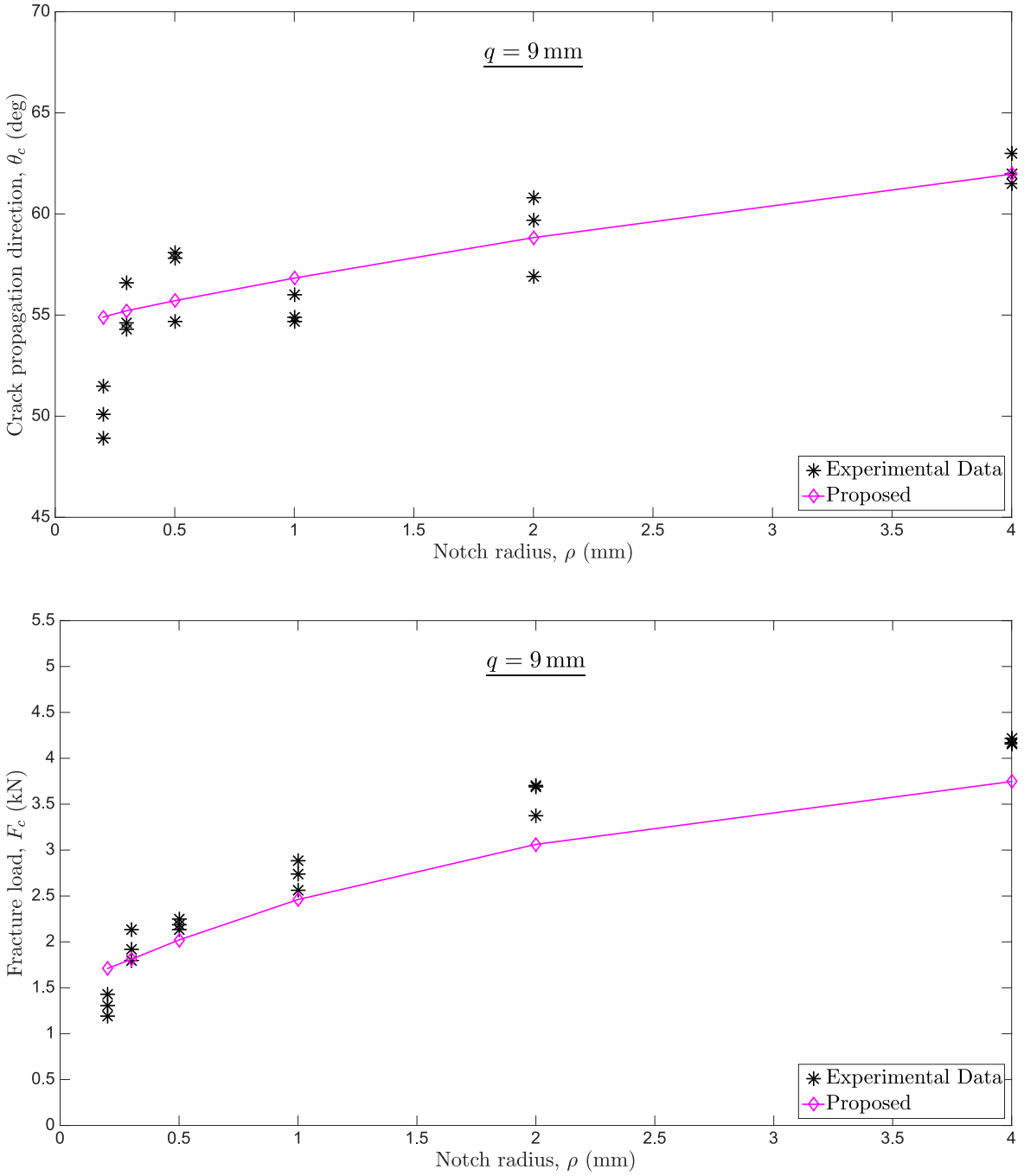


Figure 3.22. Crack initiation angles and fracture loads for U-notched specimens with loading at $q = 9$ mm.

3.7 Conclusions

A crack initiation criterion for brittle materials has been developed, which models nucleation as a sudden and discrete rupture event at the macroscopic level. The criterion is based on the finite difference form of the energy release rate equation. An energy equivalent formulation

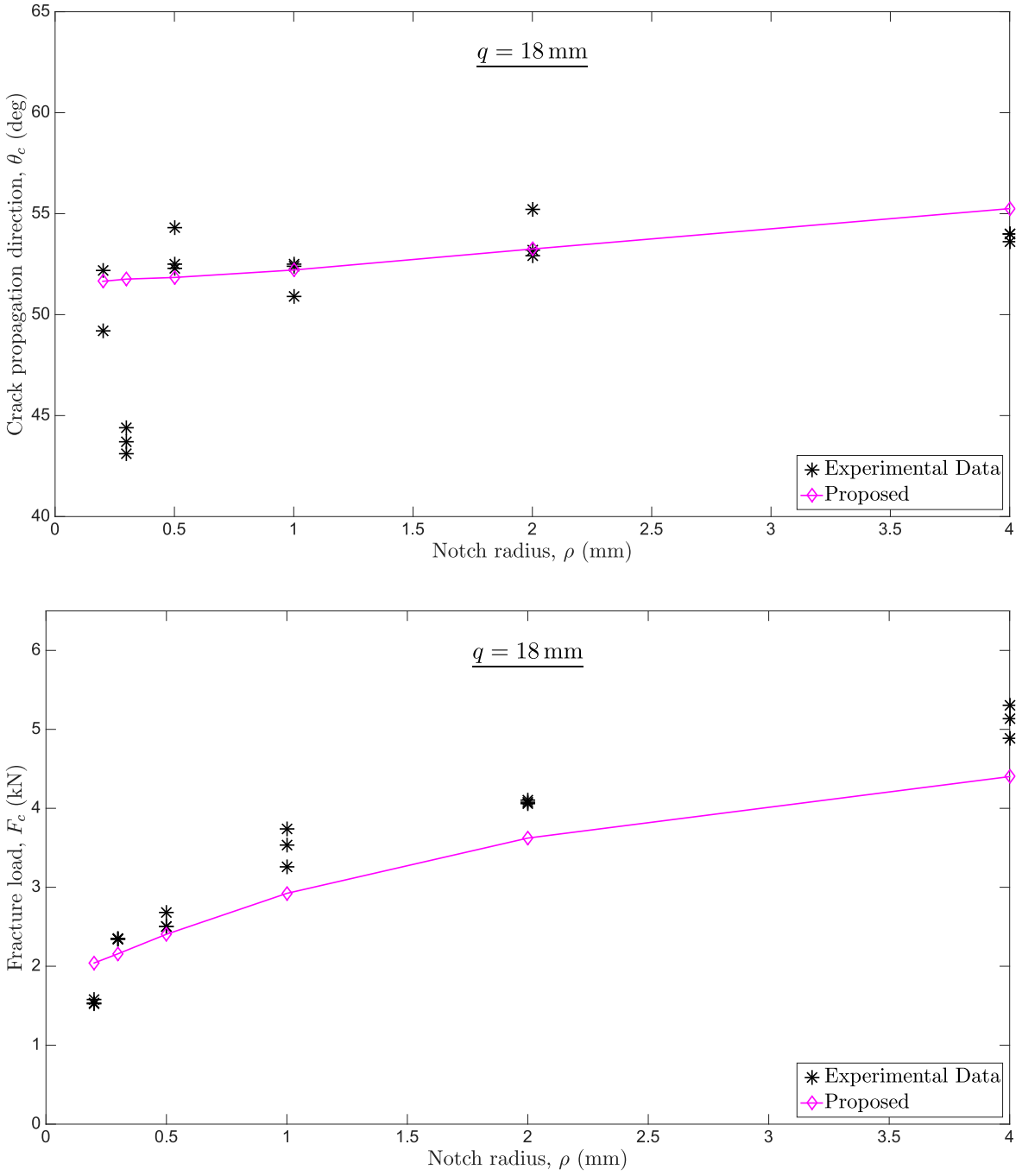


Figure 3.23. Crack initiation angles and fracture loads for U-notched specimens with loading at $q = 18$ mm.

of the Novozhilov-Seweryn criterion provides a definition of the characteristic length. The change in total potential energy associated with domain perturbation is approximated through an asymptotic argument involving the topological derivative. The only physical parameters required to apply the criterion are the fracture toughness and material (tensile) strength.

Fracture in the proposed criterion is assumed to occur when the change in total potential

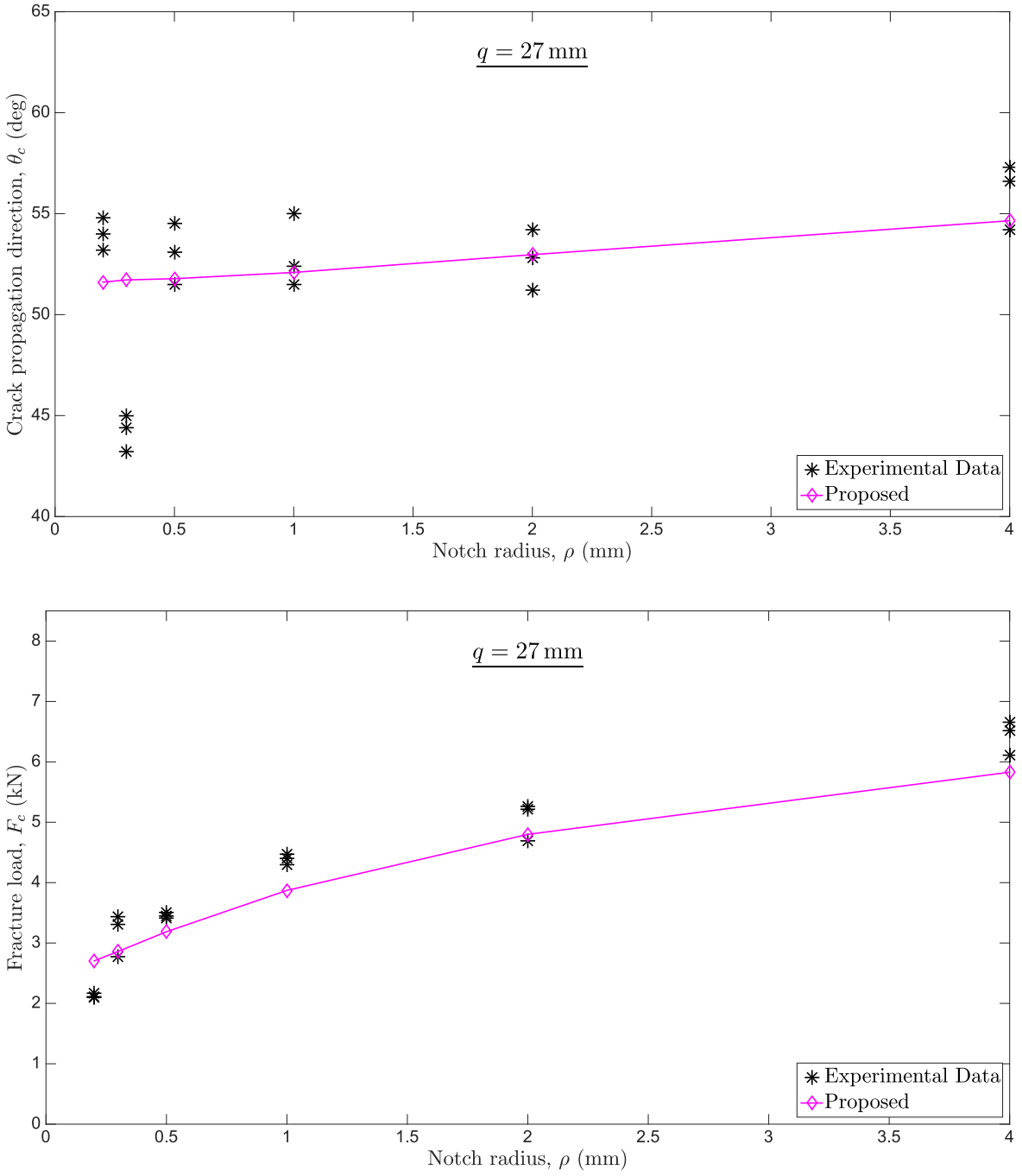


Figure 3.24. Crack initiation angles and fracture loads for U-notched specimens with loading at $q = 27$ mm.

energy associated with the creation of a small circular hole in a body, representative of a finite-length crack, is equal to the product of the characteristic length and the critical energy release rate. While the criterion can predict crack onset at a sharp or blunt notch, the radius of the (virtual) perturbing hole must be set accordingly. Additionally, the criterion can be easily implemented in a numerical setting as a post-processing procedure in a finite or boundary element framework.

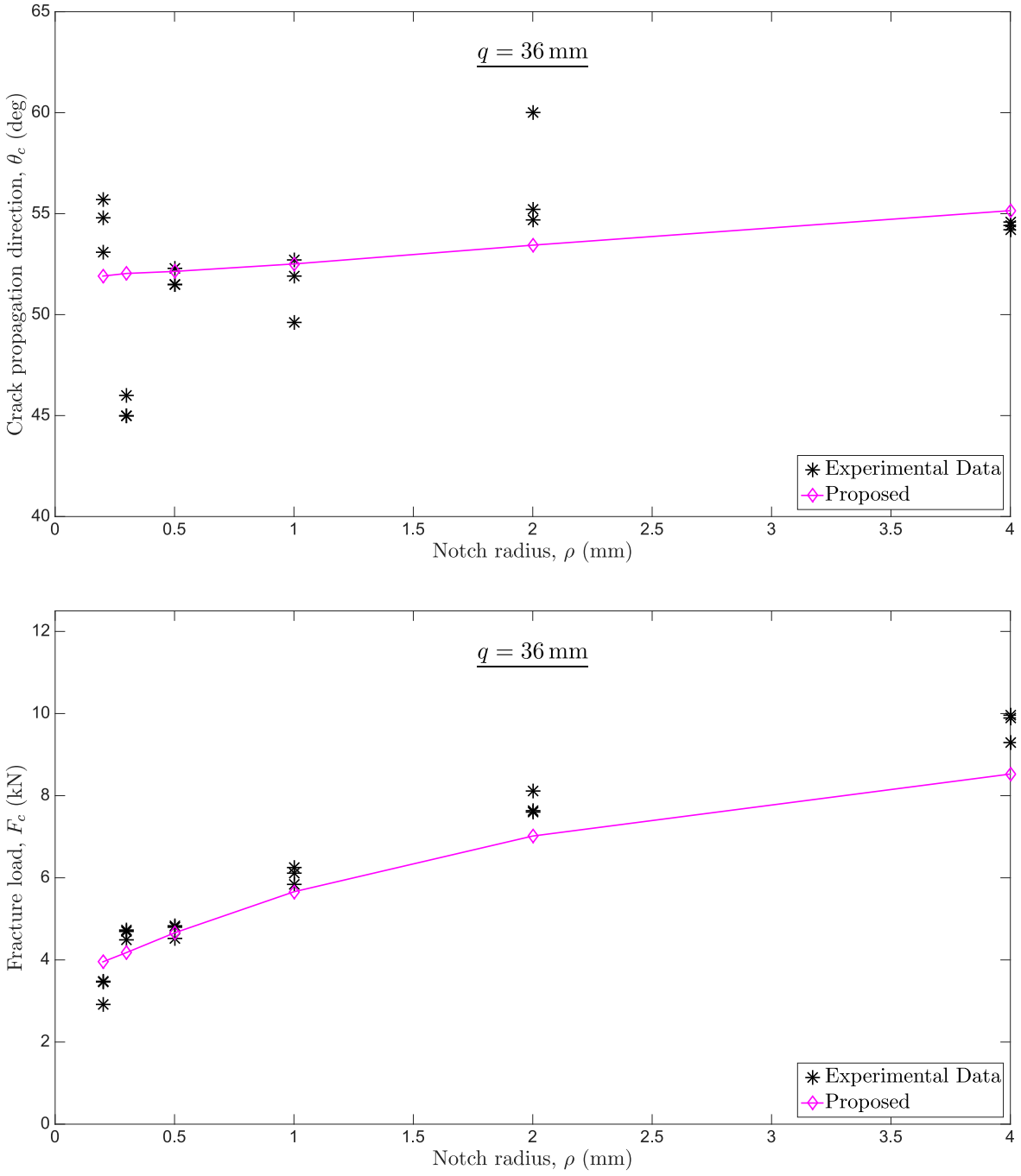


Figure 3.25. Crack initiation angles and fracture loads for U-notched specimens with loading at $q = 36$ mm.

Predictions made by the criterion were compared to experimental data for different notch geometries contained in PMMA specimens; V-notches, U-notches and circular notches. Pure Mode-I and mixed-mode fracture predictions for the V-notched specimens correlated quite well with the experimental data. Unfortunately, pure Mode-II fracture load predictions for wedge angles greater than $2\beta = 20^\circ$ were not as accurate. This can be attributed to a loss of uniqueness and the fact that the perturbing hole cannot breach the domain boundary. There

was very good agreement between the predicted crack initiation angles and their experimental counterparts in the U-notched specimens, which were subject to three-point bend tests and eccentric loading. Fracture load predictions were consistent, if not a little conservative, with the experimental data. The predictions for the circular notched specimens subjected to tensile loading were of a somewhat conservative nature. This was expected as the notch sizes were non-macroscopic. In general, predictions made by the proposed criterion were shown to be very competitive when compared with those of existing and well-established criteria. An area that requires further study is the pure Mode-II predictive capability of the proposed criterion for V-notched specimens. Further evidence showcasing the ability of the proposed criterion would provide significant support and recognition of the theory.

Chapter 4

Automatic Sharp Notch and Fracture Detection

4.1 Introduction

There are three basic and distinct modes of deformation as discussed in the previous chapter: Mode-I, normal separation; Mode-II, shear sliding normal to the crack front; and Mode-III, shear sliding parallel to the crack front. In order to simplify matters, Mode-III is not considered. Let a polar coordinate system (r, θ) be centred at the tip of a crack contained in a two-dimensional, homogeneous and isotropic linear elastic body. Then, the displacement around the crack tip with traction-free surfaces is given by

$$\mathbf{u} = \sum_{j=I}^{II} c_j r^{\frac{1}{2}} \mathbf{f}^j(\theta), \tag{4.1.1}$$

where c_j is a stress intensity factor and $\mathbf{f}^j$ is a smooth angular function. Observe that the exponent is 0.5 for both terms in the displacement solution representing Mode-I and Mode-II. Stress, on the other hand, has an inverse square root relationship with r. Therefore, as $r \to 0$, the stress becomes infinitely large. At $r = 0$, a singularity exists, which is clearly not a physical phenomenon (refer to Section 3.2.3). This is a result of operating in a linear elastic continuum framework and assuming a sharp crack tip.

Once again, a crack is an instance of a V-notch. Modes of deformation are similarly defined for V-notch surfaces. As would be expected, the displacement around a V-notch tip

with traction-free surfaces has a similar form to that of a crack, in that

$$\mathbf{u} = \sum_{j=I}^{II} c_j^{\eta} r^{\eta_j} \mathbf{g}^j(\theta), \tag{4.1.2}$$

where c_j^{η} is a generalised stress intensity factor dependent on the specimen geometry and external loading. The exponent η_j is an eigenvalue and $\mathbf{g}^j$ is the corresponding eigenfunction, which is a smooth angular function. Both are dependent on the local specimen geometry and boundary conditions. What is of real importance here is that the value of the exponents is no longer 0.5, but obtained from the following characteristic equations:

$$\sin 2\eta_I\alpha + \eta_I \sin 2\alpha = 0 \text{ and} \tag{4.1.3a}$$

$$\sin 2\eta_{II}\alpha - \eta_{II} \sin 2\alpha = 0, \tag{4.1.3b}$$

in which $\alpha = \pi - \beta$ and β is the semi-wedge angle. It is worth noting that η_I and η_{II} are real roots of the equations.

So far, the discussion has been focused on a two-dimensional idealisation. The extension of the displacement solution to three-dimensions is neither straightforward nor simple for a straight V-notch. It is usual to decompose the displacement into a number of terms that depend on the distance along the V-notch tip from a corner point, i.e. the intersection of the notch tip and free-surface. Let r represent the distance from a point on the notch tip acting in the plane normal to the notch tip at that point. Also, assume that there is a spherical coordinate system (ρ, θ, ϕ) centred at each corner point. Then, the displacement solution has terms of the type (derived from [44]):

i. r^{ν_1} in the vicinity of the notch tip interior, i.e. exclusive of the ends;

ii. ρ^{ν_2} in the vicinity of a corner point; and

iii. $\rho^{\nu_3}(\sin\phi)^{\nu_4}$ in the vicinity where the notch tip approaches a corner point.

Here, ν_i is a real parameter and of interest is $\nu_i < 1$ for $i = 1, 2, 3, 4$. More recently, Apel et al. in [45] argued that there is only one type, namely ρ^{ν}, to deal with. For any given point on the V-notch tip, which coincides with the origin of a spherical coordinate system, they expressed the solution as

$$\mathbf{u} = \sum_j c_j^{\nu} \rho^{\nu_j} \mathbf{h}^j(\theta, \phi). \tag{4.1.4}$$

As before, c_j^{ν} is the generalised stress intensity factor. However, ν_j and $\mathbf{h}^j$ (analytic in θ and ϕ) depend on the distance from the corner point. Basically, the values of the parameters at the ends vary from those in the interior of the notch tip. Additionally, and very importantly, the eigenvalues and eigenfunctions are dependent on the local geometry and boundary conditions only in the vicinity of the notch tip interior and the corner. The eigenvalues are no longer obtained from transcendental equations as in the two-dimensional case.

Assuming a cylindrical coordinate system (r, θ, z) exists in the interior of a notch tip that is aligned with the the z-axis, a splitting of the second-order elliptic elasticity operator to accommodate a separation of variables with respect to z and (r, θ) can be performed. Then, the displacement can be expressed in terms of local 2-d solutions and so-called shadow functions [46]. In this formulation, the edge stress intensity functions depend on z only. The eigenvalues and eigenfunctions, dependent on r and θ, are the solution of differential equations.

There are two prominent approaches to determine the value of the eigenvalues and eigenfunctions in (**4.1.4**) for a corner point. Both cast the problem into a lower dimension, which is then solved numerically. In the first, a Mellin transform is employed to produce a one-dimensional boundary integral equation. This method suffers from the fact that it requires complicated prior analytical calculations for every problem considered [47]. The second and more robust approach defines a two-dimensional problem with respect to the two angular variables θ and ϕ on a unit sphere centred at a corner point. This is indeed two-dimensional as the radial component is not taken into account. A modified quadratic variational boundary eigenvalue problem can then be posed [48]. Finite element discretisation produces a quadratic matrix eigenvalue problem, which has been tackled by the Skew-Hamiltonian Implicitly Restarted Arnoldi (SHIRA) algorithm in [49]. The second approach is just as valid for points on the interior of the notch tip.

The behaviour of the solution near a V-notch tip has a bearing on numerical schemes. In the FEM, convergence of the approximation of $\mathbf{u}$ in a quasi-uniform mesh is negatively impacted [50]. Strategies have, therefore, been developed to ensure that the solution is resolved with sufficient accuracy without resorting to a ridiculously fine mesh. A particularly successful method for two-dimensions involves augmenting the space of test and trial functions. The author is unaware of any literature attempting this procedure for sharp notches, other than cracks, in three-dimensions. It is not clear how to deal with the part of the notch tip between the corner point and the interior. In the case of a crack, this is usually avoided by simply hard-coding the eigenvalue and eigenfunction from the two-dimensional solution

across the length of the notch tip. An adaptive scheme that involves local mesh refinement (h-version) and/or polynomial degree increase (p-version) driven by an error estimator is another possibility. In truth, the notion of error-driven adaptivity can only ever be justified when evolution of the quantity of interest is not known a priori. Re-computing and solving the global set of algebraic equations in the FEM is quite costly. Also, this kind of adaptivity does not necessarily guarantee a desirable convergence.

The purpose of this chapter is twofold. Firstly, this chapter proposes feature detection and intelligent mesh design to address the issue of steep gradients near sharp notch tips in the FEM. Identifying a suitable data structure for finite element mesh storage and navigation is often the first course of action in the development of mesh-related algorithms. With this approach, one need not be concerned with the details of mesh representation and can simply apply the functionality available in a library like MOAB [51] for instance. After all, connectivity queries can be performed rather efficiently and there is a whole host of other functionality that is usually on offer with data structure libraries. However, there is an attached cost to the construction and storage of the incidence information in terms of both time and memory. What one gains in terms of speed-up in the development of algorithms, one might lose in the processing time of the finished product. It stands to reason that if possible, and this is not always the case, a data structure specifically designed for the task at hand should be considered. To that end, efficient algorithms for surface extraction from an original quasi-uniform tetrahedral mesh, sharp notch tip identification and size specification for mesh re-generation in TetGen [52] are presented. The concept of hashing is exploited in the development of these routines. Hash tables in the C++ standard library, available from C++11 [53], are realised in the form of unordered associative containers. Fully working source code utilising these containers is disclosed without sacrificing conciseness. Secondly, this chapter provides a procedure for crack onset prediction at a sharp notch in three-dimensions. One must keep in mind that fracture mechanics is very much a two-dimensional theory [54]. If mode-III fracture is discounted, then the non-local criteria considered in the previous chapter can be applied in a three-dimensional setting. An implementation of the procedure is pursued in FreeFem++ as proof of concept. While it would have been ideal for the implementation to be made in MoFEM, this is not possible as the displacement solution cannot be interrogated at any given point in the mesh unless selected prior to analysis.

This chapter is organised as follows. In Section 4.2, notions of discrete geometry relevant to the definition of a finite element mesh are reviewed. This provides the requisite knowledge for Section 4.3, where algorithms are developed to detect sharp notch tips and to re-generate the mesh with local refinement. This is followed in Section 4.4 with a general procedure for

application of non-local criteria to predict crack onset at a sharp notch tip. Finally, a summary of achievements as well as direction of potential future work is contained in Section 4.5.

4.2 Problem background

4.2.1 Linear and affine subspaces

Let L denote a linear subspace, which is a subset of $\mathbb{R}^d$ that is closed under the operations of addition and multiplication with real numbers. Interestingly, if $\mathbf{w} \in L$, then $-1\mathbf{w} = -\mathbf{w}$ and $\mathbf{w} + (-\mathbf{w}) = \mathbf{0}$, which leads to the discovery that the origin vector $\mathbf{0}$ is contained in any linear subspace. Geometrically, the impact of this is profound. For example, the linear subspaces of $\mathbb{R}^2$ are the origin, lines passing through the origin and $\mathbb{R}^2$ itself. In $\mathbb{R}^3$, the belonging subspaces are the origin, lines and planes passing through the origin as well as $\mathbb{R}^3$. It can be somewhat limiting in the representation of physics if there exists a distinct origin. For example, the study of kinematics and dynamics benefits from a more general setting.

An affine subspace, $M \subseteq \mathbb{R}^d$, is defined either as $M = \emptyset$ or as a translation of a linear subspace. The latter is encapsulated in

$$M = L + \mathbf{v} = \{\mathbf{w} + \mathbf{v} \,|\, \mathbf{w} \in L\}, \tag{4.2.1}$$

where $\mathbf{v} \in \mathbb{R}^d$. It is easy to check that this is true for every affine subspace, given $\mathbf{m} \in M$, then $L = M - \mathbf{m} = \{\mathbf{v} - \mathbf{m} \,|\, \mathbf{v} \in M\}$ is a linear subspace. In the case of intersecting affine subspaces M_i $(i \in I)$ of $\mathbb{R}^d$, an affine subspace is returned. This follows from the fact that there must be a common element, say $\mathbf{m}$, such that

$$\bigcap_{i \in I} M_i - \mathbf{m} = \bigcap_{i \in I} L_i, \tag{4.2.2}$$

where L_i $(i \in I)$ are linear subspaces.

Let $W \subseteq \mathbb{R}^d$, then the intersection of all affine subspaces containing W is said to be the affine hull of W, denoted $\mathrm{aff}(W)$. Clearly, the intersection is the smallest affine subspace. Note that the intersection is non-empty as W contains $\mathbb{R}^d$.

An affine analogue to a linear combination is known as an affine combination. To define an affine combination, it is first necessary to write the linear combination corresponding to points $\mathbf{p}_1, \ldots, \mathbf{p}_n \in \mathbb{R}^d$. This involves the selection of a reference point, say $\mathbf{p}_n$, then

$\iota_1(\mathbf{p}_1 - \mathbf{p}_n) + \cdots + \iota_n(\mathbf{p}_n - \mathbf{p}_n)$, where $\iota_1, \ldots, \iota_n \in \mathbb{R}$. A translation back by $\mathbf{p}_n$ results in $\iota_1\mathbf{p}_1 + \cdots + \iota_{n-1}\mathbf{p}_{n-1} + (1 - \iota_1 - \cdots - \iota_{n-1})\mathbf{p}_n$. Therefore, an affine combination is

$$\sum_{i=1}^{n} \kappa_i \mathbf{p}_i, \text{ where } \kappa_1, \ldots, \kappa_n \in \mathbb{R} \text{ with } \sum_{i=1}^{n} \kappa_i = 1. \tag{4.2.3}$$

It follows that the set of all affine combinations of elements of W is equivalent to $\text{aff}(W)$.

The points $\mathbf{p}_1, \ldots, \mathbf{p}_n$ are said to be affine dependent if

$$\sum_{i=1}^{n} \kappa_i \mathbf{p}_i = \mathbf{0}, \text{ where } \kappa_1, \ldots, \kappa_n \in \mathbb{R} \text{ with } \sum_{i=1}^{n} \kappa_i = 0. \tag{4.2.4}$$

There has to be at least one non-zero scalar coefficient for there to be a non-trivial affine dependence, otherwise the points are affine independent. Affine independence, in fact, is nothing more than the linear independence of vectors $\mathbf{p}_1 - \mathbf{p}_n, \mathbf{p}_2 - \mathbf{p}_n, \ldots, \mathbf{p}_{n-1} - \mathbf{p}_n$, where $\mathbf{p}_n$ can be replaced by any other point in the set. Notice that in an affine combination, the scalar coefficients sum up to 1, whereas for affine dependence, the scalar coefficients sum up to 0. Think of affine dependence as the ability to express one point as a combination of the others. No such relation exists in affine independence. For example, three points that do not lie on the same line are affinely independent.

Algebraically, a criterion can be formulated to determine if the points $\mathbf{p}_1, \ldots, \mathbf{p}_{d+1} \in \mathbb{R}^d$ are affinely independent. This involves taking the determinant of a $d \times d$ matrix, say $\mathbf{A}$, where the i-th column represents the vector $\mathbf{p}_i - \mathbf{p}_{d+1}$ for $i = 1, \ldots, d$. Then, affine independence is given by $\det(\mathbf{A}) \neq 0$, which corresponds to the linear independence of the columns.

Affine subspaces are classified according to their dimension, denoted by $\dim(M)$, which is equal to -1 when $M = \emptyset$ or assumes the value of the dimension of the linear subspace $\dim(L)$ otherwise. An affine subspace H of $\mathbb{R}^d$ with a $(d-1)$-dimension is called a hyperplane, and is described by

$$H = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{a} \cdot \mathbf{x} = b\}, \tag{4.2.5}$$

where $\mathbf{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$. For $d = 1$, the hyperplane is a single point. For $d = 2$, the hyperplane represents a line. A plane is a hyperplane with $d = 3$. More generally, it can be seen that a hyperplane partitions $\mathbb{R}^d$ into two (closed) half-spaces,

$$H_+ = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{a} \cdot \mathbf{x} \geq b\} \text{ and} \tag{4.2.6a}$$

$$H_- = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{a} \cdot \mathbf{x} \leq b\}. \tag{4.2.6b}$$

Obviously, $H_+ \cap H_- = H$ is the boundary; i.e. the hyperplane itself. Note that the assignment of positive and negative to the half-spaces is not inferred from the hyperplane but relies on additional orientation information.

4.2.2 Convexity

The notion of convexity in terms of geometry underpins much of this work. It stands to reason that a mathematically rigorous definition of this concept should be provided. A subset K of $\mathbb{R}^d$ is said to be convex if for any two points $\mathbf{p}, \mathbf{q} \in K$ there is a further point $\gamma\mathbf{p} + (1-\gamma)\mathbf{q} \in K$, where $\gamma \in \mathbb{R}$ and $0 \leq \gamma \leq 1$. Note that an empty set is considered convex, although trivial. Other examples are a point, a line segment and $\mathbb{R}^d$. Clearly, the intersection of a family of convex sets is convex.

Let $U \subseteq \mathbb{R}^d$, then the convex hull of U, denoted $\text{conv}(U)$, is the intersection of all convex sets containing U. This, in fact, is the smallest set containing U. Additionally, a convex hull consists of the set of all convex combinations of points of U. A convex combination of points $\mathbf{p}_1, \ldots, \mathbf{p}_n \in U$ is the linear combination

$$\sum_{i=1}^{n} \varphi_i \mathbf{p}_i, \text{ where } \varphi_1, \ldots, \varphi_n \in \mathbb{R} \text{ and } \varphi_1, \ldots, \varphi_n \geq 0 \text{ with } \sum_{i=1}^{n} \varphi_i = 1. \tag{4.2.7}$$

Consider two sets of convex combinations, say $\sum_i \zeta_i \mathbf{p}_i$ and $\sum_i \psi_i \mathbf{q}_i$, where $\mathbf{p}_i, \mathbf{q}_i \in U$. By definition,

$$\gamma \sum_i \zeta_i \mathbf{p}_i + (1-\gamma) \sum_i \psi_i \mathbf{q}_i \text{ with } 0 \leq \gamma \leq 1, \tag{4.2.8}$$

is convex. It follows that the set of convex combinations of points of U is indeed a convex set. This is the smallest convex set containing U. Thus, the set of all convex combinations of points of U is equivalent to $\text{conv}(U)$.

4.2.3 Tetrahedron

A tetrahedron is classified as a convex polytope. There are two mathematically equivalent definitions based on the construction details of the polytope. A $\mathcal{V}$-polytope is the convex hull of a finite set of points in $\mathbb{R}^d$, while a $\mathcal{H}$-polytope is the bounded intersection of a finite set of closed half-spaces. The proof of mathematical equivalence is omitted, as only the $\mathcal{V}$-polytope is relevant to the continuing discussion; a complete proof can be found in [55].

A particularly important example of a $\mathcal{V}$-polytope is the d-simplex; a point is a 0-simplex, a line segment is a 1-simplex, a triangle is a 2-simplex, and a tetrahedron is a 3-simplex. Specifically, a d-simplex is the convex hull of $d + 1$ affinely independent points $\mathbf{v}_1, \ldots, \mathbf{v}_{d+1}$ of $\mathbb{R}^d$, that is

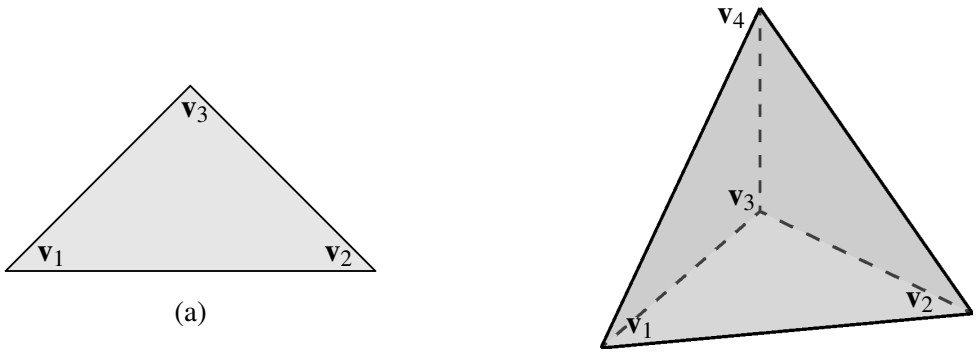
$$\sum_{i=1}^{d+1} \varphi_i \mathbf{v}_i, \text{ where } \varphi_1, \ldots, \varphi_{d+1} \in \mathbb{R} \text{ and } \varphi_1, \ldots, \varphi_{d+1} \geq 0 \text{ with } \sum_{i=1}^{d+1} \varphi_i = 1. \tag{4.2.9}$$

Note the extremal nature of points; any point represents an end of a line segment contained in the convex polytope.

Let a subset P of $\mathbb{R}^d$ be a d-dimensional convex polytope. An intersection of a hyperplane with P, $H \cap P$, which is not empty and where P is entirely contained in one of the two closed half-spaces is called a face of P. It follows that every point on the boundary of P must coincide with a hyperplane. Furthermore, a face of a face is, in fact, a face of P. The naming convention of faces is dimension-based: a 0-face is a vertex; a 1-face is an edge; and a $(d - 1)$-face is a facet. For completeness, a $(d - 2)$-face is a ridge; the edge label takes precedence in $\mathbb{R}^3$. Additionally, there are what is known as improper faces of which the empty set $\emptyset$ is one with a dimension of -1 and the other is P itself, which is said to be the d-face of P. For example, a tetrahedron has the following faces: the empty set; 4 vertices; 6 edges; 4 facets; and the tetrahedron itself.

Let $\mathbf{v}_1, \ldots, \mathbf{v}_{d+1}$ represent the vertices of a d-simplex S_d. It is possible to assign an order to the vertices. Doing so provides an orientation to S_d, which is in the form of a sign. Therefore, an oriented simplex can be either positive or negative. An ordered list of vertices is denoted as $[\mathbf{v}_1, \ldots, \mathbf{v}_{d+1}]$. A 0-simplex is taken as positive, a 1-simplex is positive if the edge points to $\mathbf{v}_2$ from $\mathbf{v}_1$ and the opposite is negative, a 2-simplex is positive if the vertices are ordered counter-clockwise and negative if clockwise, while a 3-simplex is positive if the ordering follows the so-called right-hand screw rule and negative if it corresponds to what would be the left-hand screw rule. For further clarification, see Figure 4.1.

There are $(d + 1)!$ possible vertex permutations of an ordered list representing a d-simplex. An exchange of the position of two vertices is known as a transposition. If a single transposition is made, a negative orientation results. More generally, an odd number of transpositions results in a negatively oriented simplex. However, the orientation remains unchanged by an even number of transpositions. For example, consider a 2-simplex $S_2 = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$. A single transposition, say $[\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_2]$, produces a clockwise ordering of the vertices. Whereas, two transpositions, say $[\mathbf{v}_3, \mathbf{v}_1, \mathbf{v}_2]$, maintains the counter-clockwise order of the vertices.



(a) (b)

Figure 4.1. Vertex order for positive orientation of (a) 2-simplex and (b) 3-simplex.

Orientation of a 2-simplex with vertices $\mathbf{v}_i \in \mathbb{R}^2$ for $i = 1, 2, 3$ can be obtained from the sign of the determinant

$$\begin{vmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ 1 & 1 & 1 \end{vmatrix}. \tag{4.2.10}$$

There are three possible outcomes: a positive value for a counter-clockwise order, a negative value and a zero value. While the sign of the first two informs the orientation, the zero-value indicates that the points are collinear. This follows from the cross product

$$(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_2), \tag{4.2.11}$$

which provides the positioning of $\mathbf{v}_3$ with respect to the line passing through $\mathbf{v}_1$ and $\mathbf{v}_2$.

In the case of a 3-simplex with vertices $\mathbf{v}_i \in \mathbb{R}^3$ for $i = 1, 2, 3, 4$, the formula for the 2-simplex is easily extendable:

$$\begin{vmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 \\ 1 & 1 & 1 & 1 \end{vmatrix}. \tag{4.2.12}$$

As before, a zero-value is a reference to the degenerate nature of the simplex.

An orientation is induced on the boundary in an oriented d-simplex S_d. The boundary, denoted ∂S_d, is comprised of a set of $(d + 1)$ facets that are $(d - 1)$-simplices, which is described by

$$\partial S_d = \sum_{i=0}^{d} (-1)^i \, [\mathbf{v}_1, \ldots, \bar{\mathbf{v}}_{i+1}, \ldots, \mathbf{v}_{d+1}], \tag{4.2.13}$$

where the bar signifies omission of the vertex from the ordered list. In the case of a 0-simplex, $\partial S_0 = \emptyset$. In the case of a 1-simplex and a 2-simplex, $\partial S_1 = [\mathbf{v}_2] - [\mathbf{v}_1]$ and

$\partial S_2 = [\mathbf{v}_2, \mathbf{v}_3] - [\mathbf{v}_1, \mathbf{v}_3] + [\mathbf{v}_1, \mathbf{v}_2]$, respectively. Most relevant to this work is the boundary of a 3-simplex, which is $\partial S_3 = [\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4] - [\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4] + [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4] - [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$. Given this information, it is possible to determine the outward unit normal vectors to the $(d-1)$-simplices belonging to the boundary ∂S_d in a consistent manner. For example, the outward unit normal vector $\mathbf{n} \in \mathbb{R}^3$ to the boundary simplex represented by $[\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4]$ is given by

$$\mathbf{n} = -\frac{(\mathbf{v}_3 - \mathbf{v}_1) \times (\mathbf{v}_4 - \mathbf{v}_1)}{\|(\mathbf{v}_3 - \mathbf{v}_1) \times (\mathbf{v}_4 - \mathbf{v}_1)\|}. \tag{4.2.14}$$

Let $\mathbf{n}_1 \in \mathbb{R}^3$ and $\mathbf{n}_2 \in \mathbb{R}^3$ be outward unit normal vectors to two facets that are of the 2-simplex variety. If these facets share a common edge, then the unit normal vector at that edge $\mathbf{n}_e \in \mathbb{R}^3$ is simply

$$\mathbf{n}_e = \frac{\mathbf{n}_1 + \mathbf{n}_2}{\|\mathbf{n}_1 + \mathbf{n}_2\|}. \tag{4.2.15}$$

The dihedral angle $\theta_e \in \mathbb{R}$ separating these facets is the supplement of the angle between $\mathbf{n}_1$ and $\mathbf{n}_2$,

$$\theta_e = \pi - \arccos(\mathbf{n}_1 \cdot \mathbf{n}_2). \tag{4.2.16}$$

Finally, a tetrahedral finite element mesh is composed of a finite collection of simplices, which intersect at their proper faces. It is expected that a mesh adequately represents the body domain in this work.

4.3 Sharp notch tip detection

4.3.1 Type declarations

Type declarations shown in Listing 4.1 are standard in this section. The **typedef** keyword is used here to assign an alias name to an existing data type.

```
typedef std::tuple<int,int,int> i3tuple;
typedef std::tuple<int,int,double> i2dtuple;
typedef std::tuple<int,int,int,int> i4tuple;
```

Listing 4.1: Data type declarations.

Note that a **tuple** can hold a finite collection of elements not necessarily of the same type. Furthermore, the elements can be accessed by their stored order in the **tuple**. Data types, keywords and function names are highlighted bold for purposes of clarity.

4.3.2 Input

Finite element (tetrahedral) mesh data is taken from files organised according to the legacy Visualization Toolkit (VTK) format [56]. Coordinates of vertices are read sequentially and each component is stored as an element of a **vector** (sequence container akin to a dynamic array) with type **double**. A global numbering system for the vertices follows naturally from this; the first index is 0 as is usual in computing. Tetrahedra are then read as sets of four vertex indices, which are stored as elements of a **vector** with type **i4tuple**. No connectivity information with regards to the tetrahedra is available in a VTK file. Consequently, each tetrahedron is treated as a separate entity. It is worth noting that the vertices of the tetrahedra in the VTK file are arranged in an order supporting a positive orientation of the 3-simplex.

Illustrative examples visualised in ParaView [57] demonstrate the effectiveness of the C++ functions developed in this section. The functions are applied to a tetrahedral mesh representative of: (i) a double-edge V-notched rectangular cuboid specimen with $2\beta = 1^\circ$ wedge angles; (ii) a double edge V-notched rectangular cuboid specimen with $2\beta = 40^\circ$ wedge angles; (iii) an L-shaped specimen; and (iv) a graphite fuel brick taken from an example in SOLFEC. The V-notched specimens are based on the design for the biaxial loading device in [40]. However, the length of the specimen is altered to 100 mm and the thickness is no longer 5 mm but 20 mm instead. The L-shaped specimen dimensions of interest are those in [58] employed to verify a constitutive model for concrete (see Figure 4.2).

4.3.3 Extraction of mesh boundary facets

Facets incident on the boundary of a mesh belong to only one tetrahedron. In other words, the set of vertex indices constituting a facet should occur only once in the set of all facets of the disjoint tetrahedra that form the mesh.

The most obvious technique to extract the required information begins with an arrangement of the vertex indices of every facet in a descending or ascending order (discarding orientation information). Then, an expensive search can be conducted to identify facets with the same first vertex index. Comparison of the remaining indices of every facet with the same first vertex index yields the facets incident on the boundary. This is not the most efficient of methodologies due to the search and the fact a record of all the oriented 2-simplices as well as references to them are required.

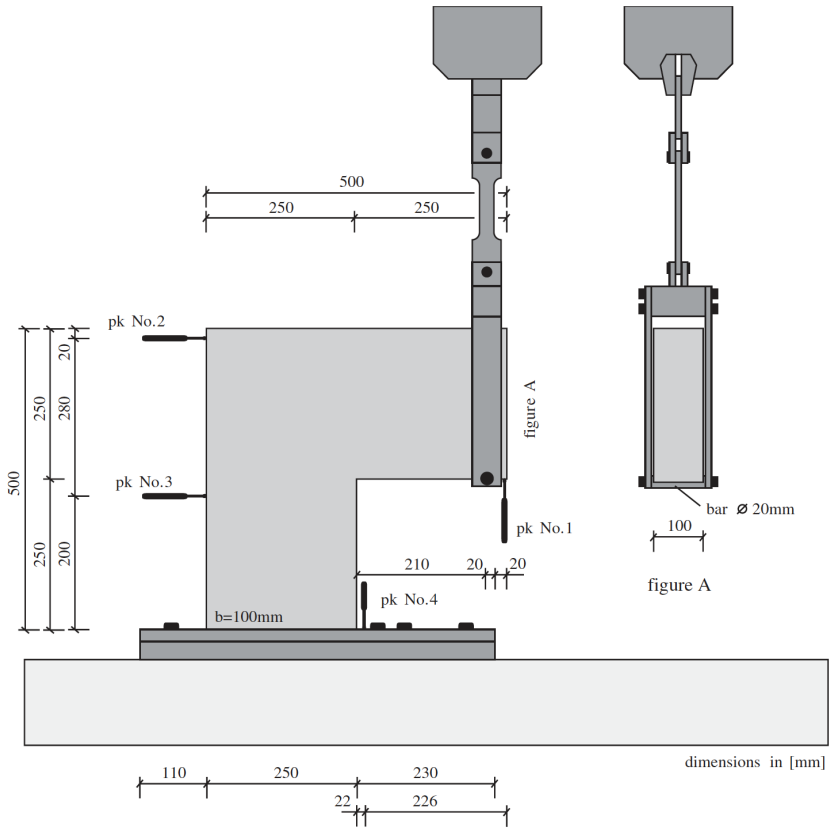


Figure 4.2. Experimental set-up for L-shaped specimens (reproduced from [59]).

A less memory-intensive and more streamlined strategy involves assigning a positive integer, unique to the combination of vertex indices, to each facet of every tetrahedron. The Cantor tripling function, refer to Appendix E, is particularly suited for this task. Then, the facets with the non-repeating assigned number represent the boundary of the mesh. Computationally, this can be accomplished efficiently through the use of an **unordered_map**. The assigned number is the key value and the set of three vertex indices of the corresponding facet is the mapped value. A hash function transforms the key into a table address, where the data is stored. An element composed of a key and mapped value pairing can only be inserted into an **unordered_map** if the key value does not already exist in the container. This provides the information necessary to identify facets with the same vertices.

Unordered associative containers are attractive as the insertion, search and erase operations all have $O(1)$ complexity. Care must be taken so as not to trigger a rehash operation, which involves mapping existing data to new locations. Otherwise, the complexity associated with insertion would no longer be constant. Therefore, the member function **reserve()** is something to consider, especially when dealing with a large dataset. All this does is prepare the hash table to receive the appropriate number of pairings based on an estimate of the

number of facets on the boundary of the mesh.

The function **BoundaryMeshFacets** in Listing 4.2 is an implementation of the strategy employing the Cantor tripling function.

```
void BoundaryMeshFacets(std::vector<i4tuple>& tets, std::vector<i3tuple>&
    bndFacets) {

  int i;
  long int uqVal;
  i3tuple facet;
  std::unordered_map<long int,i3tuple> uqVal_facet;

  // (i) Insert facets of each tetrahedron into an unordered map
  // (ii) Erase repeat instances leaving only facets incident on boundary
  for (auto& x: tets) {
    for (i = 1; i < 5; ++i) {
      GetBoundaryFacet(x, i, facet);
      uqVal = CantorTripling(std::get<0>(facet), std::get<1>(facet), std
   ::get<2>(facet));
      UpdateFacetsUnorderedMap(uqVal, facet, uqVal_facet);
    }
  }

  // Insert vertex indices of facets on the boundary into bndFacets
  for (auto& x: uqVal_facet)
    bndFacets.push_back( x.second );

  uqVal_facet.clear(); // Clear unordered map

}
```

Listing 4.2: Function for discovery of facets incident on the boundary of a body.

Note the data type of the variable **uqVal** in Listing 4.2. The number generated by the Cantor tripling function was, in fact, not in the range supported by **int** for the test models. For a greater number of tetrahedra in a mesh, **long long int** may well be necessary or maybe even a more suitable encoding function could be employed.

Observe that a further two functions are called in **BoundaryMeshFacets**. The first, **GetBoundaryFacet**, returns a facet incident on the boundary and is contained in Listing 4.3.

```
void GetBoundaryFacet(i4tuple& tet, const int perm, i3tuple& tri) {

  switch (perm) {
    case 1:
      tri = std::make_tuple(std::get<0>(tet), std::get<1>(tet), std::get
    <3>(tet));
      break;
    case 2:
      tri = std::make_tuple(std::get<1>(tet), std::get<2>(tet), std::get
    <3>(tet));
      break;
    case 3:
      tri = std::make_tuple(std::get<0>(tet), std::get<2>(tet), std::get
    <1>(tet));
      break;
    case 4:
      tri = std::make_tuple(std::get<0>(tet), std::get<3>(tet), std::get
    <2>(tet));
      break;
  }

}
```

Listing 4.3: Function to obtain the facet of a tetrahedron in the mesh.

Here, the 2-simplex is given with a positive orientation to ensure the outward nature of the unit normal. Meanwhile, the second function, **UpdateFacetsUnorderedMap**, recorded in Listing 4.4 updates the **unordered_map**.

```
void UpdateFacetsUnorderedMap(long int val, i3tuple& tri, std::
    unordered_map<long int,i3tuple>& tris) {

  if (tris.count(val) == 0)
    tris.insert( std::pair<long int,i3tuple>(val,tri) );
  else
    tris.erase(val);

}
```

Listing 4.4: Function to update the unordered map.

In truth, the procedure can be made part of input file reading, which would negate the

cost incurred by the additional loop over the tetrahedra. Refer to figure 4.3 for the results obtained using the **BoundaryMeshFacets** function.

4.3.4 Extraction of mesh boundary edges

Once the facets incident on the boundary have been identified, a procedure can then be developed to discover the repeated (shared) edges and save only one of them. An additional requirement is that the incident boundary facets are also found. As before, a positive integer can be assigned to each edge based on the combination of vertex indices. However, this time it is the Cantor pairing function that generates this number. Once again, an **unordered_map** provides the means for unique edge retrieval. Listing 4.5 represents a concise implementation of the procedure.

```
// i4tuple: (indx1,indx2,fac1,fac2)
void BoundaryMeshEdges(std::vector<i3tuple>& facets, std::vector<i4tuple
    >& edges) {

  int i = -1, j, k;
  long int uqVal;
  std::unordered_map<long int,int> edge_facet;

  // (i) Loop through each facet
  // (ii) Record edge index and insert into unordered map along with
    associated facets
  for (auto& x: facets) {

    ++i; // increment for facet numbering

    // 1st edge
    uqVal = CantorPairing(std::get<0>(x), std::get<1>(x));
    UpdateEdgesUnorderedMap(std::get<0>(x), std::get<1>(x), uqVal, i,
   edge_facet, edges);

    // 2nd edge
    uqVal = CantorPairing(std::get<0>(x), std::get<2>(x));
    UpdateEdgesUnorderedMap(std::get<0>(x), std::get<2>(x), uqVal, i,
   edge_facet, edges);

    // 3rd edge
    uqVal = CantorPairing(std::get<1>(x), std::get<2>(x));
```

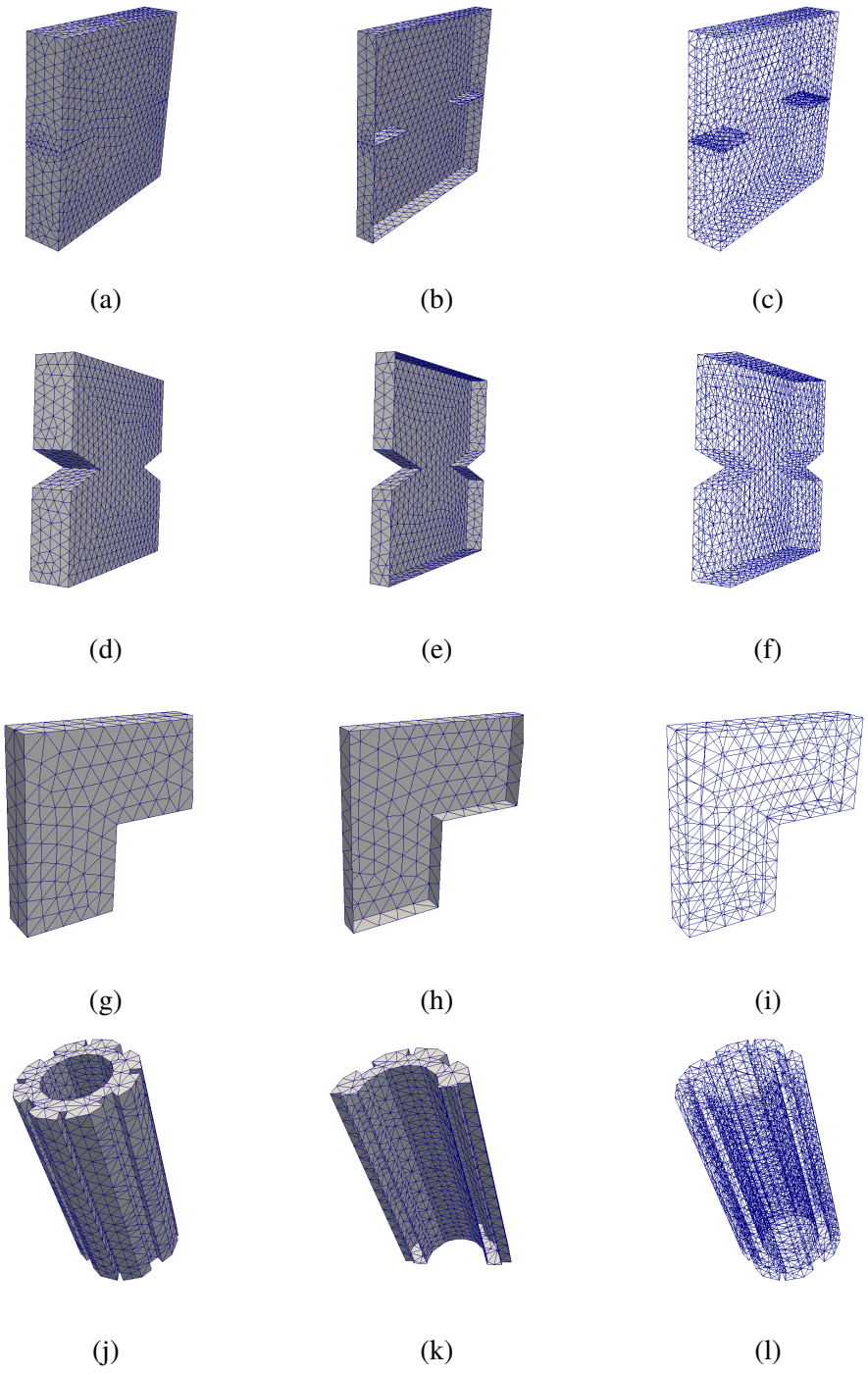


Figure 4.3. Mesh: (a) 9,765 tetrahedra, (d) 8,666 tetrahedra, (g) 1,150 tetrahedra, (j) 15,127 tetrahedra; Facets incident on boundary (split-view): (b), (e), (h), (k); Unique edges incident on boundary: (c), (f), (i), (l).

```
    UpdateEdgesUnorderedMap(std::get<1>(x), std::get<2>(x), uqVal, i,
    edge_facet, edges);

  }

  edge_facet.clear(); // Clear unordered map

}
```

Listing 4.5: Function for discovery of unique edges incident on the mesh boundary.

It is worth pointing out that the numbering of facets is based on their order in the corresponding **vector**. Thus, the index of a facet can be used directly to access the associated vertex data. The function **UpdateEdgesUnorderedMap**, as defined in Listing 4.6, serves to identify and store the unique edges by counting the instances of occurrence using the **unordered_map**.

```
void UpdateEdgesUnorderedMap(int indx1, int indx2, long int val, int n,
    std::unordered_map<long int,int>& edgFct, std::vector<i4tuple>& edgs)
    {

  if (edgFct.count(val) == 0)
    edgFct.insert( std::pair<long int,int>(val,n) );
  else
    edgs.push_back( i4tuple(indx1, indx2, n, edgFct[val]) );

}
```

Listing 4.6: Function to update unordered map and **vector** storing the edges.

Examples relating to the extraction of unique edges incident on the boundary of the mesh are presented in Figure 4.3.

4.3.5 Edge belonging to sharp notch tip

Consider an edge with two incident facets on the boundary and assign the order shown in Figure 4.4 to the vertices.

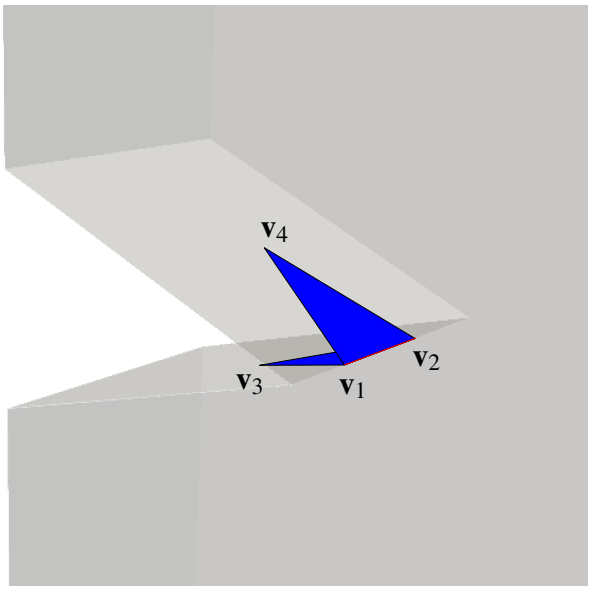


Figure 4.4. Order of vertices for sharp notch tip identification.

Then, an edge representing part or all of a sharp notch tip must satisfy $\mathbf{w}_1 \cdot (\mathbf{w}_2 \times \mathbf{w}_3) > 0$, where $\mathbf{w}_i = \mathbf{v}_{i+1} - \mathbf{v}_1$ for $i = 1, 2, 3$ and $\mathbf{v}_i \in \mathbb{R}^3$. This is not the whole story though, the exterior angle must also not exceed a user-set (scenario dependent) constant threshold value $\theta_{thresh} \in \mathbb{R}$:

$$\pi - \arccos(\mathbf{n}_1 \cdot \mathbf{n}_2) \leq \theta_{thresh}, \tag{4.3.1}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are outward unit normals of the incident facets. A poorly constructed mesh, i.e. one that is too coarse to adequately represent the domain of interest, may well contain artificial edges that could be mistakenly identified as belonging to a notch tip. The only remedy to this problem is careful design of the mesh in the first place. Edges belonging to sharp notches are highlighted in Figure 4.5, where $\theta_{thresh} = 110°$.

4.3.6 Edge connectivity on a sharp notch tip

It is often quite important to establish the connectivity of edges belonging to a sharp notch tip, especially if there is more than one notch to consider. For example, there may be an argument to discount a notch or notches from analysis. Say that a boundary condition is prescribed locally or even on part of a notch tip that might result in impact failure or prevent crack propagation. Without knowledge of the edges belonging to a notch tip and all the sharp notches present, it would not be possible to act on said information. However, this is not the only reason to establish connectivity. It is also to obtain the shape profile of the notch tip and

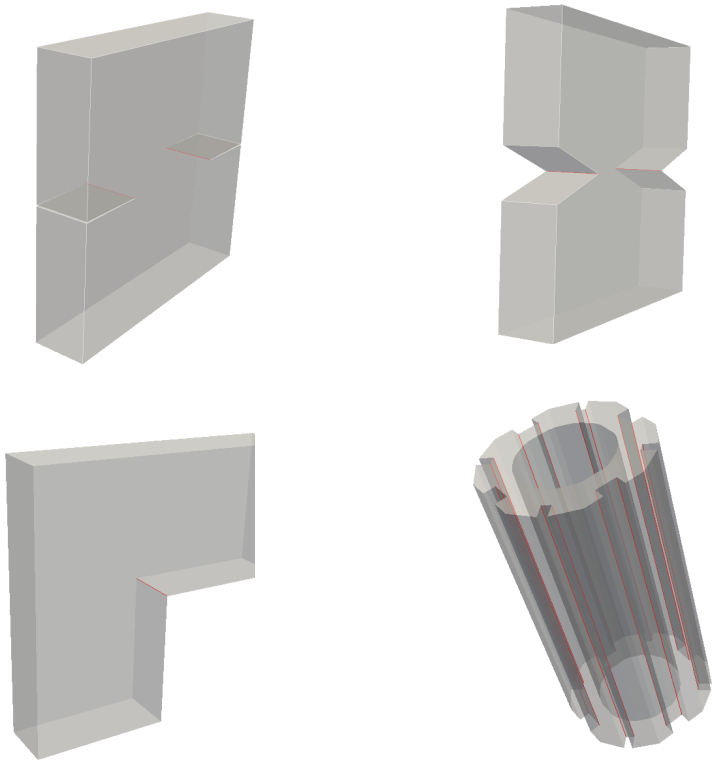


Figure 4.5. Examples of detected edges belonging to a sharp notch or notches.

to identify the vertices that represent the corners. Crack onset prediction can then be studied in a mesh-independent manner, i.e. without recourse to individual edges.

```
void EdgeConnectivity(std::vector<i4tuple>& notchEdges, std::vector< std
    ::vector< std::pair<int,int> > >& edgesConnected) {

  std::unordered_set<int> cornerPoints;
  std::vector< std::pair<int,int> > notch;
  std::unordered_multimap<int,int> edgeVerts;
  edgeVerts.reserve( 2*notchEdges.size() );

  // Store edge vertex indices in an unordered multimap container;
  // positive and negative orientation so that both vertices have key
   value position
  for (auto& x: notchEdges) {

    edgeVerts.insert( std::pair<int,int>(std::get<0>(x), std::get<1>(x))
   );
    edgeVerts.insert( std::pair<int,int>(std::get<1>(x), std::get<0>(x))
   );
```

```
  }

  // Determine corner edge points
  for (auto& x: notchEdges) {

    // 1st vertex on edge
    if (edgeVerts.count(std::get<0>(x)) == 1)
      cornerPoints.insert( std::get<0>(x) );

    // 2nd vertex on edge
    if (edgeVerts.count(std::get<1>(x)) == 1)
      cornerPoints.insert( std::get<1>(x) );

  }

  // Establish connectivity of edges for each notch
  for (auto& x: cornerPoints) {

    connectEdges(x, edgeVerts, notch);
    edgesConnected.push_back( notch ); // add notch to vector
    cornerPoints.erase( notch.back().second ); // erase opposite end
    notch.clear();

  }

  edgeVerts.clear(); // Clear unordered multimap

}
```

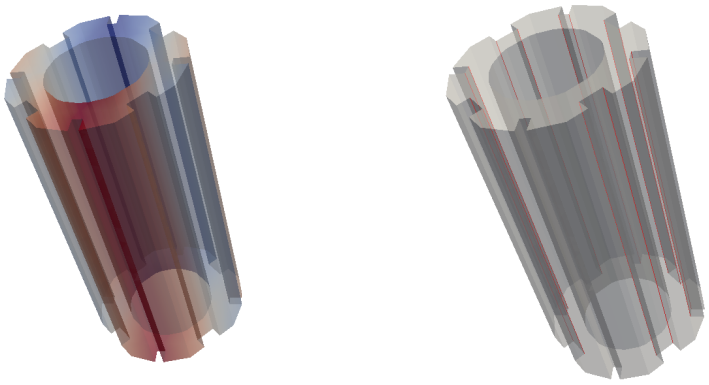
Listing 4.7: Function to return the connected edges on a single notch or notches.

Computationally, the procedure to connect the edges and identify separate notches is split into three parts as can be seen from Listing 4.7. The first part involves populating an **unordered_multimap** with the vertex indices of each edge belonging to a sharp notch. An **unordered_multimap** is somewhat similar to an unordered map but permits equivalent key values. Both orientations of an edge are accounted for in the container. Clearly, vertices that are corners are those with only one key value in the container. The second part of the procedure takes advantage of this to obtain their index values. Once the vertices at the corners are identified, connectivity can be established. This represents the third and final part of the procedure, which is contained in Listing 4.8.

```
void connectEdges(int cornerPt, i2uommp& edgVrt, std::vector< std::pair<
    int,int> >& edgConnec) {

  bool complete = false;
  int vrtIndx1 = cornerPt, vrtIndx2 = edgVrt.find(cornerPt)->second,
    VrtIndx3;
  edgConnec.push_back( std::pair<int,int>(cornerPt, vrtIndx2) );

  do {

    // Check if vrtIndx2 is not a corner point
    if (edgVrt.count(vrtIndx2) > 1) {
      // Loop through associated values of key that is vrtIndx2
      auto range = edgVrt.equal_range(vrtIndx2);
      for_each (range.first, range.second, [vrtIndx1, &VrtIndx3](i2uommp
    ::value_type& x) {
        // Check if value is not equal to vrtIndx1
        if (x.second != vrtIndx1)
          VrtIndx3 = x.second;
      } );

      // Store edge
      edgConnec.push_back( std::pair<int,int>(vrtIndx2, VrtIndx3) );

      // Update vrtIndx1 and vrtIndx2
      vrtIndx1 = vrtIndx2;
      vrtIndx2 = VrtIndx3;

    } else complete = true;

  } while (!complete);

}
```

Listing 4.8: Function to establish edge connectivity on a single notch.

Beginning from a corner, the mapped value can be employed to locate the connecting edge. It is the mapped value associated with the edge that forms the next search for the connecting edge and so on. This continues until a vertex at the opposite corner is discovered and the process is repeated for every sharp notch in the mesh. An example where sharp notches in a graphite fuel brick mesh are discounted from crack onset analysis due to imposed boundary conditions is shown in Figure 4.6.



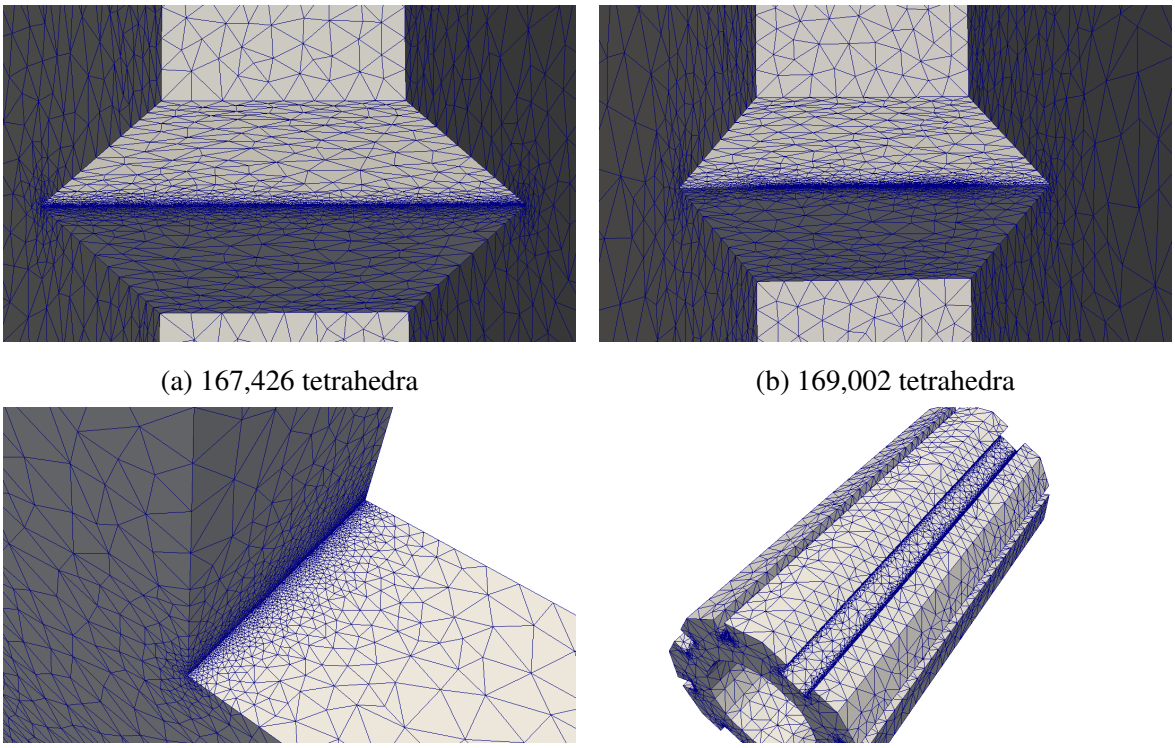
(a) (b)

Figure 4.6. (a) Prescribed boundary displacement and (b) discounted notch edges.

4.3.7 Mesh re-generation

TetGen can generate a quality boundary conforming tetrahedral mesh that takes into account a user-defined sizing function. Lengths of the edges incident at a vertex can be specified. This controls the size of the tetrahedra near the boundary. Accordingly, the local refinement desired can be achieved. From the previous functions, the vertices belonging to a sharp notch tip are known. It is then simply a matter of interacting with TetGen and the array **pointmtrlist** in particular. A function that demonstrates how to carry out this task is provided in Appendix F. The function is designed to receive sizing information at the corners separate to the rest of the sharp notch tip. This provides the option of non-uniform refinement across an entire notch tip. Note that the TetGen header file must be declared along with relevant header files of the C++ standard library to compile the source code. Finally, Figure 4.7 showcases TetGen's ability to produce user-defined local mesh refinement.

All the functions in this section, except for that which establishes edge connectivity, have been implemented in SOLFEC and can be called through the **TETRAHEDRALIZE** function.



(a) 167,426 tetrahedra (b) 169,002 tetrahedra

(c) 54,980 tetrahedra (d) 52,342 tetrahedra

Figure 4.7. Internal view of (a) uniform and (b) non-uniform mesh refinement at a notch edge of the double edge V-notched specimen with $2\beta = 40°$; Uniform mesh refinement at (c) notch of L-shaped specimen and (d) notches of graphite fuel brick.

4.4 Crack onset prediction

4.4.1 Procedure

A general procedure for the application of the non-local criteria studied in the previous chapter for crack onset prediction at a sharp notch tip consists of the following steps:

i. Approximate the solution of the boundary value problem of linear elastostatics using the FEM.

ii. Select a number of points $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n \in \mathbb{R}^3$ on the notch tip, where $\mathbf{p}_1$ and $\mathbf{p}_n$ are reserved for the corners (see Figure 4.8). A slight modification to the coordinates of the corner points is required in order to negate the potential impact of numerical precision. This can be accomplished as follows:

$$\mathbf{p}_1 + \chi \frac{\mathbf{p}_n - \mathbf{p}_1}{\|\mathbf{p}_n - \mathbf{p}_1\|} \text{ and } \mathbf{p}_n + \chi \frac{\mathbf{p}_1 - \mathbf{p}_n}{\|\mathbf{p}_1 - \mathbf{p}_n\|}, \tag{4.4.1}$$

where χ is a user-defined constant. The value of the constant should be greater than the machine epsilon to allow for propagating error in the processing of geometric queries (resulting from calculations involving the corner points on the boundary).

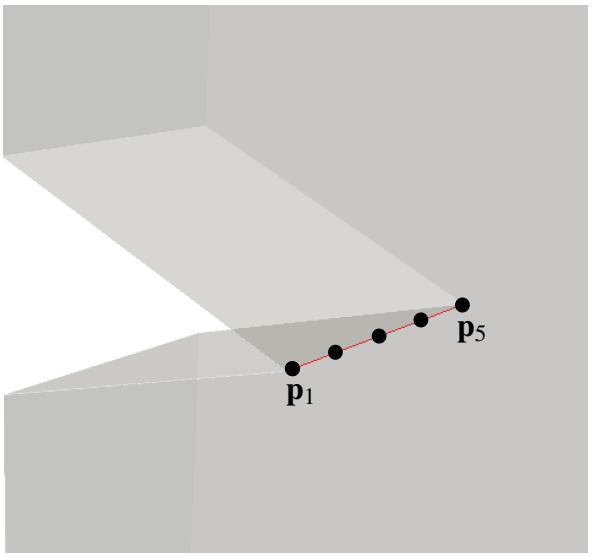


Figure 4.8. Points on a sharp notch tip example.

iii. Assign an orthonormal coordinate system to each point $\mathbf{p}_i$, where the point itself acts

as the origin. Let the basis vectors have the form

$$\tilde{\mathbf{e}}_1 = \frac{\mathbf{p}_n - \mathbf{p}_1}{\|\mathbf{p}_n - \mathbf{p}_1\|}, \tilde{\mathbf{e}}_2^{(i)} = -\mathbf{n}_e^{(i)} \text{ and } \tilde{\mathbf{e}}_3^{(i)} = -\frac{\tilde{\mathbf{e}}_1 \times \tilde{\mathbf{e}}_2^{(i)}}{\|\tilde{\mathbf{e}}_1 \times \tilde{\mathbf{e}}_2^{(i)}\|}. \tag{4.4.2}$$

Note that the edge normal $\mathbf{n}_e^{(i)}$ is specific to each point, this is done for the sake of generality. Most likely, the edge normal would be constant across the notch tip.

iv. Compute the reference point $\mathbf{q} \in \mathbb{R}^3$ ahead of the notch tip at each point $\mathbf{p}_i$, as such

$$\mathbf{q}_i = \mathbf{p}_i - \xi \mathbf{n}_e^{(i)}, \tag{4.4.3}$$

where $\xi > 0$ is the distance or length parameter for non-local criteria evaluation. Any point on a circular arc at a radial distance ahead of the notch tip can be found by rotating the reference point, that is

$$\mathbf{R}\mathbf{q}_i. \tag{4.4.4}$$

Here, $\mathbf{R} \in \mathbb{R}^{3\times3}$ is obtained from the Rodrigues formula:

$$\mathbf{R} = \mathbf{I} + [\tilde{\mathbf{e}}_1]_\times \sin\theta + [\tilde{\mathbf{e}}_1]_\times^2 (1 - \cos\theta), \tag{4.4.5}$$

in which $\tilde{\mathbf{e}}_1$ represents the axis of rotation and $\theta \in \mathbb{R}$ the rotation angle (counter-clockwise is positive).

v. (Modified McClinotck) Find the maximum circumferential stress at the critical radial distance ahead of each point $\mathbf{p}_i$.

(Novozhilov-Seweryn) Find the maximum averaged stress over the critical length ahead of each point $\mathbf{p}_i$.

(Minimum SED) Find the minimum strain energy density at the critical radial distance ahead of each point $\mathbf{p}_i$.

(Proposed) Find the minimum total potential energy change associated with domain perturbation at a critical radial distance ahead of each point $\mathbf{p}_i$.

Generally, the stress computed by the finite element analysis in the global frame must be transformed before criteria evaluation:

(a) That is, with respect to the local orthonormal basis $\tilde{\mathbf{e}}_1, \tilde{\mathbf{e}}_2^{(i)}, \tilde{\mathbf{e}}_3^{(i)}$ at each point $\mathbf{p}_i$. This can be achieved through a change of basis. The transformed Cauchy stress tensor is given by

$$\mathbf{Q}\boldsymbol{\sigma}\mathbf{Q}^{\mathrm{T}}, \tag{4.4.6}$$

where $\mathbf{Q} \in \mathbb{R}^{3\times3}$ is the proper orthogonal tensor (refer to Appendix A.2).

(b) Then, if need be, the local stress must be converted into a local cylindrical form (r, θ, z).

The discovery of an optimum is a single-variable optimisation problem with respect to the angle of rotation. Once the optimum is bracketed, Brent's method [60] can be applied for a derivative-free search or a variant of Brent's method informed by a first derivative can be employed to supply the optimum value. The derivative-free search involves a choice between a bisection or an inverse quadratic interpolation at each step until convergence is determined. At best, Brent's method exhibits superlinear convergence, at worst, linear convergence [60]. Routines for these operations can be found in [61] and as such are not reproduced here.

vi. Ahead of each point $\mathbf{p}_i$, check if the quantity of interest exceeds the critical value. If so, crack propagation is said to occur. The fracture load and the propagation direction can be reported. If not, increment the loading in step i and repeat all the steps.

4.4.2 Implementation in FreeFem++

FreeFem++ is fully capable of processing three-dimensional problems. Following the refinement operation in TetGen, the facets incident on the boundary are once again identified. A label is associated with each facet according to the nature of the boundary condition. A file with the extension .mesh [62] containing the mesh and label information is then written. This forms one of the two inputs required by FreeFem++, the second being the variational formulation of the boundary value problem of linear elastostatics. An example of the script for this is not provided here; the reader is referred to the FreeFem++ manual [62]. While continuous piecewise cubic and quartic finite elements can be employed in a two-dimensional analysis, the range of finite elements is not as extensive for the three-dimensional case. Only piecewise constant, continuous piecewise linear and quadratic finite elements are on offer. Continuous piecewise quadratic finite elements are, therefore, employed in this work to compute both the displacement and stress.

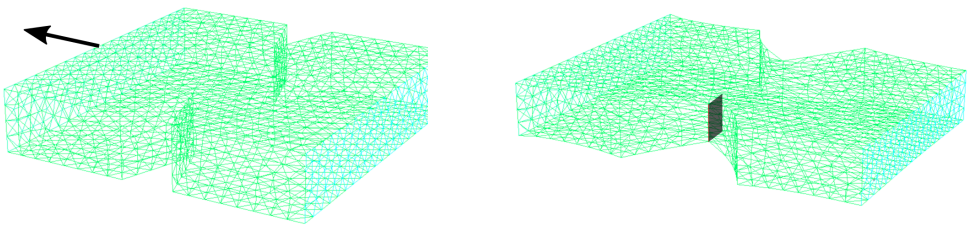
The FreeFem++ interface is a high level language that appears to take inspiration from C++. It is not truly designed to handle lengthy or complex source code. For example, there is

no class data structure or an equivalent. However, FreeFem++ does provide a comprehensive catalogue of in-built functions ranging from visualisation to optimisation routines among many others. In lieu of this, the Nelder-Mead algorithm [63] (a direct search method that maintains a non-degenerate simplex in each iteration) available can comfortably tackle the aforementioned optimisation problem in place of Brent's method.

Three numerical examples are now provided to demonstrate the capability of the developed crack onset prediction procedure. Only the crack propagation direction as determined by the modified McClintock criterion (selected so as to showcase the full procedure) is reported. This parameter is unaffected by the amplification of the loading boundary condition.

4.4.3 Double edge V-notched specimen with $2\beta = 40°$

One end (blue edges on surface in Figure 4.9a) was fully constrained in all directions so as to prevent any possible displacement. The opposite end was subject to a displacement of 0.2 mm in the outward normal direction to the surface. Material properties are those of the polymethyl metacrylate in [40] and are as follows: Young's modulus is 3.3 GPa; Poisson's ratio is 0.35; the tensile strength is 102.8 MPa; and the fracture toughness is 1.202 MPa$\sqrt{\text{m}}$. Results of the deformation along with a representation of the crack propagation direction are pictorially depicted in Figure 4.9 for a coarse mesh. The analysis was, in fact, carried out on the fine mesh version on a high performance computing platforms. As expected, crack propagation was predicted to occur in the direction that is orthogonal to the displacement.

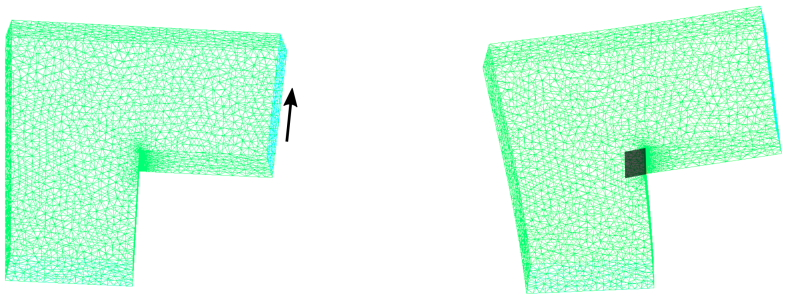


(a) (b)

Figure 4.9. (a) Reference mesh and (b) deformed mesh (exaggerated) of the double edge V-notched specimen with $2\beta = 40°$ and crack path representation.

4.4.4 L-shaped specimen

In the first scenario, the base of the vertical leg was fully constrained and a shear displacement of 0.2 mm was applied on the right-most surface as shown in Figure 4.10a. The boundary conditions come close to replicating the experiment shown in Figure 4.2. Material properties are those of the plain concrete in [58] and are as follows: Young's modulus is 25.85 GPa; Poisson's ratio is 0.18; the tensile strength is 2.7 MPa; and the fracture toughness is 1.318 MPa$\sqrt{\text{m}}$. The resulting deformation along with the crack propagation direction is presented in Figure 4.10b. Crack propagation was predicted to occur approximately 1.12° in the upward direction from the inner corner edge and is in-line with experiment [58].



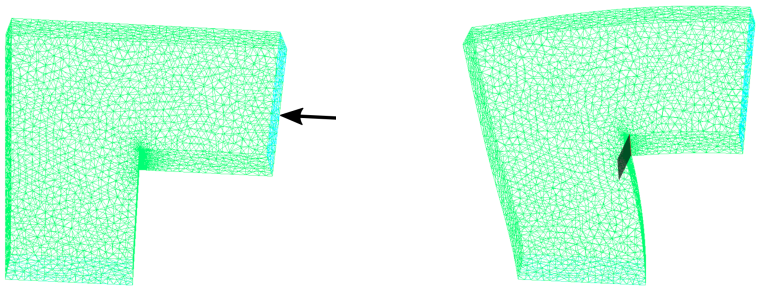
(a) (b)

Figure 4.10. (a) Reference mesh and (b) deformed mesh (exaggerated) of the L-shaped specimen subject to a shear displacement and crack path representation.

In the second scenario, the base of the vertical leg was fully constrained and a displacement of 0.2 mm was applied in the direction opposite to the outward normal of the right-most surface as shown in Figure 4.11a. Material properties are the same as before. Figure 4.11b captures the deformation. This time, crack propagation was predicted to occur approximately 52.7° in the downward direction from the inner corner edge.

4.5 Conclusions

This chapter sought to first introduce the basic notions of discrete geometry in order to provide the correct terminology and definitions relating to a finite element mesh. All too often, this is lacking in engineering literature. Yet, this knowledge is absolutely fundamental in the design of effective and efficient algorithms. A number of index-based algorithms were developed



(a) (b)

Figure 4.11. (a) Reference mesh and (b) deformed mesh (exaggerated) of the L-shaped specimen subject to a normal displacement and crack path representation.

to detect sharp notch tips in a quasi-uniform mesh. Unordered associative containers in the C++ standard library were an essential ingredient. The first two algorithms presented extract facets and edges incident on the boundary of a mesh. This information is required for the identification of edges that are a part or the entirety of a sharp notch tip. An algorithm to establish the connectivity of the edges forming a notch tip was also disclosed. An important reason for this operation is to obtain the shape profile of the notch tip, which permits the evaluation of non-local criteria ahead of the tip in a mesh-independent manner. Finally, an algorithm to re-generate the original mesh in TetGen with local refinement was provided. This mesh design strategy improves the resolution of the solution in the local vicinity of a sharp notch tip in a finite element analysis. Illustrative examples demonstrating the capability of the algorithms on a number of geometries – double edge V-notched specimens, an L-shaped specimen and a graphite fuel brick – with straight notch tips acted as validation. While these algorithms were developed with straight notch tips in mind, they are equally valid for curved sharp notch tips with corner points. However, no examples containing curved notch tips were considered. This should be addressed in future work, along with curved notch tips with no corner points.

In the second part of this chapter, a general procedure for crack onset prediction at a sharp notch tip was developed. This covered popular non-local criteria, namely the modified McClintock, Novozhilov-Seweryn and minimum SED, as well as the criterion proposed in the previous chapter. Implementation details in FreeFem++ were discussed. Three numerical examples where the crack propagation direction in pure and mixed-mode fracture scenarios was determined by the modified McClintock criterion were given. FreeFem++ was selected for the capability to interrogate the solution at any given point in the studied mesh. However, FreeFem++ does not support local p-refinement at a geometric feature. This is somewhat

unfortunate as the h-p version of the FEM is known to provide the most attractive convergence rate [64]. It makes sense that any future implementation should accommodate an increase in the polynomial degree of finite elements local to the sharp notch tip as well as mesh refinement.

Chapter 5

Summary and Future Work

In this project, methodology for coupling a dynamic multi-body finite element contact code with a crack propagation capable code, namely SOLFEC and MoFEM, has been developed. Firstly, data output by SOLFEC at a user-instigated time-step requires conditioning. This involves the mitigation of rigid body motion as it is the surface displacement that drives crack propagation in MoFEM. A cost function based on the skew-symmetric part of the displacement gradient was derived for minimisation purposes. Newton's method on the Lie group $SO(3)$ proved to be a particularly efficient procedure for the rotation component of the rigid body displacement, as the geometric structure of the underlying parameter space is exploited. There are no issues with singularities or additional computations for scaling the solution in each iteration onto the manifold to worry about.

Secondly, location of crack onset and automatic application of crack initiation criteria were tackled. It was shown that unordered associative containers in the C++ standard library and Cantor encoding are all that is needed in identifying edges in a quasi-uniform mesh representing a sharp notch tip. Algorithms related to the extraction of facets and edges incident on the surface as well as edge connectivity on a notch tip were disclosed. Furthermore, it was argued that local mesh refinement and polynomial degree increase of the finite elements near a notch tip are best suited to resolve the steep gradient resulting from the solution of the boundary value problem of linear elastostatics. Therefore, a routine for local mesh refinement in TetGen was also presented.

A non-local energy consistent fracture condition that is equally applicable to sharp and blunt notches was also proposed in this work to determine crack onset in a brittle and isotropic linear elastic body. The finite difference form of the energy release rate introduces a char-

acteristic length parameter. An energy equivalent to the Novozhilov-Seweryn criterion was derived to obtain the value. The change of total potential energy was approximated through an asymptotic argument involving the topological derivative. Experimental validation involved the successful prediction of crack onset in V-notched, circular notched and U-notched specimens made of PMMA subject to pure and mixed-mode conditions. It is noteworthy that only the fracture toughness and tensile strength are required for application of the proposed criterion. In order to evaluate the criterion, the stress solution at a critical distance from the boundary must be interrogated. A general post-processing procedure for evaluation of non-local criteria was devised and an implementation made in FreeFem++ as proof of concept.

For there to be a true coupling of SOLFEC and MoFEM, the return of fractured graphite bricks into a SOLFEC analysis must be addressed. SOLFEC cannot accommodate the modification of geometry or insertion of bodies during simulation run-times. The re-design and implementation of the relevant data structures is a serious undertaking. It is worth investigating whether an analysis can be restarted with the fractured graphite bricks following each time-step when crack onset is detected. This would require taking a snapshot of the state of the bricks, which would then be accounted for in the definition of the bodies and boundary conditions. Alternatively, if this is not possible, an analysis of the excited AGR core could be run to completion and the fractured graphite bricks could be included from the onset of a repeat analysis with the same boundary conditions. However, an assumption that fracture of one graphite brick has no bearing on others would have to be made! This leaves the question of mesh density. If coarsening is required, the surface of the mesh needs to be extracted and subjected to a mesh simplification operation. Simplification can be easily accomplished in the CGAL library [65] without the need for exact predicates or constructions. In the simplification process, selected edges are collapsed and replaced by vertices based on a user-supplied cost and placement function. This takes place without distorting the topology, that is as much as possible depending on the termination criterion. Once simplification of the surface is completed, the volume mesh can be re-generated in TetGen.

In truth, the methodology for coupling was somewhat dictated by the capability of SOLFEC. It is the inability to resolve the displacement solution with a high level of accuracy in a graphite brick that was the overarching factor. As it stands, user intervention is required to identify the onset of fracture. Unfortunately, this is an impediment to the complete automation of the coupling between SOLFEC and MoFEM. An upgrade to a parallel solver for the FEM part of SOLFEC is, therefore, in order. A scalable finite element library, such as MFEM [66], represents a potential way forward. Computationally, there is no obstacle

to the integration of functions from a library like MFEM in the source code of SOLFEC. This would, in fact, be little effort for great reward. For example, MFEM would add support for higher-order finite element spaces, local mesh refinement, mixed finite elements and discontinuous Galerkin schemes to name but a few features. This would enable the evaluation of fracture criteria within the framework of SOLFEC. Detection of crack onset could then be automated. Moreover, a fracture condition that takes into account dynamic effects would be feasible. While dynamic effects are thought to play only a minor role in the fracture of graphite bricks in an AGR core, a criterion informed by the history is more realistic. To that end, the Structural-Temporal criterion [67], where a time dimension is added to the Novozhilov-Seweryn criterion, could be the first port of call. The criterion assesses the integral of the force in the Novozhilov-Seweryn criterion with respect to time. This means that there is an additional characteristic time parameter to work with. It is thought that this parameter must be some kind of incubation period for macro-fracture development [67].

The achievements in this project can be seen as a first step towards a computational framework for the study of fracture in graphite bricks through the coupling of SOLFEC and MoFEM. It is abundantly clear that an update to the FEM solver in SOLFEC is necessary. This would provide the backdrop for complete automation of the coupling and support implementation of a dynamic fracture criterion. Also, the ability to study deformation in an AGR core analysis with fractured graphite bricks is very important.

Appendix A

Vectors and Tensors

A.1 Vector and tensor algebra

The Cartesian components of vector **a** are

$$a_i = \mathbf{a} \cdot \mathbf{e}_i. \tag{A.1.1}$$

It follows that

$$\mathbf{a} = a_i \mathbf{e}_i. \tag{A.1.2}$$

The inner product of vectors **a** and **b** can be expressed in component form as

$$\mathbf{a} \cdot \mathbf{b} = a_i b_i, \tag{A.1.3}$$

where the Einstein summation (over the repeated index) is observed. A (linear) mapping of a vector **a** into another vector can be accomplished through a second-order tensor, as such

$$\mathbf{b} = \mathbf{S}\mathbf{a}. \tag{A.1.4}$$

A second-order tensor of note is the identity tensor, defined thus:

$$\mathbf{I}\mathbf{a} = \mathbf{a}. \tag{A.1.5}$$

Now, the second-order tensor **S** has the components

$$S_{ij} = \mathbf{e}_i \cdot \mathbf{S}\mathbf{e}_j. \tag{A.1.6}$$

Then,

$$(\mathbf{Sa})_i = S_{ij}a_j. \tag{A.1.7}$$

Note that $(\mathbf{I})_{ij} = \delta_{ij} = \mathbf{e}_i \cdot \mathbf{e}_j$. A contraction of the tensor $\mathbf{S}$ is defined as S_{ii}, which returns a scalar. The scalar is known as the trace of $\mathbf{S}$, and is typically written $\operatorname{tr}\mathbf{S}$. Additionally, the tensor product of two vectors gives rise to a second-order tensor,

$$(\mathbf{a} \otimes \mathbf{b})_{ij} = a_i b_j, \tag{A.1.8}$$

in which the symbol $\otimes$ denotes said operation.

The product of two second-order tensors $\mathbf{S}$ and $\mathbf{T}$ in component form is

$$(\mathbf{ST})_{ij} = S_{ik}T_{kj}. \tag{A.1.9}$$

Their inner product is given by a double contraction, that is

$$\mathbf{S} : \mathbf{T} = S_{ij}T_{ij}. \tag{A.1.10}$$

Matrix representation of a tensor, say $\mathbf{S}$, is written as $[\mathbf{S}]$. However, this is not always necessary, especially when matrix operations are obvious. Transpose of tensor $\mathbf{S}$ is simply $(\mathbf{S}^{\mathrm{T}})_{ij} = (\mathbf{S})_{ji}$, where the superscript T indicates the operation. A useful relation employing the transpose of a second-order tensor is

$$\mathbf{Sa} \cdot \mathbf{b} = \mathbf{a} \cdot \mathbf{S}^{\mathrm{T}}\mathbf{b}. \tag{A.1.11}$$

Also, note that

$$(\mathbf{ST})^{\mathrm{T}} = \mathbf{T}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}} \text{ and } (\mathbf{S} + \mathbf{T})^{\mathrm{T}} = \mathbf{S}^{\mathrm{T}} + \mathbf{T}^{\mathrm{T}}. \tag{A.1.12}$$

Transpose of the matrix $[\mathbf{S}]$ is just $[\mathbf{S}]^{\mathrm{T}} = [\mathbf{S}^{\mathrm{T}}]$. A vector can be thought of as a column matrix, so that

$$\mathbf{a} \cdot \mathbf{b} = [\mathbf{a}]^{\mathrm{T}}[\mathbf{b}]. \tag{A.1.13}$$

A second-order tensor $\mathbf{S}$ is said to be symmetric if

$$\mathbf{S} = \mathbf{S}^{\mathrm{T}} \tag{A.1.14}$$

and skew-symmetric if

$$\mathbf{S} = -\mathbf{S}^{\mathrm{T}}. \tag{A.1.15}$$

It is possible to decompose any second-order tensor into symmetric and skew-symmetric parts, e.g.

$$\mathbf{S} = \mathbf{V} + \mathbf{W}, \tag{A.1.16}$$

where the symmetric tensor $\mathbf{V}$ is given by

$$\mathbf{V} = \frac{1}{2}(\mathbf{S} + \mathbf{S}^{\mathrm{T}}) \tag{A.1.17}$$

and the skew-symmetric tensor $\mathbf{W}$ by

$$\mathbf{W} = \frac{1}{2}(\mathbf{S} - \mathbf{S}^{\mathrm{T}}). \tag{A.1.18}$$

If second-order tensor $\mathbf{S}$ is invertible, then there exists a tensor $\mathbf{S}^{-1}$ with the property

$$\mathbf{S}^{-1}\mathbf{S} = \mathbf{S}\mathbf{S}^{-1} = \mathbf{I}. \tag{A.1.19}$$

Furthermore, $\mathbf{S}$ is positive-definite if

$$\mathbf{v} \cdot \mathbf{S}\mathbf{v} > 0, \tag{A.1.20}$$

for any vector $\mathbf{v} \neq \mathbf{0}$.

A fourth-order tensor, say $\boldsymbol{C}$, maps a second-order tensor $\mathbf{S}$ into a second-order tensor, as such

$$\mathbf{T} = \boldsymbol{C}\mathbf{S}. \tag{A.1.21}$$

The fourth-order equivalent to the second-order identity tensor is defined as

$$(\mathbf{II})_{ijkl} = \frac{1}{2}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \tag{A.1.22}$$

which serves the purpose of mapping a symmetric second-order tensor onto itself.

A.2 Change of basis

Consider a rotation of the basis in a three-dimensional right-handed Cartesian coordinate system with the origin fixed. Let the transformed basis be identified by an overhead hat, that is $\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3$. The vector $\mathbf{a}$ in this coordinate system can be expressed as

$$\mathbf{a} = \hat{a}_i \hat{\mathbf{e}}_i. \tag{A.2.1}$$

Recall that $\mathbf{a} = a_i \mathbf{e}_i$ in the original coordinate system. The components of $\mathbf{a}$ in the transformed coordinate system can be written in terms of those in the original, as such

$$\hat{a}_i = \hat{\mathbf{e}}_i \cdot \mathbf{a} = \hat{\mathbf{e}}_i \cdot (a_j \mathbf{e}_j) = (\hat{\mathbf{e}}_i \cdot \mathbf{e}_j) a_j. \tag{A.2.2}$$

Accordingly, a vector subject to a change of basis transforms according to

$$\hat{a}_i = Q_{ij} a_j, \tag{A.2.3}$$

where $\mathbf{Q} = \hat{\mathbf{e}}_i \cdot \mathbf{e}_j$ represents the cosine of the angle between $\hat{\mathbf{e}}_i$ and $\mathbf{e}_j$. It is possible to express the components of $\mathbf{a}$ in the original coordinate system in terms of the transformed ones through

$$a_i = \mathbf{e}_i \cdot \mathbf{a} = \mathbf{e}_i \cdot (\hat{a}_j \hat{\mathbf{e}}_j) = Q_{ji} \hat{a}_j. \tag{A.2.4}$$

It can be seen that $\mathbf{Q}^{-1} = \mathbf{Q}^{\mathrm{T}}$. This condition is the very definition of an orthogonal tensor. Let the determinant of a second-order tensor be that of the matrix representation, i.e. $\det \mathbf{Q} = \det [\mathbf{Q}]$. Since $\det (\mathbf{Q}^{\mathrm{T}} \mathbf{Q}) = (\det \mathbf{Q}^{\mathrm{T}})(\det \mathbf{Q})$ and $\det \mathbf{Q}^{\mathrm{T}} = \det \mathbf{Q}$, then

$$1 = \det \mathbf{I} = \det (\mathbf{Q}^{\mathrm{T}} \mathbf{Q}) = (\det \mathbf{Q}^{\mathrm{T}})(\det \mathbf{Q}) = (\det \mathbf{Q})^2 \tag{A.2.5}$$

and

$$\det \mathbf{Q} = \pm 1. \tag{A.2.6}$$

Therefore, the determinant of an orthogonal tensor is either $+1$ or -1. A rotation is an orthogonal tensor with $\det \mathbf{Q} = +1$, which is said to be a proper orthogonal transformation.

Finally, a second-order tensor transforms according to the rule

$$\hat{S}_{ij} = Q_{ip} Q_{jq} S_{pq} \tag{A.2.7}$$

under a change of basis.

A.3 Pseudovectors

This time, consider a transformation of the three-dimensional right-handed Cartesian coordinate system:

$$(Q_{1l} \mathbf{e}_l \times Q_{2m} \mathbf{e}_m) \cdot Q_{3n} \mathbf{e}_n = Q_{1l} Q_{2m} Q_{3n} (\mathbf{e}_l \times \mathbf{e}_m) \cdot \mathbf{e}_n. \tag{A.3.1}$$

It follows that

$$Q_{1l}Q_{2m}Q_{3n}(\mathbf{e}_l \times \mathbf{e}_m) \cdot \mathbf{e}_n = \det \mathbf{Q}\,(\mathbf{e}_1 \times \mathbf{e}_2) \cdot \mathbf{e}_3. \tag{A.3.2}$$

If the transformation is proper, $\det \mathbf{Q} = +1$, the handedness of the coordinate system is maintained. However, this is not true if $\det \mathbf{Q} = -1$, which is labelled as an improper transformation. Introducing the Levi-Civita symbol $\epsilon_{ijk} = (\mathbf{e}_i \times \mathbf{e}_j) \cdot \mathbf{e}_k$, then

$$\epsilon_{ijk} = \begin{cases} 0 & \text{if } i = j \text{ or } i = k \text{ or } j = k, \\ +1 & \text{if } i, j, k \text{ is an even (cyclic) permutation of } 1, 2, 3, \\ -1 & \text{if } i, j, k \text{ is an odd (cyclic) permutation of } 1, 2, 3, \end{cases} \tag{A.3.3}$$

and the determinant of a second-order tensor $\mathbf{S}$ can be expressed as

$$\det \mathbf{S} = \epsilon_{ijk} S_{1i} S_{2j} S_{3k} = \epsilon_{ijk} S_{i1} S_{j2} S_{k3}. \tag{A.3.4}$$

More generally,

$$\epsilon_{ijk} S_{li} S_{mj} S_{nk} = \epsilon_{lmn} \det \mathbf{S}. \tag{A.3.5}$$

A useful identity relating the product of two Levi-Civita symbols to a series of Kronecker deltas is

$$\epsilon_{ijk}\epsilon_{ilm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl}. \tag{A.3.6}$$

A pseudovector adheres to the transformation law

$$\hat{a}_i = \det(\mathbf{Q})\, Q_{ij} a_j. \tag{A.3.7}$$

Thus, a pseudovector behaves like a standard vector under a proper transformation but changes sign when subject to an improper transformation. An important example is the cross product of two vectors, $\mathbf{c} = \mathbf{a} \times \mathbf{b}$, which in component form is written as

$$c_i = \epsilon_{ijk} a_j b_k. \tag{A.3.8}$$

Another is associated with a second-order skew-symmetric tensor $\mathbf{W}$ through

$$w_i = -\frac{1}{2}\epsilon_{ijk} W_{jk}. \tag{A.3.9}$$

Note that a vector $\mathbf{a}$ can be cast into a second-order skew-symmetric tensor, as such

$$W_{ij} = \epsilon_{ikj} a_k. \tag{A.3.10}$$

Let $[\mathbf{a}]_\times = \epsilon_{ikj} a_k$, then the cross product can also be expressed as

$$\mathbf{a} \times \mathbf{b} = [\mathbf{a}]_\times \mathbf{b}. \tag{A.3.11}$$

A.4 Gradient and divergence

In continuum mechanics, the concept of a field is frequently encountered. A function that assigns a quantity to a point is said to be a field. Thus, a field can be a scalar, vector or even a second-order tensor depending on the nature of the quantity. Given a smooth scalar field $\phi(\mathbf{x})$, the gradient at point $\mathbf{x}$ is written $\boldsymbol{\nabla}\phi(\mathbf{x})$, where

$$\boldsymbol{\nabla} = \mathbf{e}_i \frac{\partial}{\partial x_i} \tag{A.4.1}$$

is the gradient operator. Observe that a vector results. Indeed, the gradient of a smooth vector field $\mathbf{a}$ at $\mathbf{x}$, written $\boldsymbol{\nabla}\mathbf{a}(\mathbf{x})$, returns a second-order tensor. In component form, the gradient of a scalar and vector can be expressed as

$$(\boldsymbol{\nabla}\phi)_i = \frac{\partial \phi}{\partial x_i} = \phi_{,i} \tag{A.4.2}$$

and

$$(\boldsymbol{\nabla}\mathbf{a})_{ij} = \frac{\partial a_i}{\partial x_j} = a_{i,j}, \tag{A.4.3}$$

respectively.

The divergence of vector field $\mathbf{a}$ is $\operatorname{div}\mathbf{a} = \boldsymbol{\nabla}\cdot\mathbf{a}$, which produces a scalar:

$$\operatorname{div}\mathbf{a} = \frac{\partial a_i}{\partial x_i} = a_{i,i}. \tag{A.4.4}$$

Similarly, a vector results from the divergence of a smooth second-order tensor field $\mathbf{S}$:

$$(\operatorname{div}\mathbf{S})_i = \frac{\partial S_{ij}}{\partial x_j} = S_{ij,j}. \tag{A.4.5}$$

Finally, the curl of vector field $\mathbf{a}$, denoted $\operatorname{curl}\mathbf{a}$, is given by

$$\boldsymbol{\nabla}\times\mathbf{a} = \epsilon_{ijk} a_{k,j}. \tag{A.4.6}$$

A.5 Divergence theorem of Gauss

An integral theorem that plays an important role in this thesis belongs to Gauss. The divergence theorem provides a relationship between the surface integral of a scalar, vector or second-order tensor field and the volume integral of the divergence of the quantity in question.

Let $\mathcal{B}$ be a bounded and connected open subset of $\mathbb{R}^d$ with a piecewise smooth boundary $\partial\mathcal{B}$. Mathematically, the divergence theorem of Gauss can be written in component form as

$$\int_{\partial\mathcal{B}} \phi n_i \, ds = \int_{\mathcal{B}} \phi_{,i} \, dy, \tag{A.5.1a}$$

$$\int_{\partial\mathcal{B}} a_i n_i \, ds = \int_{\mathcal{B}} a_{i,i} \, dy, \tag{A.5.1b}$$

$$\int_{\partial\mathcal{B}} S_{ij} n_j \, ds = \int_{\mathcal{B}} S_{ij,j} \, dy, \tag{A.5.1c}$$

where $\mathbf{n}$ is the outward unit normal vector to $\partial\mathcal{B}$.

Appendix B

Proof of $\mathbf{R}(\mathbf{a} \times \mathbf{b}) = \mathbf{Ra} \times \mathbf{Rb}$

Proof. Employing the Levi-Civita symbol on the right-hand side of the equation:

$$\begin{aligned}(\mathbf{Ra} \times \mathbf{Rb})_k &= \epsilon_{ijk} R_{im} R_{js} a_m b_s \\ &= \epsilon_{ijt} \delta_{kt} R_{im} R_{js} a_m b_s \\ &= \epsilon_{ijt} R_{kr} R_{tr} R_{im} R_{js} a_m b_s,\end{aligned} \tag{B.0.1}$$

as $R_{kr} R_{tr} = \delta_{kt}$. Recall that $\det \mathbf{R} = 1$, then $\epsilon_{ijt} R_{im} R_{js} R_{tr} = \epsilon_{msr} \det \mathbf{R} = \epsilon_{msr}$. Thus,

$$\begin{aligned}(\mathbf{Ra} \times \mathbf{Rb})_k &= R_{kr} \epsilon_{msr} a_m b_s \\ &= R_{kr} (\mathbf{a} \times \mathbf{b})_r \\ &= \left(\mathbf{R}(\mathbf{a} \times \mathbf{b})\right)_k.\end{aligned} \tag{B.0.2}$$

□

Appendix C

V-notch Stress Distribution Angular Functions

The components of the angular functions $m_{ij}^{I}(\theta)$ and $m_{ij}^{II}(\theta)$ are taken from [68]. For Mode-I:

$$m_{rr}^{I}(\theta) = \frac{1}{\sqrt{2\pi}J_I}\left[H_I\cos(\eta_I+1)\theta - (\eta_I-3)\cos(\eta_I-1)\theta\right],$$

$$m_{\theta\theta}^{I}(\theta) = \frac{1}{\sqrt{2\pi}J_I}\left[-H_I\cos(\eta_I+1)\theta + (\eta_I+1)\cos(\eta_I-1)\theta\right],$$

$$m_{r\theta}^{I}(\theta) = \frac{1}{\sqrt{2\pi}J_I}\left[-H_I\sin(\eta_I+1)\theta + (\eta_I-1)\sin(\eta_I-1)\theta\right], \tag{C.0.1}$$

where

$$H_I = \eta_I\cos 2\alpha + \cos(2\eta_I\alpha) \tag{C.0.2}$$

and

$$J_I = \eta_I + 1 - H_I. \tag{C.0.3}$$

For Mode-II:

$$m_{rr}^{II}(\theta) = \frac{1}{\sqrt{2\pi}J_{II}}\left[-H_{II}\sin(\eta_{II}+1)\theta + (\eta_{II}-3)\sin(\eta_{II}-1)\theta\right],$$

$$m_{\theta\theta}^{II}(\theta) = \frac{1}{\sqrt{2\pi}J_{II}}\left[H_{II}\sin(\eta_{II}+1)\theta - (\eta_{II}+1)\sin(\eta_{II}-1)\theta\right],$$

$$m_{r\theta}^{II}(\theta) = \frac{1}{\sqrt{2\pi}J_{II}}\left[-H_{II}\cos(\eta_{II}+1)\theta + (\eta_{II}-1)\cos(\eta_{II}-1)\theta\right], \tag{C.0.4}$$

where

$$H_{II} = \eta_{II}\cos 2\alpha - \cos(2\eta_{II}\alpha) \tag{C.0.5}$$

and

$$J_{II} = \eta_{II} - 1 - H_{II}. \tag{C.0.6}$$

Appendix D

Topological Derivative Proofs

D.1 Hole extension

Defining the topological derivative for an extended hole as

$$D_T^* \coloneqq \lim_{\substack{\xi\to 0\\ \delta\xi\to 0}} \frac{\chi(\Omega_{\xi+\delta\xi}) - \chi(\Omega_\xi)}{f(\xi+\delta\xi) - f(\xi)}. \tag{D.1.1}$$

The following provides a proof for $D_T(\mathbf{x}_0) = D_T^*(\mathbf{x}_0)$, which is reproduced from [69].

Proof. For a shape functional with an extended hole, the first-order asymptotic expansion has the form

$$\chi(\Omega_{\xi+\delta\xi}) = \chi(\Omega) + f(\xi+\delta\xi)D_T(\mathbf{x}_0) + \mathscr{R}\big(f(\xi+\delta\xi)\big). \tag{D.1.2}$$

Subtracting the (first-order) asymptotic expansion of $\chi(\Omega_\xi)$ given in (3.3.1), we have

$$\chi(\Omega_{\xi+\delta\xi}) - \chi(\Omega_\xi) = \big(f(\xi+\delta\xi) - f(\xi)\big)D_T(\mathbf{x}_0) + \mathscr{R}\big(f(\xi+\delta\xi)\big) - \mathscr{R}\big(f(\xi)\big). \tag{D.1.3}$$

Dividing throughout by $f(\xi+\delta\xi) - f(\xi)$ yields

$$\frac{\chi(\Omega_{\xi+\delta\xi}) - \chi(\Omega_\xi)}{f(\xi+\delta\xi) - f(\xi)} = D_T(\mathbf{x}_0) + \frac{\mathscr{R}\big(f(\xi+\delta\xi)\big) - \mathscr{R}\big(f(\xi)\big)}{f(\xi+\delta\xi) - f(\xi)}. \tag{D.1.4}$$

Taking the limits $\delta\xi \to 0$ and $\xi \to 0$ in the following manner

$$D_T^*(\mathbf{x}_0) = \lim_{\substack{\xi\to 0\\ \delta\xi\to 0}} \frac{\chi(\Omega_{\xi+\delta\xi}) - \chi(\Omega_\xi)}{f(\xi+\delta\xi) - f(\xi)} = D_T(\mathbf{x}_0) + \lim_{\substack{\xi\to 0\\ \delta\xi\to 0}} \frac{\mathscr{R}\big(f(\xi+\delta\xi)\big) - \mathscr{R}\big(f(\xi)\big)}{f(\xi+\delta\xi) - f(\xi)}, \tag{D.1.5}$$

and considering (3.3.2) to complete the proof as

$$\lim_{\substack{\xi\to 0\\ \delta\xi\to 0}} \frac{\mathscr{R}(f(\xi+\delta\xi))-\mathscr{R}(f(\xi))}{f(\xi+\delta\xi)-f(\xi)} = 0. \quad \square$$

$\square$

D.2 Shape sensitivity

Defining the topological derivative as

$$D_T(\mathbf{x}_0) = \lim_{\xi\to 0} \frac{1}{f'(\xi)} \frac{d}{d\tau}\chi(\Omega_\tau)\Big|_{\tau=0},$$

where $f(\xi)$ is a function selected so that $0 < |D_T(\mathbf{x}_0)| < \infty$. The proof is based on that in [69].

Proof. The sensitivity of a shape functional to hole extension is given by

$$\frac{d}{d\xi}\chi(\Omega_\xi) = \lim_{\delta\xi\to 0} \frac{\chi(\Omega_{\xi+\delta\xi})-\chi(\Omega_\xi)}{\delta\xi}. \tag{D.2.1}$$

Characterising the hole extension by the mapping function $\boldsymbol{\varphi}(\mathbf{x},\tau)$, and for sufficiently small $\tau\in\mathbb{R}^+$ the limit can be re-written as follows:

$$\lim_{\delta\xi\to 0} \frac{\chi(\Omega_{\xi+\delta\xi})-\chi(\Omega_\xi)}{\delta\xi} = \lim_{\tau\to 0} \frac{\chi(\Omega_\tau)-\chi(\Omega_\xi)}{\tau}, \tag{D.2.2}$$

where $\delta\xi = \tau V$ (recall that V is equal to unity). The limit can then be expressed as a derivative with respect to τ, as such

$$\lim_{\tau\to 0} \frac{\chi(\Omega_\tau)-\chi(\Omega_\xi)}{\tau} = \frac{d}{d\tau}\chi(\Omega_\tau)\Big|_{\tau=0}. \tag{D.2.3}$$

Taking the derivative of the (first-order) asymptotic expansion in (3.3.1) with respect to ξ yields

$$\frac{d}{d\xi}\chi(\Omega_\xi) = f'(\xi)D_T(\mathbf{x}_0) + \mathscr{R}'(f(\xi))f'(\xi). \tag{D.2.4}$$

Substituting the result in (D.2.3) into (D.2.4) and dividing throughout by $f'(\xi)$ returns

$$D_T(\mathbf{x}_0) + \mathscr{R}'(f(\xi)) = \frac{1}{f'(\xi)} \frac{d}{d\tau}\chi(\Omega_\tau)\Big|_{\tau=0}. \tag{D.2.5}$$

Applying the limit $\xi\to 0$ and considering (3.3.2) to complete the proof as

$$\lim_{\xi\to 0} \mathscr{R}'(f(\xi)) = 0. \quad \square$$

$\square$

Appendix E

Cantor Pairing

Encoding, in set theory, involves a one-to-one mapping from a set to another. Of interest is integer encoding, which is an encoding from a set into $\mathbb{N}$. Informally, this can be thought of as the association of a single unique natural number with several pieces of information. An integer pairing function dealing with non-negative integers is an encoding from $\mathbb{N}\times\mathbb{N}$ into $\mathbb{N}$. Cantor discovered such a function, $\langle\cdot,\cdot\rangle : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$, that is a second-degree polynomial:

$$\langle x_1, x_2\rangle = \frac{1}{2}(x_1 + x_2)(x_1 + x_2 + 1) + x_1, \qquad \forall\,(x_1, x_2) \in \mathbb{N}^2. \tag{E.0.1}$$

Note that the function is not only injective, but also surjective. Therefore, the Cantor pairing function can be readily inverted. Let $m, r, z \in \mathbb{N}$ and introducing the triangular number $T_r = \frac{1}{2}r(r+1) \in \mathbb{N}$, then (**E.0.1**) can be re-written as

$$z = T_m + x_1, \tag{E.0.2}$$

where $m = x_1 + x_2$. The inequality $T_m \leq z < T_{m+1}$ is obvious. This leads to

$$\lambda \leq m < \lambda + 1, \tag{E.0.3}$$

in which $\lambda \in \mathbb{R}$ and $\lambda = \left(-1 + \sqrt{1+8z}\right)/2$. Thus,

$$m = \left\lfloor \frac{-1 + \sqrt{1+8z}}{2} \right\rfloor, \tag{E.0.4}$$

where $\lfloor\,\rfloor$ is the floor function. Accordingly, m can be calculated for any given value of z. Substituting m and z into (**E.0.2**) returns x_1, which subtracted from m yields x_2.

A tripling function, $\langle\cdot,\cdot,\cdot\rangle : \mathbb{N}^3 \to \mathbb{N}$, can be obtained from the composition

$$\langle x_1, x_2, x_3\rangle = \langle x_1, \langle x_2, x_3\rangle\rangle. \tag{E.0.5}$$

More generally, an n-tupling function $\langle \cdot, \ldots, \cdot \rangle : \mathbb{N}^n \to \mathbb{N}$ with $n \in \mathbb{N}$ is defined in [70]:

$$\langle x_1, \ldots, x_n \rangle = \sum_{k=1}^{n} \left\{ \frac{1}{k!} \prod_{j=0}^{k-1} \left[\left(\sum_{i=1}^{k} x_i \right) + j \right] \right\}, \qquad \forall\, (x_1, \ldots, x_n) \in \mathbb{N}^n. \tag{E.0.6}$$

This is an n-degree polynomial and is also a bijection.

Appendix F

Local Mesh Refinement in TetGen

```
// Function to generate a locally refined mesh at sharp notch tips
void GenerateMesh(std::string& file_name, std::vector<double>&
    coordinates, std::vector<i3tuple>& bndFacets, std::vector<  std::
    vector< std::pair<int,int> >  >& notches, double default_size, double
    verts_size, double ends_size) {

  int i;
  std::unordered_set<int> ptsMtrFilled;
  tetgenio in, out;
  tetgenio::facet *f;
  tetgenio::polygon *p;

  // Index starts at 0
  in.firstnumber = 0;

  // Reserve memory for vertex coordinates
  in.numberofpoints = coordinates.size() / 3;
  in.pointlist = new REAL[in.numberofpoints * 3];

  // Reserve memory for sizing data
  in.numberofpointmtrs = 1;
  in.pointmtrlist = new REAL[in.numberofpoints];

  // Set coordinate data
  for (i = 0; i < coordinates.size(); ++i)
    in.pointlist[i] = coordinates[i];

  // Set sizing at each nodal point
```

```
// --> Beginning with ends
for (const auto& x: notches) {
  // One end of notch
  in.pointmtrlist[x.front().first] = ends_size;
  ptsMtrFilled.insert(x.front().first);
  // Opposite end of notch
  in.pointmtrlist[x.back().second] = ends_size;
  ptsMtrFilled.insert(x.back().second);
}

// --> For non-corner on notch
for (const auto& x: notches) {
  for (const auto& y: x) {
      // First point on edge
    if (ptsMtrFilled.count(y.first) == 0) {
      in.pointmtrlist[y.first] = verts_size;
      ptsMtrFilled.insert(y.first);
    }
    // Second point on edge
    if (ptsMtrFilled.count(y.second) == 0) {
      in.pointmtrlist[y.second] = verts_size;
      ptsMtrFilled.insert(y.second);
    }
  }
}

// --> Rest
for (i = 0; i < in.numberofpoints; ++i) {
  if (ptsMtrFilled.count(i) == 0)
    in.pointmtrlist[i] = default_size;
}

// Reserve memory for facets
in.numberoffacets = bndFacets.size();
in.facetlist = new tetgenio::facet[in.numberoffacets];

// Set vertex index list for each facet
for (i = 0; i < bndFacets.size(); ++i) {
  f = &in.facetlist[i];
  f->numberofpolygons = 1;
  f->polygonlist = new tetgenio::polygon[f->numberofpolygons];
  f->numberofholes = 0;
  f->holelist = NULL;
```

```
    p = &f->polygonlist[0];
    p->numberofvertices = 3;
    p->vertexlist = new int[p->numberofvertices];
    p->vertexlist[0] = std::get<0>(bndFacets[i]);
    p->vertexlist[1] = std::get<1>(bndFacets[i]);
    p->vertexlist[2] = std::get<2>(bndFacets[i]);
  }

  // Generate mesh
  tetrahedralize("pqm", &in, &out);

  // Output VTK file
  TetGen2VTK(file_name, out);

}

// Function to output mesh in VTK file format
void TetGen2VTK(std::string& fname, const tetgenio& input) {

  int i;
  std::ofstream fin;
  fname += ".vtk";
  fin.open( fname.c_str() );

    // Print out the header data
    fin << "# vtk DataFile Version 2.0\n";
    fin << "Output Tetrahedral Mesh\n";
    fin << "ASCII\n";

    // Print out the coordinates
    fin << "DATASET UNSTRUCTURED_GRID\n";
    fin << "POINTS " << input.numberofpoints << " float\n";

    for (i=0; i < (3*input.numberofpoints); i+=3)
      fin << input.pointlist[i] << " " << input.pointlist[i+1] << " " <<
   input.pointlist[i+2] << std::endl;

    // Print out vertex index list for each tetrahedron
    fin << std::endl;
    fin << "CELLS " << input.numberoftetrahedra << " " << 5*input.
   numberoftetrahedra << std::endl; // (4+1)*in.number_of_tet
    for (i = 0; i < (4*input.numberoftetrahedra); i+=4)
```

```
        fin << "4 " << input.tetrahedronlist[i] << " " << input.
    tetrahedronlist[i+1] << " " << input.tetrahedronlist[i+2] << " " <<
    input.tetrahedronlist[i+3] << std::endl;

    // Print out VTK cell type
    fin << std::endl;
    fin << "CELL_TYPES " << input.numberoftetrahedra << std::endl;
    for (i = 0; i < input.numberoftetrahedra; ++i)
      fin << "10\n";

    fin.close();

}
```

References

[1] R Broomby. New cracks in hunterston core. http://www.bbc.co.uk/news/science-environment-29481481, 2014 (accessed February 1, 2015).

[2] E Nonbøl. Description of the advanced gas cooled type of reactor (agr). Technical report, Nordisk Kernesikkerhedsforskning, 1996.

[3] SOLFEC. https://www.openhub.net/p/solfec.

[4] MoFEM. https://bitbucket.org/likask/mofem-joseph/wiki/Home.

[5] BJ Marsden, SL Fok, TJ Marrow, and Mummery PM. The relationship between strength and modulus in nuclear graphite. *In: Proceedings of HTR-2004 Beijing*, 2004.

[6] N McLachlan, RT Szczepura, MA Davies, RCB Judge, and BJ Marsden. A probabilistic approach to assessing agr core life. *In: International Conference on Life Management of Power Plants*, pages 182–191, 1994.

[7] JJ Moreau. Numerical aspects of the sweeping process. *Computer Methods in Applied Mechanics and Engineering*, 177(3):329–349, 1999.

[8] M Jean. The non-smooth contact dynamics method. *Computer Methods in Applied Mechanics and Engineering*, 177(3):235–257, 1999.

[9] T Koziara. Solfec - simplified core. https://code.google.com/p/solfec/wiki/Simplified_Core, 2010 (accessed February 1, 2015).

[10] JD Eshelby. The elastic energy-momentum tensor. *Journal of Elasticity*, 5(3-4):321–335, 1975.

[11] C Miehe, E Gürses, and M Birkle. A computational framework of configurational-force-driven brittle fracture based on incremental energy minimization. *International Journal of Fracture*, 145(4):245–259, 2007.

[12] Ł Kaczmarczyk, MM Nezhad, and CJ Pearce. Three-dimensional brittle fracture: configurational-force-driven crack propagation. *International Journal for Numerical Methods in Engineering*, 97(7):531–550, 2014.

[13] R Mackenzie. Brick slice with slot. https://bitbucket.org/likask/mofem-joseph/wiki/Brick%20slice%20with%20slot, 2014 (accessed February 1, 2015).

[14] BF De Veubeke. The dynamics of flexible bodies. *International Journal of Engineering Science*, 14(10):895–913, 1976.

[15] Ł Kaczmarczyk, T Koziara, and CJ Pearce. Corotational formulation for 3d solids. an analysis of geometrically nonlinear foam deformation. *arXiv preprint arXiv:1110.5321*, 2011.

[16] CJ Taylor and DJ Kriegman. Minimization on the lie group so(3) and related manifolds. 1994.

[17] ST Smith. Geometric optimization methods for adaptive filtering. *arXiv preprint arXiv:1305.1886*, 2013.

[18] R Mahony. *Optimization algorithms on homogeneous spaces*. PhD thesis, Australian National University, 1994.

[19] A Edelman, TA Arias, and ST Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[20] Y Ma, J Košecká, and S Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *International Journal of Computer Vision*, 44(3):219–249, 2001.

[21] RM Murray, Z Li, SS Sastry, and SS Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[22] TetGen. http://wias-berlin.de/software/tetgen/index.html.

[23] MATLAB. *Version 8.5 (R2015a)*. The MathWorks Inc., Natick, Massachusetts, 2015.

[24] VV Novozhilov. On a necessary and sufficient criterion for brittle strength. *Journal of Applied Mathematics and Mechanics (USSR)*, 33(2):212–222, 1969.

[25] J Sokołowski and A Żochowski. On the topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, 37(4):1251–1272, 1999.

[26] N Van Goethem and AA Novotny. Crack nucleation sensitivity analysis. *Mathematical Methods in the Applied Sciences*, 33(16):1978–1994, 2010.

[27] G Allaire, F Jouve, and N Van Goethem. Damage and fracture evolution in brittle materials by shape optimization methods. *Journal of Computational Physics*, 230(12):5010–5044, 2011.

[28] K Ando, BA Kim, M Iwasa, and N Ogura. Process zone size failure criterion and probabilistic fracture assessment curves for ceramics. *Fatigue & Fracture of Engineering Materials & Structures*, 15(2):139–149, 1992.

[29] ME Gurtin. *An Introduction to Continuum Mechanics*. Academic press, 1982.

[30] JR Barber. *Elasticity*. Solid Mechanics and Its Applications. Springer, 1992.

[31] T Lewiński and J Sokołowski. Energy change due to the appearance of cavities in elastic solids. *International Journal of Solids and Structures*, 40(7):1765–1803, 2003.

[32] AA Griffith. The phenomena of rupture and flow in solids. *Philosophical Transactions of the Royal Society of London*, A221:163–198, 1921.

[33] GR Irwin. Analysis of stress and strains near the end of a crack traversing a plate. *Journal of Applied Mechanics*, 24:361–364, 1957.

[34] GC Sih. Strain-energy-density factor applied to mixed mode crack problems. *International Journal of Fracture*, 10(3):305–321, 1974.

[35] FA McClintock. Ductile fracture instability in shear. *Journal of Applied Mechanics*, 25:582–588, 1958.

[36] D Gross and T Seelig. *Fracture Mechanics: With an Introduction to Micromechanics*. Mechanical Engineering Series. Springer Berlin Heidelberg, 2011.

[37] A Seweryn, S Poskrobko, and Z Mróz. Brittle fracture in plane elements with sharp notches under mixed-mode loading. *Journal of Engineering Mechanics*, 123(6):535–543, 1997.

[38] J Li and XB Zhang. A criterion study for non-singular stress concentrations in brittle or quasi-brittle materials. *Engineering Fracture Mechanics*, 73(4):505–523, 2006.

[39] A Seweryn. Brittle fracture criterion for structures with sharp notches. *Engineering Fracture Mechanics*, 47(5):673–681, 1994.

[40] A Seweryn and A Łukaszewicz. Verification of brittle fracture criteria for elements with v-shaped notches. *Engineering Fracture Mechanics*, 69(13):1487–1510, 2002.

[41] FJ Gómez, M Elices, F Berto, and P Lazzarin. Local strain energy to assess the static failure of u-notches in plates under mixed mode loading. *International Journal of Fracture*, 145(1):29–45, 2007.

[42] A Rohatgi. Version 3.3 of webplotdigitizer, 2014.

[43] F Hecht. New development in freefem++. *Journal of Numerical Mathematics*, 20(3-4):251–266, 2012.

[44] Z Yosibash. Computing singular solutions of elliptic boundary value problems in polyhedral domains using the p-fem. *Applied Numerical Mathematics*, 33(1):71–93, 2000.

[45] T Apel, D Leguillon, C Pester, and Z Yosibash. Edge singularities and structure of the 3-d williams expansion. *Comptes Rendus Mecanique*, 336(8):629–635, 2008.

[46] Z Yosibash, N Omer, M Costabel, and M Dauge. Edge stress intensity functions in polyhedral domains and their extraction by a quasidual function method. *International Journal of Fracture*, 136(1-4):37–73, 2005.

[47] E Glushkov, N Glushkova, and O Lapina. 3-d elastic stress singularity at polyhedral corner points. *International Journal of Solids and Structures*, 36(8):1105–1128, 1999.

[48] T Apel, AM Sändig, and SI Solov'ev. Computation of 3d vertex singularities for linear elasticity: Error estimates for a finite element method on graded meshes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(6):1043–1070, 2002.

[49] T Apel, V Mehrmann, and D Watkins. Structured eigenvalue methods for the computation of corner singularities in 3d anisotropic elastic structures. *Computer Methods in Applied Mechanics and Engineering*, 191(39):4459–4473, 2002.

[50] AL Mazzucato, V Nistor, and Q Qu. Quasi-optimal rates of convergence for the generalized finite element method in polygonal domains. *Journal of Computational and Applied Mathematics*, 263:466–477, 2014.

[51] TJ Tautges, C Ernst, C Stimpson, RJ Meyers, and K Merkley. Moab: a mesh-oriented database. Technical report, Sandia National Laboratories, 2004.

[52] H Si. TetGen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2):11:1–11:36, February 2015.

[53] NM Josuttis. *The C++ Standard Library: A Tutorial and Reference*. Addison-Wesley, 2012.

[54] J Chang, J Xu, and Y Mutoh. A general mixed-mode brittle fracture criterion for cracked materials. *Engineering Fracture Mechanics*, 73(9):1249–1263, 2006.

[55] J Matoušek. *Lectures on Discrete Geometry*, volume 212. Springer-Verlag New York, 2002.

[56] W Schroeder, K Martin, and B Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics (4th ed.)*. Kitware, 2006.

[57] J Ahrens, B Geveci, C Law, CD Hansen, and CR Johnson. 36-paraview: An end-user tool for large-data visualization, 2005.

[58] B Winkler, G Hofstetter, and G Niederwanger. Experimental verification of a constitutive model for concrete cracking. *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials Design and Applications*, 215(2):75–86, 2001.

[59] B Winkler, G Hofstetter, and H Lehar. Application of a constitutive model for concrete to the analysis of a precast segmental tunnel lining. *International Journal for Numerical and Analytical Methods in Geomechanics*, 28(7-8):797–819, 2004.

[60] RP Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425, 1971.

[61] WH Press, SA Teukolsky, WT Vetterling, and BP Flannery. *Numerical Recipes in C (2nd Ed.): The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[62] F Hecht. Freefem++ (3rd edition). http://www.freefem.org/ff++/ftp/freefem++doc.pdf, 2015.

[63] JA Nelder and R Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[64] M Dauge. Singularities of corner problems and problems of corner singularities. *ESAIM: ProcS*, 6:19–40, 1999.

[65] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[66] MFEM: Modular finite element methods. mfem.org.

[67] N Morozov and Y Petrov. *Dynamics of Fracture*. Springer Science & Business Media, 2013.

[68] DH Chen. Stress intensity factors for v-notched strip under tension or in-plane bending. *International Journal of Fracture*, 70(1):81–97, 1994.

[69] AA Novotny, RA Feijóo, E Taroco, and C Padra. Topological sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 192:803–829, 2003.

[70] M Lisi. Some remarks on the cantor pairing function. *Le Matematiche*, 62(1):55–65, 2007.