Numerical simulation of fracture pattern development and implications for fluid flow

Adriana Paluszny Rodriguez
Earth Science and Engineering Department
Imperial College

A thesis submitted for the degree of
Doctor of Philosophy
London 2008
to Fabiana, Omaira, Marco, and Sören
Acknowledgements

Completing a PhD is an experience of a lifetime. I would not have been able to accomplish this great challenge without the support of a range of sources. First of all, from my adviser, Prof. Stephan Matthäi, who has taught me about geology, numerical methods, programming, and the discipline of good writing. His guidance has been invaluable and motivational. Thank you for all the time invested in bringing my education to the next level. I can only hope to repay you with countless years of collaborative work to come.

I thank the Earth Science and Engineering Department at Imperial College for supporting this research through a Janet Watson Scholarship. Thank you for providing a superb environment for the development of this work and bringing together a set of highly skilled colleagues that have greatly enriched this experience.

I’d like to thank my colleagues Lateef Akanji, Hamid M. Nick, Anthony Lamb, Martin Iding, Xavier Garcia, Dr. Mandefro Belayneh, Dr. Murtaza Gulamali, Dr. Dim Comou, Dr. Ran Qi, Dr. Matthew Rhodes, Dr. Huiyun Lu, and Dr. Martin Putz, for valuable discussions, thoughtful comments, free coffees, and other important contributions. Especially, to Dr. Evren Unsal for being a friend and supportive agent throughout the last years of this effort. I’d also like to thank internship students Kelechi Ibekwe (Shell Nigeria) and Virginie Mucha (Université Bordeaux) for the commitment and dedication to our project and fruitful discussions about the propagation algorithm.

Special thanks to Prof. Martin Blunt, Prof. Chris Pain, Dr. Sebastian Geiger (Heriot Watt), Dr. Thomas Driesner (ETH Zurich), Dr. Rafi Blumenfeld (Cambridge University), and Prof. Milan Jirasek (TU Prague) for their numerous contributions to this work. The course of this project was greatly influenced by the input of our industrial partners during the semestral ITF-ISF meetings: Dr. Susan Agar (ExxonMobile), Dr. Olivier Gosselin (Total), Dr. Paul Armitage (PetroCanada), Dr. Roar Kjestadli (BP), and Dr. Chris Harris (Shell). They have witnessed the evolution of my work and have enriched it with numerous discussions about the industrial applicability of fracture propagation methods.
I’d like to thank Dr. Harry Hatzakis, for allowing me to conduct programming and writing marathon sessions at the Biotronics3D office at Trinity Buoy Wharf. Thanks to Dr. Ernesto Coto, and my colleagues from the TU Wien Dr. Stephan Bruckner, Peter Rautek, Dr. Katja Buehler, and Prof. Eduard Groeller, for their inspiration toward starting this project three years ago.

Finally, I would like to thank the women who have inspired my life: Cruz Gomez, Hanka Kluczynsky, Josefina Navas, Nayda Scialoia, Emperatriz Rodriguez, Vianey Scialoia, Hanna Moser, Ines Tibi, Martha Hasenstab, and Stephanie Sleiman for their endless support, encouragement, and inspiration. Especially to my sister, Fabiana Paluszny, my soulmate, for being the best friend one could wish - thank you for always visiting me and inspiring me with your wit. And my mother, Prof. Omaira Rodriguez, my role model, for introducing me to Computer Sciences and being an unconditional friend and adviser throughout my life. A very special thanks goes to my father, friend, and colleague, Prof. Marco Paluszny, for his dedication to my education, for introducing me to NURBS, for his interest in my work, and for providing me with true insights into the academic life.

Finally, I’d like to thank Dr. Sören Grimm for his endless support and patience throughout this journey. It is but just to recognise that life is a team effort. Thanks for being the orange that fits the half.
Abstract

Simulations are instrumental to understanding flow through discrete fracture geometric representations that capture the large-scale permeability structure of fractured porous media. The contribution of this thesis is threefold: an efficient finite-element finite-volume discretisation of the advection/diffusion flow equations, a geomechanical fracture propagation algorithm to create fractured rock analogues, and a study of the effect of growth on hydraulic conductivity. We describe an iterative geomechanics-based finite-element model to simulate quasi-static crack propagation in a linear elastic matrix from an initial set of random flaws. The cornerstones are a failure and propagation criterion as well as a geometric kernel for dynamic shape housekeeping and automatic remeshing. Two-dimensional patterns exhibit connectivity, spacing, and density distributions reproducing en echelon crack linkage, tip hooking, and polygonal shrinkage forms. Differential stresses at the boundaries yield fracture curving. A stress field study shows that curvature can be suppressed by layer interaction effects. Our method is appropriate to model layered media where interaction with neighbouring layers does not dominate deformation. Geomechanically generated fracture patterns are the input to single-phase flow simulations through fractures and matrix. Thus, results are applicable to fractured porous media in addition to crystalline rocks. Stress state and deformation history control emergent local fracture apertures. Results depend on the number of initial flaws, their initial random distribution, and the permeability of the matrix. Straight-path fracture pattern simplifications yield a lower effective permeability in comparison to their curved counterparts. Fixed apertures overestimate the conductivity of the rock by up to six orders of magnitude. Local sample percolation effects are representative of the entire model flow behaviour for geomechanical apertures. Effective permeability in fracture dataset subregions are higher than the overall conductivity of the system. The presented methodology captures emerging patterns due to evolving geometric and flow properties essential to the realistic simulation of subsurface processes.
# Contents

1 Introduction

2 Hybrid finite element-finite volume discretisation of fractured rocks
   2.1 Abstract ......................................................... 7
   2.2 Introduction .................................................. 8
   2.3 Methodology .................................................... 11
      2.3.1 Step 1: Building the geometric model with CAD ............ 12
      2.3.2 Step 2: Generating the hybrid finite element mesh ........ 14
      2.3.3 Step 3: FE-FV discretisation of the governing equations ... 18
      2.3.4 Finite element-finite volume stencils ....................... 20
      2.3.5 Volume and surface integration of finite volumes .......... 22
      2.3.6 Vector and integral transformations .......................... 26
      2.3.7 Facet area .................................................. 26
      2.3.8 Normal orientation ......................................... 27
      2.3.9 Integral computations in physical space .................... 28
      2.3.10 Step 4: Tracer advection in fractured porous rock .......... 29
   2.4 Results ........................................................ 32
      2.4.1 Mesh refinement and increasing geometric complexity ....... 32
      2.4.2 Step 5: \textit{A posteriori} mesh adaptation guided by finite element
                        discretisation error ...................................... 35
      2.4.3 Finite-volume integration error ............................. 38
      2.4.4 Application of implicit first-order transport scheme to a discrete
                        3D fracture model .......................................... 39
   2.5 Discussion .................................................... 41
   2.6 Conclusions .................................................... 42

3 Fracture Propagation
   3.1 Abstract ......................................................... 44
   3.2 Introduction .................................................. 44
   3.3 Governing Equations ........................................... 49
      3.3.1 Fracture representations .................................... 50
      3.3.2 Displacements, stresses, and strains ....................... 52
3.3.3 Discretisation ........................................... 53
3.3.4 Isoparametric quadratic triangles ....................... 54
3.3.5 Failure and propagation criteria ......................... 56
3.3.6 Propagation angle ..................................... 59
3.3.7 Stress intensity factors ................................ 60

3.4 Fracture growth methodology ............................. 62
3.4.1 Propagation algorithm .................................. 64
3.4.2 Initial flaws ............................................ 65
3.4.3 Geometric handling of fracture propagation ............ 67
3.4.4 Fracture tips ........................................... 68
3.4.5 Fracture advance ....................................... 68
3.4.6 Fracture intersection ................................... 70
3.4.7 Fracture closure ........................................ 72
3.4.8 Algorithm implementation ............................... 73

3.5 Adaptive remeshing ........................................ 73
3.5.1 Element quality-based mesh adaptivity ................. 74
3.5.2 Fracture-to-mesh and mesh-to-fracture .................. 76
3.5.3 Mapping between meshes ................................ 79

3.6 Fracture characterisation ................................... 81

3.7 Testing and benchmarks ..................................... 82
3.7.1 Validation of the elasticity kernel ....................... 83
3.7.2 Validation and verification of the crack propagation kernel 83
3.7.3 Benchmarking eight datasets ............................ 85
3.7.4 Performance analysis ................................... 89

3.8 Fracture growth applications ............................... 91
3.8.1 Tension: 10 fractures .................................. 91
3.8.2 Tension: 200 fractures ................................ 92
3.8.3 Effect of local material heterogeneities ................ 92
3.8.4 Effect of initial random flaws: orientations and spacing 96
3.8.5 Shrinkage .............................................. 100

3.9 Extensions of the method ................................... 102
3.9.1 Including dynamic effects .............................. 102
3.9.2 Non-linear material constitutive laws ................... 103
3.9.3 Crack propagation in 3D ................................ 104

3.10 Conclusions ................................................ 105
List of Figures

2.1 Example of mapped fracture data ...................................... 9
2.2 Faceted versus NURBS representation ................................ 13
2.3 Boolean operations ....................................................... 14
2.4 Mesh types: rectilinear, curvilinear and hybrid .................... 15
2.5 Mesh artifacts ............................................................. 16
2.6 3D Hybrid element mesh ............................................... 16
2.7 Fracture discretisation examples ..................................... 17
2.8 FE-FV 1D Stencil ......................................................... 19
2.9 FE-FV 2D Stencil .......................................................... 21
2.10 FE-FV 3D Stencil ........................................................ 22
2.11 Integration points and weights (I) ................................... 24
2.12 Integration points and weights (II) ................................... 25
2.13 Random generated fractures .......................................... 33
2.14 Mesh properties as a function of size ................................ 35
2.15 Error estimation of mesh ............................................. 37
2.16 Directional element length ............................................ 38
2.17 3D Transport example ................................................ 40

3.1 Example of fractures in nature ........................................ 45
3.2 Stochastically generated fractures .................................... 46
3.3 Geomechanically generated fractures ................................ 49
3.4 Basic failure model types .............................................. 51
3.5 Triangular isoparametric quadratic element ......................... 55
3.6 Single crack propagation ............................................... 57
3.7 Numerical simulation of hooking fracture tips ..................... 60
3.8 Quarter point displacement technique ............................... 61
3.9 Quarter point elements around a tip .................................. 62
3.10 Fracture tip interaction in diatomite and concrete ............... 63
3.11 Random flaw distributions ............................................ 66
3.12 Polygonal representation of a crack ................................. 68
3.13 Extension of the fracture polygon ................................... 69
3.14 Fracture intersection detail ............................................ 71
4.9 Stress contours around fractures embedded in a three-dimensional layer 116
4.10 Stress field around fractures in a three-dimensional single layer . . . . 117
4.11 Stress field around fractures in a three-dimensional double layer . . . 118
4.12 Stress field around fractures in a three-dimensional sandwich layer with
differential stress ......................................................... 119
4.13 Straight versus curving paths: tensile stress ............................. 121
4.14 Straight versus curving paths: isotropic shrinkage ...................... 122
5.1 Snapshots of fracture growth ........................................ 130
5.2 Fracture centreline and aperture approximation .......................... 131
5.3 Effective permeability boundary conditions ............................. 131
5.4 Initial and final steps of a growth simulation ............................ 133
5.5 Fracture connectivity .................................................. 136
5.6 Cluster size ............................................................ 137
5.7 Fracture density ........................................................ 138
5.8 Spacing and length ..................................................... 139
5.9 Fracture length distribution as a function of growth ..................... 140
5.10 Flaw area and apertures as a function of growth ......................... 141
5.11 Flaw area initial apertures ............................................ 142
5.12 Aperture distribution .................................................. 143
5.13 Flaw area aperture bins .............................................. 144
5.14 Aperture distribution .................................................. 145
5.15 Observation area A: effective permeability before percolation .......... 146
5.16 Observation area B: effective permeability before percolation .......... 146
5.17 Observation area A: effective permeability ............................ 147
5.18 Observation area B: effective permeability ............................ 147
5.19 Fracture-matrix flux ratio ............................................ 149
5.20 Velocity variation within fractures .................................... 150
5.21 Average velocity variation within fractures ............................ 151
5.22 Velocities ............................................................. 152
5.23 Flaw area versus observation area .................................... 153
5.24 Straight fractures ..................................................... 154
5.25 Flaw area straight fracture pattern: effective permeability .......... 155
5.26 Curved versus straight: effective permeability ......................... 156
5.27 Curved versus straight: fracture-matrix flux ratio ...................... 157
5.28 Pattern anisotropy: fracture measurements ............................. 159
1 Introduction

Fractured carbonate reservoirs currently trap 60% of proven oil reserves and 40% of the world’s gas reserves [BP, 2007]. Studying fractures in porous media is important because they significantly influence the flow properties of rock masses [Ingebritsen and Sanford, 1998]. Numerical simulations are instrumental in the engineering of these complex and heterogeneous reservoirs due to the geometric and physical complexity of fracturing and flow phenomena [Granet et al., 1998; Bastian et al., 2000; Bogdanov et al., 2003, 2007; Matthai and Belayneh, 2004; Matthai et al., 2007]. Simulating these processes requires models that capture emerging flow and geomechanical patterns in a flexible, accurate, and efficient manner.

Geometric characterisation of fractured reservoirs is a key unresolved aspect of reservoir modelling. There are three main sources of data for their geometric characterisation [Berkowitz, 2002]: geological mapping, geophysical imaging, and borehole data. Geological mapping typically relies on manual tracing of field fractures including length, aperture, spacing, and shape measurements [e.g. Segall and Pollard, 1983]. Characterisation can be aided by means of artificial intelligence neural networks that automatically identify fractures and faults from field photos [Zellou and Ouenes, 2003; Zellou et al., 1995]. Geophysical techniques such as ground penetrating seismic imaging are used to describe stratigraphic beds and faults at depth [e.g. Jouanna, 1993]. Borehole data measure flow properties, such as hydraulic conductivity, while collecting aperture and orientation data of intersecting fractures [e.g. Novakowski and Bickerton, 1997]. These techniques all focus on generating reservoir proxies and probability distributions that describe the geometry and topology of the occluded subsurface fractures.

Seismic data is often interpreted to reconstruct large-scale fault and bedding models [Caumon and Mallet, 2006; Pepper and Bejarano, 2005]. Faults and stratigraphic boundaries are ideally represented using smooth parametric surfaces, such as NURBS [Matthai et al., 2005a; Paluszny et al., 2007]. Additionally, data collected from geological mapping and boreholes are used to generate stochastic fracture datasets. This modelling technique captures statistics by generating sets of planar fractures of random orientation, size, and location [Dershowitz and Einstein, 1988]. Its main advantage is its speed and robustness in generating two- and three-
dimensional fracture datasets. However, pure stochastic approaches make various simplifying assumptions:

- when using mapped data to generate stochastic datasets we assume that they are analogue to subsurface patterns. Thus, we assume the source outcrops were formed under similar conditions, and, therefore, display similar topological and geometric characteristics;

- when using borehole data to characterise the fracture set, we assume that the measured data is representative of the entire fractured rock mass, as opposed to only sampling its local geometry;

- fractures are assumed planar, elliptically shaped, stand-alone features, as opposed to free form intersecting surfaces that form in nature;

- local boundary stresses as determined by local fault interaction are not taken into account, although they are key to fracture pattern formation [Pollard and Aydin, 1988];

- fracture interactions are neglected, although field observations have shown that they play an important role in fracture pattern formation;

- and, apertures are assumed constant or proportional to length, as opposed to being conditioned by the history of deformation and present stress state.

Various alternatives coexist with the pure stochastic approach. For example, Srivastava et al. [2005] developed a sophisticated data-informed geostatistic model that honours geometric field data by pseudo-randomly growing cracks. Rives et al. [1994] devised a set of rules to generate fracture sets based on probability and geological constraints including strain and curvature attributes. In contrast to statistic based approaches, full geomechanical modelling reconstructs the geometry of a system based on the analysis of the events that formed it [Nelson, 1985]. It seeks to reproduce topology and connectivity of fracture networks as well as fracture sizes, apertures, and patterns that arise due to growth, interaction, and coalescence.

Early studies in civil engineering focused on criteria to predict failure of materials and structures. Inglis [1913] and Griffith [1921] recognised that tips of cracks act as stress concentrators inducing cracks to propagate as a response to external load. They identified that the local work due to crack propagation is a function of the free energy of the surfaces that form during cracking. Neuber [1937] pointed out that this stress
concentration is proportional to the length of the crack and the curvature of its tip. Irwin [1948] further extended this conceptual model by identifying that local work due to growth is significantly greater than the surface free energy. This phenomenon was later identified as unstable crack growth. Independently, Obreimoff [1930] identified quasi-static crack growth as a conceptual model that prescribes the slow and steady growth of cracks under equilibrium. Gurney and Hunt [1967] identified that, in this case, the energy required to propagate fractures varies inversely as the root of their size.

Over the last century, numerous failure criteria have been devised to describe the triggering of fracturing due to stress concentrations including: Rankine, Mohr-Coulomb, Hoek-Brown [Hoek and Brown, 1980; Hoek, 1983], Drucker-Prager [Drucker and Prager, 1952], Mogi [Mogi, 1971], and their generalisations, derivations, and combinations [e.g. Bigoni and Piccolroaz, 2004]. However, experiments [e.g. McLean and Addis, 1990; Vernik and Zoback, 1992] have shown that their applicability is restricted to specific material and in-situ stress conditions of the rock [Al-Ajmi, 2006; Al-Ajmi and Zimmerman, 2005]. Furthermore, failure criteria cannot predict the complex crack paths that originate during propagation due to interaction with neighbouring cracks [Brace and Bombolakis, 1963]. Failure of a specimen is rarely given by the propagation of a single crack, instead it is triggered by the coalescence of multiple aligned cracks that form during deformation [Hoek and Bieniawski, 1965]. Since then, a variety of numerical methods have been developed to capture irreversible deformation of rock masses [Nayak and Zienkiewicz, 1972; Cervenka, 1970; Jirasek and Zimmermann, 1998; Jirasek, 1997; Yang et al., 2005; Bazant et al., 1996; Bazant and Ozbolt, 1990; Bazant et al., 1999; Bazant and Prat, 1988].

Geologists studying fracture propagation focused on identifying propagation criteria that describe how a crack propagates as a function of the energy accumulated around its tips [e.g. Olson and Pollard, 1989; Renshaw and Pollard, 1994b]. The main idea was to develop techniques to simulate the on-going fracturing and study evolving patterns. As a result, a variety of geometric-based propagation methods have been devised to simultaneously grow multiple fractures [Ingraffea and Saouma, 1985; Olson and Pollard, 1989; Olson, 1993; Renshaw and Pollard, 1994b; Belytschko and Black, 1999; Huang et al., 2003]. Renshaw and Pollard [1994b] rely on the analytical description of the stress field around multiple straight cracks to estimate growth. However, analytical solutions are not applicable to arbitrary crack geometries. Olson and Pollard [1989] applied the boundary element method to numerically compute the energy release rates and estimate growth of a set of straight cracks. Later, Olson
[1990] and Olson et al. [2001a] published extensions of this method to curved crack paths. Ingraffea and Saouma [1985] introduced a finite element based method for crack propagation. Belytschko and Black [1999] developed the extended finite element method, in which cracks are kept track of independently of the mesh. Finally, Huang et al. [2003] demonstrated the applicability of this method to the simultaneous growth of multiple cracks. The previous studies gave rise to two main approaches for fracture analysis: discrete and smeared, also known as geometric/non-geometric or grid/sub-grid methods. Each has a domain of application: the first are appropriate to simulate one or more dominant cracks, while the second are designed to model diffuse cracking patterns that arise due to the heterogeneity of rocks and other quasi-brittle materials [de Borst et al., 2004].

The main disadvantage of all these mechanical approaches is the amount of resources needed to perform the simulations. For thousands of fractures, ensuing multi-million node datasets require huge amounts of memory and time. However, with increasing computational resources becoming available, these techniques have become viable for the study of the effect of fractures on subsurface fluid flow [Ingraffea and Saouma, 1985; Olson and Pollard, 1989; Olson, 1993; Renshaw and Pollard, 1994b; Belytschko and Black, 1999; Huang et al., 2003].

Realistic simulation of flow behaviour is challenging because of the complexity inherent to the geometric handling of multiple fractures as simultaneously growing entities and the domain differentiation they ensue. A model capable of capturing emerging patterns that arise during the simulations must:

- be suitable for arbitrary geometries;
- have an independent discretisation of the fracture and rock domains;
- adapt geometry to deformation;
- adapt the discretisation to the varying geometry and solution field;
- propagate fractures at the onset of failure at their tips;
- depend on measurable input parameters that do not require specific material calibration;
- dynamically compute aperture variations due to in-situ stresses;
Fracture datasets are used as reservoir proxies for the simulation of fluid flow through fractures, transport in porous media, and other multi-physics simulations [Sternlof et al., 2006; Bogdanov et al., 2007; Matthai et al., 2005a, 2007]. Such simulations are of economic importance because fractures in these resource-rich rocks induce permeability variations that range over several orders of magnitude [Matthai et al., 1998; Sternlof et al., 2006]. Bogdanov et al. [2003] identified that to quantify the effect of fracture density and connectivity on flow, we must simulate flow through matrix and fractures simultaneously. Therefore, both fracture and matrix domains need to be discrete in order to address individual fracture and matrix properties and avoid simplifying assumptions.

Previous work in this field includes the following. Granet et al. [1998] simulated single-phase flow on two-dimensional discrete fractures. Bastian et al. [2000] published a method that discretises fracture and matrix domains by generating two independent meshes and simulates multi-phase flow using the finite volume method. Geiger et al. [2004] published a method for multiphase flow modelling on highly heterogeneous and structurally complex geologic media including multiple discrete fractures and faults. Matthai et al. [2007] simulated multi-phase flow in fractured media using a finite element-node-centred finite-volume approach on unstructured hybrid-element meshes. Paluszny et al. [2007] described a stencil-based method for advection-diffusion modelling on structurally complex fractured media. The main disadvantage of this rigorous approach is, again, efficiency because large amounts of memory and computational time are required to run the simulations. However, accelerating computational resources that range from multi-core desktops to high-performance computing facilities with hundreds to thousands of cores render this disadvantage away. For example, Coumou et al. [2008] recently presented a parallel application of the same numerical scheme to solve fluid flow in complex geologic media. However, the previous methods rely on fractures that are either mapped or randomly generated. Thus, they cannot study flow behaviour as a function of growth.

The aim of this study was to develop a numerical method, which can efficiently deal with the simultaneous propagation of multiple fractures and can be applied to address fluid flow problems. As suggested by Ingraffea and Wawrzynek [2004] we have developed a methodology that allows for interchangeable modules for failure and propagation criteria that can be assembled together to form problem-specific simulations. The failure criteria specifies which cracks will extend while the propagation criterion determines how they propagate [Lawn, 1993]. A discrete representation of matrix and fracture domains allows for flow simulations to be
directly conducted using the same discretisation. We present a linear-elastic fracture mechanics-based, adaptively meshed, FEM approach to investigate fracture pattern formation based on a local failure criterion. The proposed multi-fracture propagation algorithm applies to quasi-static growth [Paluszny and Matthai, 2008b]. It simultaneously grows a set of discrete fractures within a matrix and keeps track of their domains separately. Fractures grow from a set of initial flaws subjected to displacements applied at the model boundaries. A geometry kernel is used to keep track of intersections and potential closure. The output is an evolving fracture pattern, including aperture distribution and stress state. The main contribution is the integration of the geometry-based, discrete propagation method with existent multi-physics methods for flow simulation. We showcase the combination of our discretisation techniques with the geomechanical generation of fractured datasets. We measure hydraulic conductivity as a function of growth, connectivity, and density [Ibekwe, 2007; Paluszny and Matthai, 2008a], and show that simplifying assumptions such as fixed apertures, planar fractures, and model under-sampling yield measurements that range over several orders of magnitude.

Benchmarking verifies that the numerical methods and propagation algorithm, implemented in the object-oriented code Complex System Modelling Platform (CSMP) [Matthai et al., 2001], perform adequately for various sub-problems of fracture set formation.

This thesis is organised as follows. Chapter 2 describes the finite element - finite volume discretisation scheme used for solving diffusion/advection equations. Chapter 3 focuses on the fracture propagation algorithm. It explains the theory behind the simulator: discrete fracture propagation using the finite element method. It also reviews the details of the computational geometry algorithms that were developed for the housekeeping of the fracture set geometries. Chapter 4 describes the applicability of the propagation method to layered media. Chapter 5 introduces the effective permeability and fracture-matrix flux ratio as a flow property of interest and focuses on the effects of growth and stress state on the conductivity of rock masses. It is followed by a discussion of the contributions of this work and a summary of the conclusions and future work. Finally, appendix B contains the User’s Guide for the CSMP mechanical module and appendix C contains an reference guide to the generated source code.
2 Hybrid
finite element-finite volume
discretisation of fractured rocks

2.1 Abstract
The generation of computational meshes for complex geological objects is a challenge: their shape needs to be retained, resolution has to adapt to local detail, and
solution shape and strong variation of material properties across volumetric domains have to be taken into account. We present an unstructured hybrid finite-element, node-centred finite-volume discretisation suitable for fluid flow, reactive transport, and mechanical partial differential equations solved on a complex geometry with inhomogeneous material domains. We show that resulting meshes accurately capture free-form material interfaces as described by non-uniform rational B-spline curves and surfaces. The resulting discretisation error is analyzed for the elliptic pressure equation and we introduce an error metric to guide mesh refinement. Finite-elements and finite-volumes are represented in parametric space where integrations are conducted numerically. Subsequently, integral properties are mapped to physical space using Jacobian transformations. This method retains its validity when the mesh is deformed. The resulting generic formulation is demonstrated with a transport calculation performed on a complex discrete fracture model.

2.2 Introduction

Recent advances in computer software and hardware capabilities allow hydro-mechanical computations to be conducted with an unprecedented degree of physical realism. The definition of model geometry in three dimensions remains a challenging problem. Due to the irregular shape of material domains and their heterogeneity, the governing partial differential equations cannot be solved analytically and numerical methods are used to find approximate solutions. These methods approximate unknown functions of multiple variables by integration of simpler sub-equations for the coupling among discrete points. The discretisation derives from the spatial subdivision of the domain of interest into a mesh of geometrically primitive cells. Contributions from the sub-equations combine into a system of algebraic equations constraining the domain solution.

The first step of any simulation workflow is the adequate geometrical representation of the model. The introduction of graphical tools suitable for the geologic interpretation of seismic data, maps, and cross-sections from mines has facilitated the generation of rigorous three-dimensional structural geologic models. The migration of these complex interpretations into internally consistent simulation models, without loss of crucial geological detail, constitutes an important new field of applied research. The representation of faults and fractures is especially challenging as they have a large aspect ratio, displace layers relative to one-another, and display complex cross-cutting relationships. Figure 2.1 (a) shows an example of a complexly fractured fold exposed
at Kilve beach, UK. Typically, fractures have a variable aperture, intersect at small angles, and range in size over several orders of magnitude.

Figure 2.1: Geologic complexity of an outcrop of Liassic limestone on the southern Bristol Channel coast, UK. Lid of camera for scale has 5 cm diameter. (a) Photograph of small-scale fold structure intersected by carbonate veins crosscut by later Alpine joints; (b) free-form NURBS CAD model of the same structure with numerous intersection curves and acute intersection angles between fractures. Photo is a courtesy of Dr M. Belayneh, Imperial College London.

Commonly used polyhedral representations of geological models are sufficient for visual display, but unnecessarily constrain volume discretisation. Severe discretisation problems arise at the intersections of multiple polyhedral surfaces [Bogdanov et al., 2003]. In contrast to differentiable curves and surfaces polygons and polyhedra force the mesh to honour predefined node points constraining model resolution. Additionally, such models are often characterised by a lack of topological information: geometric entities are not properly connected, intersection curves are often absent, and models contain occluded gaps, holes, and overlaps. Meshing algorithms, however, require water-tight models with internally consistent space subdivisions. For these a mesh can be generated, and simulation may begin.
We introduce a new rigorous workflow for the simulation of flow and transport in deformable fractured rocks using a hybrid finite-element (FEM) finite-volume method (FVM). The three obstacles to overcome before this FEFVM method can be applied to large complex fractured geometries, are: (1) the representation of the flow geometry by an internally consistent computer-aided design model; (2) the efficient, at least semi-automatic, finite-element discretisation of the latter; and (3) the construction of a complementary and internally consistent finite-volume mesh.

A growing body of simulation results highlights the potential of unstructured mesh-based simulations for the understanding of complex subsurface processes [Helmig, 1997; Bogdanov et al., 2003; Kim and Deo, 2000; Karimi-Fard et al., 2004; Matthai et al., 2005a]. Finite-element methods dominate heat transfer, mechanical and fluid dynamics modelling [e.g. Chung, 2002]. They operate on unstructured meshes by default. Finite-volume methods are applied to solve transport problems on both regular grids and unstructured meshes. Geometrically complex fluid flow problems have been solved by combining FEM with FVM, in an operator splitting approach where the elliptic parts of a partial differential equation are solved with FEM and the hyperbolic ones with FVM [Durlofsky, 1994; Helmig, 1997; Bogdanov et al., 2003; Karimi-Fard et al., 2004; Leveque, 2003; Reichenberger et al., 2006; Matthai et al., 2005a]. FEM and FVM have much in common: they are integral formulations, use shape functions for interpolation, and are applicable to unstructured meshes. This chapter summarises the contribution of our method: a suite of new generic finite element-based finite-volume stencils defined in parametric space for tetrahedral, prism, pyramid, hexahedral, quadrilateral, triangular, and bar elements. A representation of the finite-volume mesh in physical space is not required. Jacobian transformations are used to map properties from parametric to physical space. Well-established integration techniques are employed for the efficient processing of the stencils. The FEFVM derived and employed in this work is an extension of Baliga and Patankar’s (1980) original formulation for the solution of the advection-diffusion equation. This method is embedded into a novel simulation workflow that begins with the Computer Aided Design (CAD) model and ends with the efficient numeric solution of the governing equations by using the algebraic multigrid method [Ruge and Stüben, 1987; Stüben, 2001].
2.3 Methodology

Modelling and discretisation of complex geological objects are intricate procedures by which approximations and simplifications are introduced [Ewing and Spagnuolo, 2003]. Power-law fracture-aperture and diameter distributions, complicated cross-cutting relationships, and curved layer boundaries require a spatially variable model resolution. To assign rock properties material domains are necessary. Part 1 of this method section addresses such model building issues.

Once the model has been constructed, numerical solution methods for the governing equations demand its spatial discretisation. This is the subject of the second part of this method section. We first discuss the generation of a static mesh concerned primarily with the model’s accurate geometric representation. Then we introduce the finite-volume stencils and apply them to the first-order discretisation of the advection - diffusion equation.

Individual steps of our simulation workflow facilitated by these methods are summarised as follows:

1. Representation of the geologic inventory by non-rational uniform B-splines (NURBS). This boundary representation (BREP) is explained further below. Fractures and faults are represented using equal or lower dimensional geometrical BREP approximations.

2. Discretisation of the model with a hybrid finite-element mesh composed of triangles, quadrilaterals, tetrahedra, hexahedra, prisms, and pyramids. High aspect ratio features like thin layers, faults or fractures are discretised with large aspect ratio prism and hexahedral elements.


4. Trial simulation.

5. Adaptation of the geometrically conforming meshes to solution shape.

Topological information, such as the grouping and facing directions of surfaces delimiting a volume, is mapped onto the mesh and retained throughout the simulation process. Material domains persist through the calculations and can be visualised individually to gain insights into the behaviour of the system.
2.3.1 Step 1: Building the geometric model with CAD

Geologic modelling relies on accurate representations of the subsurface at various levels of detail, including features that may differ in size over several orders of magnitude. For instance, a sand-shale sequence may contain hundreds of large aspect ratio layers truncated and offset along fault discontinuities. This calls for a framework that supports scalable geometrical entities such as parametric curves and surfaces [Farin, 2002].

For forty years, smooth parametric curves and surfaces have been used to define boundaries of free-form objects in CAD [Farin, 2002]. Non-uniform rational B-spline (NURBS) curves and surfaces allow accurate representation of simple polylines, conic sections, and free form objects [Farin, 2002]. Storage requirements for NURBS are modest and NURBS allow local control during generation and modification. Local control means that point editing only affects the immediate neighbourhood of a point so that surfaces which simultaneously match multiple boundaries can still be reshaped. Therefore, NURBS meet the two most important criteria of geometrical representation: flexibility - the ability to initially represent a feature, and fidelity - the ability to represent the feature throughout modifications [Ellens, 1997]. By modelling geological objects with NURBS we are able to capture them with a tolerance-based level of detail, independent of scale. They do not prescribe a specific resolution as is the case for faceted representations (Figure 2.2).

Faceted representations based on polylines and surface triangulations are flexible and easy to use, but they introduce discontinuities into the model that are not present in the original geometry, see Figure 2.2. By contrast, NURBS are differentiable, smoothly representing shapes as continuous features without the polyline disadvantages. Meshing of tip-lines of polygonal fractures, for example, will produce an element node at each polygon vertex while NURBS fracture representations will allow a purpose dependent adaptation of the mesh to the smooth geometry.

Using NURBS, volumetric objects are defined by grouping curve-delimited surfaces together to define the boundary of a volume. This widely used technique is called the boundary representation (BREP) referring to a hierarchical, internally consistent tree structure of points (nodes), holes and surfaces (loops), and surface-enclosed volumes (bodies), recording their relations to each other. This structure defines the topology of each body, facilitating Boolean operations, such as union, intersection, and difference (Figure 2.3).
Figure 2.1 illustrates a complex geological object and its NURBS-BREP representation. This fractured and folded layer of limestone is exposed at the Bristol Channel coast, UK. It is of special interest because it serves as an analogue of a fractured hydrocarbon reservoir [Al-Mahruqi, 2001; Belayneh and Cosgrove, 2005]. Fractures were traced with NURBS curves and extruded into surfaces orthogonal to the layering. Rock matrix blocks are delimited by NURBS surfaces forming a composite boundary enclosing material domains. This set of objects represents the macroscopic geometry accurately.

Preparation and repair of CAD models for meshing is very time consuming. Volume overlaps make a unique discretisation of space impossible such that labour intensive geometric model healing is required [Beall et al., 2003]. Artifacts, including gaps, hamper the mesh generation process and must be avoided at the CAD model construction stage. BREP significantly reduces potential inconsistencies within a model and is an effective tool in producing boundary conforming, watertight, and
2.3.2 Step 2: Generating the hybrid finite element mesh

CAD geometry can be discretised with three types of meshes: rectilinear, curvilinear and unstructured, see Figure 2.4. Rectilinear grids have a fixed resolution. The refinement they require in order to track material interfaces that are not aligned with the coordinate axes is prohibitively high, especially if these interfaces are curved. Yet, their accurate resolution is very important because material properties, like permeability, can vary by several orders of magnitude across them. Here, curvilinear and unstructured grids can provide the necessary adaptive resolution. Structured curvilinear grids, also known as O-grids, are able to capture free-form objects by mapping curves and surfaces to topologically cubic blocks in parametric space.
However, even for geometrically simple models, this subdivision requires significant manual intervention. Therefore, curvilinear grids are ill-suited for the discretisation of complex geological models [cf. Owen, 1998].

![Rectilinear, Curvilinear, and Unstructured Meshes](image)

**Figure 2.4:** Three common types of meshes: (a) rectilinear, (b) curvilinear and (c) unstructured.

We advocate unstructured grids because they can track free-form geometrical entities, such as NURBS with spatially variable refinement, and can also be generated automatically. The disadvantage of unstructured grids is that mesh coordinates cannot be calculated from indices. They therefore must be stored. However, in practice, this does not significantly increase memory requirements because the majority of storage is taken up by the discretised physical variables.

Traditional unstructured mesh generation approaches, such as Delaunay and advancing front, require watertight geometric models because they anchor any volume on the surface mesh forming its boundary. Alternative octree-based mesh generation algorithms perform well on slightly imperfect models. Factors which still compromise mesh quality are gaps and overlaps, faces with narrow angles or small areas, and dangling edges (Figure 2.5). We refer the interested reader to the reviews by Owen [1998] and Lo [2002].

Pure hexahedral meshes tend to be more compact and have better computational properties than tetrahedral meshes of the same order [Benzley et al., 1995; Blacker, 2000]. However, automatic generation of pure hexahedral unstructured meshes for free-form geometry is extremely difficult. While a number of new robust algorithms for hexahedral meshing have been proposed [e.g. Owen, 1998], these methods cannot handle arbitrary geometry without intensive manual intervention. As a good compromise between the complexity of all-hexahedral meshing and
the simplicity of automatic tetrahedralisation, hexahedra-dominated hybrid element meshes with tetrahedra, prisms, and pyramids have become increasingly popular in the finite-element community [Huber and Helmig, 2000; Khawaja and Kallinderis, 2000; Reichenberger et al., 2006; Matthai et al., 2005a]. A typical hybrid mesh contains hexahedral elements in geometrically unconstrained regions, while the more shape-adaptive tetrahedral elements are used to capture geometric complexities and intentional refinement variations, see Figure 2.6. Pyramid and prism elements are introduced to connect hexahedra with tetrahedra.

Hybrid element meshes can be generated automatically from tetrahedral meshes. This indirect approach starts with the generation of a pure tetrahedral mesh. It is partially converted to hexahedra by merging and splitting of elements and the introduction of prisms and pyramids [Zgainski et al., 1996; Thompson et al., 1998].
To evaluate the quality of the resulting mesh, the element to node ratio can be used. For realistic hybrid meshes of free-form geometry we will show in the result section that a ratio close to two can be obtained for hybrid meshes as compared with 5-6 for pure tetrahedral meshes [Bogdanov et al., 2003].

Fractures and faults in three-dimensional models may be represented either by surfaces [e.g. Reichenberger et al., 2006] or by volumes [e.g. Matthai et al., 2005a]. Our workflow supports both approaches. Using a lower dimensional representation increases efficiency and is flexible enough to represent the characteristics of the fractures implicitly [Kim and Deo, 2000; Monteagudo and Firoozabadi, 2004]. However, their approach is permissible only when the fractures are more permeable than the rock. This excludes sealed fractures and faults (Figure 2.7 (a)). Moreover, numerical simulation of capillary pressure-driven fluid exchange between fractures and rock matrix requires multiple degrees of freedom across the fracture in order to represent sharp-gradational or discontinuous saturation variations. Here volumetric meshes come into play.

![Discrete fracture representation](image1)
![Layered fracture representation](image2)

**Figure 2.7:** Photograph of a carbonate vein in Jurassic limestone at Kilve, UK. (a) 2D NURBS model using a curve with a thickness attribute discretised by line elements surrounded by triangular elements representing the rock matrix. (b) Areal representation of vein by triangles and quadrilaterals. Inset shows comparatively inefficient triangle mesh.

While tetrahedral automatic meshing performs well for isometric material domains, large aspect-ratio geological structures like thin strata, faults, or fractures, attract a large number of very small elements, see inset Figure 2.7 (b). Some incremental meshing algorithms produce highly distorted tetrahedra leading to large finite-element interpolation errors [Thompson et al., 1998]. Others avoid these errors.
by subdividing elements into smaller, better shaped tetrahedra based on geometrical measures such as aspect ratio and minimum internal angle [Liu and Joe, 1994]. This leads to the prohibitively large number of elements. We circumvent these problems using fracture-aligned high aspect ratio prism and hexahedral elements. These perform well even if they have a large aspect ratio [Khawaja and Kallinderis, 2000]. This dramatically reduces the number of elements required to represent fractures. We achieve this discretisation by extruding surface meshes into volumes composed of high aspect ratio prism elements. Figure 2.7b displays a cross section of a fracture that has been discretised with prisms and hexahedra. Multiple element layers provide the desired internal nodal degrees of freedom. Tapering of prism layers near fracture tips elegantly models pointy terminations.

All hybrid meshes generated for flow computation were generated, post-processed, and optimised using the Ansys ICEM Tetra Mesher.

2.3.3 Step 3: FE-FV discretisation of the governing equations

The main idea behind the finite-element method [Courant, 1942; Zienkiewicz and Cheung, 1965] is that an unknown continuous function, $\Psi(x, y, z, t)$, is modeled by the interpolation functions, $N_i$, defined in a piecewise fashion inside each finite element. To capture incremental changes of $\Psi$, integrals over their spatial derivatives, $\nabla N_i$, are accumulated over the domain of interest - element by element - into a system of algebraic equations, $A$. The solution of this matrix equation gives approximate values of $\hat{\Psi}$, but only at the finite-element nodes. Unique values of $\hat{\Psi}$ can only be found if an integration constant is specified. To this end, $\hat{\Psi}$ is prescribed at some node points in the boundary condition vector, $b$. Figure 2.8 shows a one-dimensional mesh where each finite element is represented by a line connecting two nodes. Two linear interpolation functions, $N_i$ and $N_j$, model $\Psi$ at any point, $X$, inside each finite element:

$$\tilde{\Psi}(X) = \sum_{i=1}^{2} N_i(X) \hat{\Psi}_i \quad X \in [n_1, n_2] \quad (2.1)$$

Since they are linear, their derivatives are piecewise constant functions and are most accurate at the element centre, but discontinuous across, and therefore undefined at the nodes (Figure 2.8). This also applies to fluid pressure derivatives, $\nabla \hat{p}$, computed to calculate Darcy velocity, $v$. As will be shown further below, flux computations for transport modelling require integration of the discontinuous velocity field. This is accomplished with node-centred finite volumes bounded by faceted surfaces. Vector
quantities, like \( \mathbf{v} \), are projected onto facet normals, \( \mathbf{n} \). These dot products are multiplied by facet area and summed over all facets of the finite volume to complete the surface integral. Remember Green’s theorem about the equivalence of surface and volume integrals:

\[
\oint_{FV} \langle \mathbf{v}, \mathbf{n} \rangle \, dS = \int_{FE} \nabla \cdot \mathbf{v} \, dV
\]  

(2.2)

where \( \mathbf{n} \) and \( \mathbf{v} \) represent the unit normal field and the Darcy field, respectively. This approach elegantly avoids using the discontinuous \( \nabla N \) derivatives at the finite element boundaries (Figure 2.8). Instead, surface integrations are performed inside the finite elements where \( \nabla N \) is continuous.

Figure 2.8: Finite element node-centred finite volume discretisation in one dimension. The variation of the solution variable \( \Psi(x) \) is approximated by \( \hat{\Psi}(x) \) using piecewise linear interpolation functions \( N_i \) and \( N_j \) associated with the nodes of each finite element; \( \partial \hat{\Psi}/\partial x \) and \( \hat{\Psi} \) are the piecewise constant derivative of this solution and finite volume representation of \( \hat{\Psi} \) respectively.

For single element-type meshes this approach was pioneered by Baliga and Patankar [1980] and extended by Baliga and Patankar [1983]; Cordes and Kinzelbach [1992]; Durlofsky [1993, 1994]; Huber and Helmig [2000]. More recently, it has also been applied to combinations of triangular and tetrahedral elements by Bogdanov et al. [2003]; Monteagudo and Firoozabadi [2004] and Reichenberger et al. [2006]. This work further extends it to arbitrary combinations of the element types: bar (line) element, triangle and quadrilateral (surface) elements, tetrahedron, hexahedron, pyramid and prism (volume) elements. This is accomplished by element partitioning.
into finite volumes in parametric space. As a result, there is no storage required for the finite volume mesh. The concrete implementation of this scheme is described below, including integral calculations and the mapping of variables from parametric to physical space. Multiphase flow simulations with the hybrid-element FE-FVM are presented by Matthai et al. [2005a].

2.3.4 Finite element-finite volume stencils

Each finite element (FE) contributes to as many finite volumes (FV) as it has nodes. We call the resulting FE partitions sectors and the set of equations ensuing from each element, FV stencil. Within each FE, sectors and therefore FVs are bounded by facets. Sectors are volumes in 3D, surfaces in 2D, and lines in 1D. Corresponding facets are surfaces, lines and points. Figure 2.9 illustrates these basic concepts for triangles and quadrilaterals, showing how adjacent FEs contribute to an FV surrounding a shared node. Importantly, facets connect at element barycentres and are subdivided in the middle of element faces (=the outer surfaces of the finite elements). This approach is referred to as barycentric tessellation and guarantees a robust decomposition of the FE mesh into non-degenerate FVs. It also permits the integration of material properties that vary from element to element. The alternative Voronoi tessellation [Voronoi, 1908; Cai et al., 1997] is simpler, requiring only half the number of facet-related computations (Figure 2.9 (b)). However, material properties varying from element to element cannot be integrated. An even bigger disadvantage is that Voronoi cells cannot be represented in parametric space. This rules out the Voronoi tessellation for our purposes.

New barycentric FV tessellations of the bar, triangle, quadrilateral, tetrahedron, hexahedron, prism and pyramid elements in parametric space are shown in Figures 2.9 and 2.10. Taking \( B(...) \) as a function which returns the coordinates of barycentre points of elements, \( e \), element faces, \( a \), and element edges, \( X_i X_j \), facet corner points \( F_{ij} \) are generated from element node coordinates as follows:

\[
F_{ij} \supset B(X_i, X_j), B(a_1), B(e), B(a_2)
\]  \hspace{1cm} (2.3)

Here, \( a_1 \) and \( a_2 \) are the two element faces bordering \( X_i X_j \), numbered following the right-hand rule using the direction of this edge as one of the axes. This guarantees that facet corner points are ordered counter-clockwise, the facet is outward facing relative to node \( i \), and the scalar projection of the facet normal onto the corresponding element edge is positive:

\[
(B(a_1) - B(X_i, X_j)) \times (B(e) - B(a_1)) \geq 0
\]  \hspace{1cm} (2.4)
Figure 2.9: Node-centred finite volumes discretising surface and volume elements. (a) Four neighbouring triangles and quadrilaterals share node $X$ around which finite volume is built using FE barycentres $b^{(e)}$ and midpoints of faces. FE are subdivided into sectors delimited by FV facets, $f$, with outward pointing normals, $n$, and the boundaries of finite elements. (b) Voronoi tessellation of the same geometry where FVs are bounded by orthogonal bisectors cutting adjacent finite element faces. (c) 3D finite volume composed of six pyramid finite elements. (d) Cross-section revealing node at FV centre.

Equation 2.4 uses the cross-product of two interconnected facet edges to find $n_{ij}$. To reduce the amount of floating point operations, point locations, $X^{(rst)}_i$ in parametric space can be memorised and interpolated to physical space on demand:

$$X^{(xyz)}_i = \sum_{j=1}^{n} N_j(X^{(rst)}_i)X^{(N,xyz)}_j$$

As the FE-FVM is based on isoparametric elements, i.e. their interpolation function order is the same as that of so called shape functions describing element shape [Taig, 1961], Equation 2.5 guarantees a projection of $X$ which is consistent with the spatial discretisation. This point is revisited in the discussion of discretisation errors.

In three-dimensional FEs, sectors are hexahedra and facets are planar quadrilateral-
Figure 2.10: Finite volume stencils for isoparametric linear tetrahedron, pyramid, prism and hexahedron. The apical sector of the pyramid on the right is an octahedron with warped basal facets making surface integrations inexact. To circumvent this problem the warped quadrilaterals are replaced by two planar triangular facets each.

...erals. The pyramid is an interesting exception: it has four non-planar facets which delimit an octahedral sector below its apex (Figure 2.10 (b)). As is alluded to in [Eaton et al., 2003] special provisions need to be made to perform integrations on this element. This is discussed in more detail in the section on parametric-to-physical transformations.

2.3.5 Volume and surface integration of finite volumes

We apply Gaussian quadrature [e.g. Burden et al., 1978], to obtain surface and volume integrals from the FVs. Integration is performed for specific integration points, $X_i^G$ with corresponding weights, $w_i$, chosen to give the best approximation of the unknown function. In contrast with linear FEM which integrates “weak” approximations of the field variable via the linear $N$ functions, FVM integrates constant approximations. These averages are products of the integrand and a single $M$ basis function which is unity inside the FV and zero elsewhere. In our node-centred FE-FVM, the presence of the $N$ functions permits to interpolate integrand values inside of FV sectors to locations where an approximation by a constant leads to an exact volume integral. The specific location in a sector where the value of the trilinearly interpolating $N$
functions is an exact average of the integrand, is its barycentre. More integration points are warranted only, if higher-order $N$ functions are used by the FEM. The volume integration weight for the sector barycentre, $w_1$, is set to the unit volume fraction of the FE which the sector occupies in parametric space.

Equation 2.1 shows that surface integration of a velocity field yields the divergence of flow inside an FV. This implies, for example, that the change in the concentration, $c$, of a solute in an FV can be computed as a flux balance (flux, $q = tcA|v|$, in kg m$^{-3}$) of incoming versus outgoing fluid with spatially variable concentration. The total flux, $q$, is accumulated by summing the fluxes across the facets of each sector of the FV:

$$q = t \int_{FV} \langle n, v, c \rangle \, dS = t \sum_{i=1}^{s} \sum_{j=1}^{f(i)} \tilde{c}_i A_i n_i \cdot v_{j}^{(e)}$$

(2.6)

Here, $t$ refers to the duration of the flow period and, $\tilde{c}$ is the concentration interpolated from the FE nodes (Equation 2.1). Again choices of number and position of integration points for the scalar quantity concentration and normals for projection of the vector quantity velocity are required. Consistency demands their location to be the same. For the nodal quantity, $c$, interpolated by $N$, the same arguments apply as stated for volume integration over FV sectors: Only one integration point located at the facet barycentre is warranted. Velocity, as it is computed from grad $N$, is constant inside each finite element and across individual facets. It varies only between facets located in different elements. For planar facets, therefore, the location of the normal does not matter and it can be placed at the facet barycentre. Facets delimiting the octahedral sector in the pyramid, however, are non-planar. Here, experiments show that while the accuracy of integrated facet area can be improved with multiple integration points, internal consistency of flux computations is obtained only for a single facet normal coinciding with an integration point at the facet barycentre. To achieve both, consistency and accuracy, we divide each non-planar quadrilateral facet into two planar triangular facets. This is discussed further in the context of the FV integration error.

In summary, we compute surface integrals of vector and scalar quantities and their products using a single facet normal and integration point located at the barycentre of the facet. The individual surface integration weights, $w_i$, assigned to $f_i$, are set equivalent to their area, $A_i$, in parametric space. Weights, integration point locations and facet normals have to be computed only once for each finite element type, see Tables 2.11 and 2.12.
### Table: Integration Points and Weights

<table>
<thead>
<tr>
<th>Nodes</th>
<th>Integration Point</th>
<th>Weight</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isoparametric Linear Bar (2 nodes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1, 2)</td>
<td>(0,0,0)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>Sector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1)</td>
<td>(-1/2,0,0)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2 (2)</td>
<td>(1/2,0,0)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Isoparametric Linear Triangle (3 nodes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1,2)</td>
<td>(5/12,1/6,0)</td>
<td>√3/6</td>
<td>(2/√3,1/√3,0)</td>
</tr>
<tr>
<td>2 (2,3)</td>
<td>(5/12,5/12,0)</td>
<td>√2/6</td>
<td>(-1/√2,1/√2,0)</td>
</tr>
<tr>
<td>3 (3,1)</td>
<td>(1/6,5/12,0)</td>
<td>√5/6</td>
<td>(-1/√5,-2/√5,0)</td>
</tr>
<tr>
<td>Sector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1)</td>
<td>(7/36,7/36,0)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>2 (2)</td>
<td>(11/18,7/36,0)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>3 (3)</td>
<td>(7/36,11/18,0)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>Isoparametric Linear Quadrilateral (4 nodes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1,2)</td>
<td>(0,-1/2,0)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>2 (2,3)</td>
<td>(1/2,0,0)</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>3 (3,4)</td>
<td>(0,1/2,0)</td>
<td>1</td>
<td>(-1,0,0)</td>
</tr>
<tr>
<td>4 (4,1)</td>
<td>(-1/2,0,0)</td>
<td>1</td>
<td>(0,-1,0)</td>
</tr>
<tr>
<td>Sector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1)</td>
<td>(-1/2,-1/2,0)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2 (2)</td>
<td>(1/2,-1/2,0)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3 (3)</td>
<td>(1/2,1/2,0)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4 (4)</td>
<td>(-1/2,1/2,0)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Isoparametric Linear Tetrahedron (4 nodes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1,2)</td>
<td>(13/36,5/36,5/36)</td>
<td>√6/24</td>
<td>(√6/3,√6/6,√6/6)</td>
</tr>
<tr>
<td>2 (2,3)</td>
<td>(13/36,13/36,5/36)</td>
<td>√2/24</td>
<td>(-√2/2,√2/2,0)</td>
</tr>
<tr>
<td>3 (3,1)</td>
<td>(5/36,13/36,5/36)</td>
<td>√2/24</td>
<td>(-√2/2,0,√2/2)</td>
</tr>
<tr>
<td>4 (1,4)</td>
<td>(5/36,5/36,13/36)</td>
<td>√2/24</td>
<td>(0,√2/2,√2/2)</td>
</tr>
<tr>
<td>Sector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1)</td>
<td>(23/144,23/144,23/144)</td>
<td>1/24</td>
<td></td>
</tr>
<tr>
<td>2 (2)</td>
<td>(25/48,23/144,23/144)</td>
<td>1/24</td>
<td></td>
</tr>
<tr>
<td>3 (3)</td>
<td>(23/144,25/48,23/144)</td>
<td>1/24</td>
<td></td>
</tr>
<tr>
<td>4 (4)</td>
<td>(23/144,23/144,23/144)</td>
<td>1/24</td>
<td></td>
</tr>
<tr>
<td>Isoparametric Linear Prism (8 nodes)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1,2)</td>
<td>(5/12,1/6,-1/2)</td>
<td>√6/6</td>
<td>(2/√3,1/√3,1/√3)</td>
</tr>
<tr>
<td>2 (2,3)</td>
<td>(5/12,5/12,-1/2)</td>
<td>√2/6</td>
<td>(-√2/2,√2/2,0)</td>
</tr>
<tr>
<td>3 (3,1)</td>
<td>(1/8,5/12,1/2)</td>
<td>√6/6</td>
<td>(-1/√3,-2/√3,0)</td>
</tr>
<tr>
<td>4 (1,4)</td>
<td>(7/36,7/36,1/2)</td>
<td>1/6</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>5 (2,5)</td>
<td>(11/18,7/36,1/2)</td>
<td>1/6</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>6 (3,6)</td>
<td>(7/36,11/18,1/2)</td>
<td>1/6</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>7 (4,5)</td>
<td>(7/12,1/6,1/2)</td>
<td>√6/6</td>
<td>(2/√3,1/√3,0)</td>
</tr>
<tr>
<td>8 (5,6)</td>
<td>(5/12,5/12,1/2)</td>
<td>√2/6</td>
<td>(-√2/2,√2/2,0)</td>
</tr>
<tr>
<td>9 (6,4)</td>
<td>(1/8,5/12,1/2)</td>
<td>√6/6</td>
<td>(-1/√3,-2/√3,0)</td>
</tr>
<tr>
<td>Sector</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (1)</td>
<td>(5/24,5/24,-1/2)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>2 (2)</td>
<td>(7/24,7/24,-1/2)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>3 (3)</td>
<td>(5/24,7/24,-1/2)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>4 (4)</td>
<td>(5/24,5/24,1/2)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>5 (5)</td>
<td>(7/24,7/24,1/2)</td>
<td>1/6</td>
<td></td>
</tr>
<tr>
<td>6 (6)</td>
<td>(5/24,7/12,1/2)</td>
<td>1/6</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.11: Integration points and weights. Integration points and weights in parametric space of isoparametric linear finite elements (Part I).
### Isoparametric Linear Pyramid (5 nodes)

<table>
<thead>
<tr>
<th>Facet</th>
<th>Nodes</th>
<th>Integration Point</th>
<th>Weight</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,2)</td>
<td>(0,-1/9, 5/36)</td>
<td>1/4</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>(2,3)</td>
<td>(4/9,0,5/36)</td>
<td>1/4</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>3</td>
<td>(3,4)</td>
<td>(0,4/9,5/36)</td>
<td>1/4</td>
<td>(-1,0,0)</td>
</tr>
<tr>
<td>4</td>
<td>(4,1)</td>
<td>(-4/9,0,5/36)</td>
<td>1/4</td>
<td>(0,-1,0)</td>
</tr>
</tbody>
</table>

### Non-planar upper facets

<table>
<thead>
<tr>
<th>Facet</th>
<th>Nodes</th>
<th>Integration Point</th>
<th>Weight</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>(1,5)</td>
<td>(-7/24,7/24,17/48)</td>
<td>√8/48</td>
<td>(3√74/74,√74,√74)</td>
</tr>
<tr>
<td>6</td>
<td>(2,5)</td>
<td>(7/24,7/24,17/48)</td>
<td>√8/48</td>
<td>(3√74/74,√74,√74)</td>
</tr>
<tr>
<td>7</td>
<td>(3,5)</td>
<td>(7/24,7/24,17/48)</td>
<td>√8/48</td>
<td>(-√74/74,√74,√74)</td>
</tr>
<tr>
<td>8</td>
<td>(4,5)</td>
<td>(-7/24,7/24,17/48)</td>
<td>√8/48</td>
<td>(3√74/74,√74,√74)</td>
</tr>
</tbody>
</table>

### Planar triangular upper facets

<table>
<thead>
<tr>
<th>Facet</th>
<th>Nodes</th>
<th>Integration Point</th>
<th>Weight</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>(1,5)</td>
<td>(-7/24,-1/6,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
<tr>
<td>6</td>
<td>(1,5)</td>
<td>(-1/6,-7/18,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
<tr>
<td>7</td>
<td>(2,5)</td>
<td>(1/6,-7/18,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
<tr>
<td>8</td>
<td>(2,5)</td>
<td>(7/18,1/6,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
<tr>
<td>9</td>
<td>(3,5)</td>
<td>(7/18,1/6,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
<tr>
<td>10</td>
<td>(3,5)</td>
<td>(1/6,7/18,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
<tr>
<td>11</td>
<td>(4,5)</td>
<td>(-1/6,7/18,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
<tr>
<td>12</td>
<td>(4,5)</td>
<td>(-7/18,1/6,13/36)</td>
<td>√8/48</td>
<td>(3√74/74,-√74,-√74)</td>
</tr>
</tbody>
</table>

### Sector

<table>
<thead>
<tr>
<th>Sector</th>
<th>Nodes</th>
<th>Integration Point</th>
<th>Weight</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1)</td>
<td>(-31/72,-31/72,23/144)</td>
<td>1/4</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>(2)</td>
<td>(31/72,-31/72,23/144)</td>
<td>1/4</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>3</td>
<td>(3)</td>
<td>(31/72,31/72,23/144)</td>
<td>1/4</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>4</td>
<td>(4)</td>
<td>(-31/72,31/72,23/144)</td>
<td>1/4</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>5</td>
<td>(5)</td>
<td>(0,0,25/48)</td>
<td>1/3</td>
<td>(1,0,0)</td>
</tr>
</tbody>
</table>

### Isoparametric Linear Hexahedron (8 nodes)

<table>
<thead>
<tr>
<th>Facet</th>
<th>Nodes</th>
<th>Integration Point</th>
<th>Weight</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,2)</td>
<td>(0,-0.5,-0.5)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>(2,3)</td>
<td>(0.5,0,-0.5)</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>3</td>
<td>(3,4)</td>
<td>(0,0.5,-0.5)</td>
<td>1</td>
<td>(-1,0,0)</td>
</tr>
<tr>
<td>4</td>
<td>(4,1)</td>
<td>(-0.5,0,-0.5)</td>
<td>1</td>
<td>(0,-1,0)</td>
</tr>
<tr>
<td>5</td>
<td>(1,5)</td>
<td>(-0.5,0,-0.5)</td>
<td>1</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>6</td>
<td>(2,6)</td>
<td>(0.5,0,-0.5)</td>
<td>1</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>7</td>
<td>(3,7)</td>
<td>(0,0.5,0)</td>
<td>1</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>8</td>
<td>(4,8)</td>
<td>(-0.5,0.5)</td>
<td>1</td>
<td>(0,0,1)</td>
</tr>
<tr>
<td>9</td>
<td>(5,6)</td>
<td>(0,-0.5,0.5)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>10</td>
<td>(6,7)</td>
<td>(0.5,0,0.5)</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>11</td>
<td>(7,8)</td>
<td>(0,0.5,0.5)</td>
<td>1</td>
<td>(-1,0,0)</td>
</tr>
<tr>
<td>12</td>
<td>(8,5)</td>
<td>(-0.5,0,0.5)</td>
<td>1</td>
<td>(0,-1,0)</td>
</tr>
</tbody>
</table>

### Sector

<table>
<thead>
<tr>
<th>Sector</th>
<th>Nodes</th>
<th>Integration Point</th>
<th>Weight</th>
<th>Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1)</td>
<td>(-0.5,-0.5,-0.5)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>2</td>
<td>(2)</td>
<td>(0.5,-0.5,-0.5)</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>3</td>
<td>(3)</td>
<td>(0.5,0.5,-0.5)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>4</td>
<td>(4)</td>
<td>(-0.5,0,0.5)</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>5</td>
<td>(5)</td>
<td>(-0.5,-0.5,0.5)</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>6</td>
<td>(6)</td>
<td>(0.5,-0.5,0.5)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
<tr>
<td>7</td>
<td>(7)</td>
<td>(0.5,0.5,0.5)</td>
<td>1</td>
<td>(0,1,0)</td>
</tr>
<tr>
<td>8</td>
<td>(8)</td>
<td>(-0.5,0.5,0.5)</td>
<td>1</td>
<td>(1,0,0)</td>
</tr>
</tbody>
</table>

Figure 2.12: Integration points and weights. Integration points and weights in parametric space of isoparametric linear finite elements (Part II).
Importantly, all computations described thus far are performed in parametric space. The next section explains how the obtained integrals are scaled to their actual values in physical space.

2.3.6 Vector and integral transformations

Sector volume is mapped from parametric to physical space by a Jacobian transformation [Barr, 1984]. The standard square Jacobian matrix, $J$, for a specific FE type and integration point therein is obtained by pre-multiplying the element’s node coordinate vector with the shape function derivative matrix evaluated in parametric space at the integration point $X_i^{G(rst)}$ of interest. Thus, for the bar element (Figure 2.8) in a one-dimensional space

$$J|_{X_i^{P}} = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} \end{bmatrix} \begin{bmatrix} X_1^{(x)} \\ X_2^{(x)} \end{bmatrix}$$

(2.7)

where $X_i$ refers to its node coordinates.

This procedure is extensively documented in the literature [e.g. Zienkiewicz and Taylor, 2000]. However, FE interpolation functions and their derivatives are rarely spelt out in full. To aid the reproduction of our hybrid element FE-FVM, we provide a complete list of shape functions, their derivatives, and corresponding node, face, and edge numbering conventions in Appendix A.

2.3.7 Facet area

Facets of surface elements are lines. To map their length from parametric to global space we use the line’s Jacobian vector $J_n$. Its determinant multiplied by the weight, $w_1$, yields the length of the facet in physical space. In this case, a single integration point and a unit length of the facet line in parametric space are assumed:

$$J_n|_{X_i^{P}} = \begin{bmatrix} X_1^{(x)} & X_1^{(y)} \\ X_2^{(x)} & X_2^{(y)} \end{bmatrix} \begin{bmatrix} \frac{\partial N_1}{\partial r} = -1/2 \\ \frac{\partial N_2}{\partial r} = 1/2 \end{bmatrix}$$

(2.8)

$$\Delta X^{(xyz)} = w_1 |J_m|$$

(2.9)

$$|J_m| = \sqrt{|J_m J_m^T|}$$

(2.10)

$$w_1 = 1$$

(2.11)
In Equation 2.8 $X_i$ refers to the coordinates of the endpoints of the facet line interpolated from parametric space to physical space (Equation 2.5).

For this type of transformations from a lower to a higher dimensional space, $J$ is not square and calculating its determinant involves transformation techniques given by [Juanes et al., 2002]. The essence of this procedure is to add (a) row(s) to $J$ to make it square and therefore invertible. The dimensions of the Jacobian of a surface element embedded in three-dimensional space are $n \times m$ (number of dimensions in physical space by number of dimensions in parametric space). This matrix is made square by multiplication with its transpose. Its determinant is the square root of the covariant metric tensor $\sqrt{G}$, where $G = J^T m \neq n J m \neq n$, see Equation 2.10.

The facets of volume elements are planar quadrilaterals or triangles. The determinants of their non-square Jacobians relate area in parametric space to that in physical space. A shape and integration point location as well as number-dependent weighting factor, $w_i$, is required so that the mapping yields the correct facet area in physical space (Table 1). For quadrilateral facets

$$J_{m=n}|X_i^G = \begin{bmatrix} \frac{\partial N_1}{\partial r} |X_i^G & \cdots & \frac{\partial N_4}{\partial r} |X_i^G \\ \frac{\partial N_1}{\partial s} |X_i^G & \cdots & \frac{\partial N_4}{\partial s} |X_i^G \end{bmatrix} \begin{bmatrix} X_1^{(x)} & X_1^{(y)} & X_1^{(z)} \\ X_2^{(x)} & X_2^{(y)} & X_2^{(z)} \\ X_3^{(x)} & X_3^{(y)} & X_3^{(z)} \\ X_4^{(x)} & X_4^{(y)} & X_4^{(z)} \end{bmatrix}$$ (2.12)$$

and for the triangular facets in the apex of the pyramid

$$J_{m=n}|X_i^G = \begin{bmatrix} \frac{\partial N_1}{\partial r} |X_i^G & \cdots & \frac{\partial N_3}{\partial r} |X_i^G \\ \frac{\partial N_1}{\partial s} |X_i^G & \cdots & \frac{\partial N_3}{\partial s} |X_i^G \end{bmatrix} \begin{bmatrix} X_1^{(x)} & X_1^{(y)} & X_1^{(z)} \\ X_2^{(x)} & X_2^{(y)} & X_2^{(z)} \\ X_3^{(x)} & X_3^{(y)} & X_3^{(z)} \end{bmatrix}$$ (2.13)$$

The coordinates $X_i$ mark corner points of the facet mapped from parametric to physical space. For internal consistency, mapped facet area must be used together with mapped normal orientation unless velocity projections are also carried out in parametric space.

### 2.3.8 Normal orientation

Facet normals are mapped from parametric space to physical space by using the standard Jacobian inverse relationship described by Barr [1984], where

$$n^{xyz} = (J_{m=n}^{-1})^T n^{rst}$$ (2.14)
and $\mathbf{J}$ is the Jacobian of the parent element formed at the facet integration point of interest. Once computed, the Euclidean length of the normal is scaled to 1. Facets of line elements have a normal parallel to the element pointing from node $n−1$ to node $n$.

2.3.9 Integral computations in physical space

The mapping of facet normals from parametric to physical space is more computationally expensive than their calculation in physical space. For our optimised C++ template-based implementation the extra number of clock cycles required varies with parent element type between 5% for the tetrahedron and 90% for the quadrilateral element in three dimensions. However, this disadvantage is outweighed by the internal consistency delivered by the mapped approach: quadrilateral facets in physical space are rarely planar so that a distortion-dependent error is incurred in the area computation and it is difficult to find a representative facet normal. A remedy would be to replace all quadrilateral facets by triangular facets because these are planar by default. This is how we treat the apical facets in the pyramid element because these are distorted already in parametric space. As expected, triangular facets are faster to compute and map than quadrilateral facets (approximately 8%). However, twice as many facets, normals and projections would be required if such a discretisation was used for all facets. Computational cost could only partially be offset by storing mapped normals and facet areas for repeated use.

For linear line elements, the facet normal is simply an element-parallel unit vector pointing from node to node. For surface elements, facet normals lie in the plane of the element and are calculated by

$$
\mathbf{n}_{ij} = \left( \frac{X_i + X_j}{2} - B \right) \times \mathbf{n}^{(e)}
$$

(2.15)

where $\mathbf{n}^{(e)}$ is the normal to the parent surface element in physical space, $i$ and $j$ are its node points on either side of the facet, and $B$ is its barycentre.

For volume elements, each facet is a quadrilateral or triangular isoparametric FE with a surface orientation defined by the two counter-clockwise edge vectors originating in $B$. Their cross product is the facet outward pointing normal. A single Jacobian transformation is needed to find facet area in physical space.
quadrilateral facets

\[
J_{m=n|B} = \begin{bmatrix}
-1/4 & 1/4 & 1/4 & -1/4 \\
-1/4 & -1/4 & 1/4 & 1/4
\end{bmatrix}
\begin{bmatrix}
X_1^{(x)} & X_1^{(y)} & X_1^{(z)} \\
X_2^{(x)} & X_2^{(y)} & X_2^{(z)} \\
X_3^{(x)} & X_3^{(y)} & X_3^{(z)} \\
X_4^{(x)} & X_4^{(y)} & X_4^{(z)}
\end{bmatrix}
\]

(2.16)

\[w_1 = 4\]  

(2.17)

For triangle facets:

\[
J_m = \begin{bmatrix}
-1 & 1 & 0 \\
-1 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_1 & y_1 & z_1 \\
x_2 & y_2 & z_2 \\
x_3 & y_3 & z_3
\end{bmatrix}
\]

(2.18)

\[w_1 = 1/2\]  

(2.19)

The facet corner point coordinates, \(X_i\), in equations (2.3-2.5) are calculated directly in physical space. To obtain unit normals, all facet normals are divided by their length.

We favour the parametric space approach combined with Jacobian mapping over the direct computation of sector volume, facet area and facet normals in physical space. Noteworthy is that for the transport modelling described later, computation speed can be improved if the transport velocity is also computed in parametric space, eliminating, among other steps, the mapping of the facet normal. The presented numerical method is suitable also for quadratic or cubic finite elements. The main idea of finite volumes surrounding finite element nodes remains the same. Notice, however, that the improved continuity for such elements permits to treat them as finite volumes. This constitutes another discretisation approach, an element-centred FE-FVM.

### 2.3.10 Step 4: Tracer advection in fractured porous rock

In this section we discretise the governing equations for fluid flow and tracer advection with the FE-FVM for a demonstration on a model of the intensely fractured San Andreas formation in Section 2.4. We will assume that transport occurs in a time invariant velocity field without feedback between tracer concentration and velocity. Huyakorn and Pinder [1983] solved this problem by coupling the solution of two partial differential equations for pressure and tracer concentration, respectively. As
the pressure equation is elliptic and the transport equation is hyperbolic, we apply a simple operator splitting technique solving for pressure with the FEM and for concentration with the FVM. Note, however, that our FE-FVM is completely general, forming the basis of a higher order accurate, semi-implicit transport method for two-phase flow [Matthai et al., 2005b], and calculations of thermohaline convection including the effects of phase separation of highly compressible steam [Geiger et al., 2006a,b]. The discussion presented here serves the sole purpose of illustration of how such schemes can be constructed with the data structures that we provide.

**Governing pressure equation**

Assuming that the divergence of the velocity field is zero, i.e. that there are no fluid sources or sinks in the model domain $\Omega$,

$$\nabla \cdot \mathbf{u} = 0 \quad \text{(conservation of volume)} \quad (2.20)$$

If the volumetric flow, $\mathbf{u}$, obeys Darcy’s law,

$$\mathbf{u} = \frac{A \mathbf{k}}{\mu} \nabla p$$ 

it follows that,

$$\frac{A \mathbf{k}}{\mu} \nabla^2 p = 0 \quad (2.22)$$

where $p$ is pressure, $A$ is the area of the flow cross-section, $\mathbf{k}$ is the piecewise-constant permeability tensor, and $\mu$ is a piecewise-constant dynamic fluid viscosity. FE integration of PDE (2.22) yields the element contributions to the domain integral $\int \Omega$ for the model

$$\int dN^T A \frac{k}{\mu} dNdV = 0 \times \int N^T N dV \quad (2.23)$$

where $dN$ represents the interpolation function matrix of spatial derivatives (see Appendix A). This integration requires a value of the integration constant (fluid pressure) fixed at, at least, one node in the model.

**Governing advection–diffusion equation**

Advection and diffusion of a non-reactive tracer through a fractured porous medium with discrete fracture representations can be described by

$$\phi \frac{\partial c}{\partial t} + D \nabla^2 c + \nabla \cdot \mathbf{v} c - q = 0 \quad (2.24)$$
The variables in this continuity equation are the porosity, \( \phi \), tracer concentration, \( c \), macro dispersivity of the tracer, \( D \), and \( v \) the Darcy velocity; \( q \) refers to sources or sinks of the tracer. In many practical cases it is not possible to parameterise \( D \) due to small-scale geometric uncertainty but we retain \( D \) as a diagonal tensor to illustrate how diffusion and advection can be solved for simultaneously using the FE-FVM.

Equation 2.16 is discretised spatially using the FV framework and temporally using a fully implicit (Backward-Euler) finite difference scheme. For each FV, this yields the volume and surface integrals listed below. These are accumulated into a sparse solution matrix, \( A \), of linear algebraic equations with coefficients \( j \) and a right-hand vector, \( b \), indexed with \( i \):

\[
\int_{V_f} M_i \frac{\partial c_i}{\partial t} dV + \int_{\Gamma_{out}} \mathbf{n} \cdot \mathbf{u}_c + D_n \frac{\partial c_c}{\partial n} c_c dS + \int_{\Gamma_{in}} \mathbf{n} \cdot \mathbf{u}_c + D_n \frac{\partial c_u}{\partial n} c_u dS - \int_{V_f} M_i q_i dV = 0
\]

(2.25)

\[
A_{(i,i)} = \frac{V_f}{\Delta t} + \sum_{j=0}^{n_f(FV)} A_j \cdot n_j \cdot u_j + k \frac{\partial c}{\partial n_j} H \left( n_j \cdot u_j + D_n \frac{\partial c}{\partial n_j} \right)
\]

(2.26)

\[
A_{(i,j)} = \sum_{j=0}^{n_f(FV)} A_j \cdot n_j \cdot u_j + k \frac{\partial c}{\partial n_j} \left( 1 - H \left( n_j \cdot u_j + D_n \frac{\partial c}{\partial n_j} \right) \right)
\]

(2.27)

\[
b_{(i)} = \frac{c^t V_f}{\Delta t} + q^{t+1}_i
\]

(2.28)

\[
H = \begin{cases} 
0 & \text{if } \mathbf{n}_f \cdot \mathbf{u}_f + D_n \frac{\partial c}{\partial n_j} < 0 \\
1 & \text{otherwise}
\end{cases}
\]

(2.29)

where \( \Gamma_{out} = FV(\mathbf{n}_f \cdot \mathbf{u}_c < 0) \) and \( \Gamma_{in} = FV(\mathbf{n}_f \cdot \mathbf{u}_c > 0) \). Concentration, \( c \), is discretised on the FVs by the piecewise constant interpolation functions \( M_i \). The subscripts, \( u \) and \( c \) refer to the upstream and current finite volumes relative to a facet, \( f \), across which the contribution to the total flux is calculated. Each FV facet has the unit normal \( n \) and area \( A \) (Figure 2.8). The Heaviside function \( H() \), is used to discard couplings between FVs which would arise from fluxes in the downstream direction. As \( H \) evaluates to zero in this case, only incoming fluxes create couplings between adjacent FVs, i.e. diagonal and off-diagonal terms in the solution matrix. This technique is known as upstream weighting and is unconditionally stable for first-order schemes [Baliga and Patankar, 1980].
Diffusion of the tracer is modelled simultaneously with advection projecting concentration gradients onto the facet normals, see Equation 2.6.

At the domain boundaries

\[ A_{ii} = A_{ii} + \sum_{j=1}^{n_f} A_j \cdot n_j \cdot u_j \quad \text{if} \quad \sum_{j=1}^{n_f} A_j \cdot n_j \cdot u_j < 0 \]  
\[ b_i = b_i + \bar{c}_i \sum_{j=1}^{n_f} A_j \cdot n_j \cdot u_j \quad \text{if} \quad \sum_{j=1}^{n_f} A_j \cdot n_j \cdot u_j > 0 \]  

Thus, a left-hand-side implicit (meaning that \( c \) does not need to be specified) compensation is made for finite volumes split by outflow boundaries and an explicit one for inflow boundaries. The resulting non-symmetric but diagonally dominant and positive definite matrix of linear algebraic equations, \( Ax = b \), as well as the symmetric matrix arising from the pressure equation, is solved with the algebraic multigrid method for systems, SAMG [Stüben, 2001]. The results are stored at the finite element nodes. The new hybrid element FE-FVM is implemented in the Complex System Modelling Platform (CSMP++), an object-oriented application programmer interface (API), designed for the simulation of complex geological processes and their interactions (formerly CSP, [Matthai et al., 2001]).

2.4 Results

We evaluate the new hybrid FE-FVM framework in four ways: (1) we measure the impact of increasing geometric complexity on the quality of the generated hybrid element meshes, (2) we measure the FE discretisation error for the pressure equation, (3) we evaluate the FV integration error by integrating prescribed and computed non-divergent velocity fields over the model, and (4) we estimate speed and memory requirements of the implicit first-order transport scheme applied to the intensely fractured San Andreas model.

2.4.1 Mesh refinement and increasing geometric complexity

Three fracture models built using FracMan Reservoir Edition (FRED) (Golder Associates, Seattle, WA, USA), are used to illustrate the effects of increasing geometric complexity on mesh quality (Figure 2.13). These models are populated with randomly oriented disc-shaped fractures with a power law length distribution. The power law exponent is 1.5 and fracture diameter ranges from 5 to 50 m (box size
The ratio between the fracture surface area and the volume of the model, also known as p32 [cf. Dershowitz, 1984], is used as a proxy for geometric complexity. In contrast with the polygonal representation in FracMan, however, we represent each fracture by a circular NURBS surface.

Each model was meshed at three levels of refinement decreasing target element size from 10m to 5m to 2m. The resulting tetrahedral meshes were converted into hybrid elements by transforming groups of 12 tetrahedra into single hexahedra. This procedure succeeds in regions of the CAD model that are far from confining surfaces. Conversion rate depends on feature density relative to target element size.

In all cases, the mesh was adaptively refined down to one-tenth of the reference size to match the curved tip lines of the fractures. The least complex model, model A, has a p32 of 0.05, and contains 66 fractures. Model B has 108 fractures and a p32 of 0.1, and model C, the most complex model, contains 334 fractures with a p32 of 0.3.

In total, 18 meshes (3 models × 3 refinement levels × 2 mesh types) were generated to evaluate mesh characteristics such as the impact of refinement on the number of nodes, number of elements, element to node ratio, and hexahedral to tetrahedral element ratio (Figure 2.14). This ratio is of interest because it influences the sparseness of the solution matrix as well as the complexity of the FV cells. Our detailed analysis shows that:
• As refinement increases more and more tetrahedra are converted to hexahedra. Feature (=fracture) poor regions are discretised exclusively with hexahedra while fracture-rich regions remain tetrahedra. Tetrahedra dominate near inclined fractures. With increasing refinement, tip line curvature becomes less influential on the conversion rate and larger volume fractions are converted to hexahedra.

• In model A, tetrahedral to hexahedral element conversion reduces the number of nodes by 3.48-37%. The number of volume elements is reduced by 9.59-76%, respectively. The element to node ratio decreases from an average of 5.82 for pure tetrahedral meshes (consistent with observations of Bogdanov et al. [2003]), to an average of 2.23 elements in the most refined hybrid meshes.

• In model B conversion decreases the number of nodes by 0.95-29%. The number of volume elements is reduced by 2.61-60%. Element to node ratio changes from an average of 5.68 in the tetrahedral meshes to 4.63 in the hybrid ones.

• In the intensely fractured model C, differences between tetrahedral and hybrid meshes are not as pronounced as in A and B. The hybrid mesh only has up to 10% less nodes and elements, and the element to node ratio stays around 5.46. Thus, as model complexity increases, less and less elements are converted into hexahedra. However, this should not be interpreted as a weakness of the method because conversion percentage relates directly to element size in relation to feature density. As feature density increases, the domain is more intensively subdivided and less tetrahedral elements are converted into hexahedra.

As accurate resolution of the curved pressure field between fractures requires multiple elements (see Step 5 of workflow), a poor conversion ratio almost always indicates that the mesh is too coarse to resolve the input geometry.

The fractures in models A, B and C are represented by surfaces. For a volumetric discretisation by high aspect ratio prism elements, the number of additional elements required scales with the number of element layers. The representation of a fracture by multiple layers of prism elements with a large aspect ratio of 150 requires several orders of magnitude less elements than a volumetric representation by tetrahedra. For models A-C and a target element size of 2 m, a conversion of fractures from triangulated surfaces to volumes with four prism layers increases the total element number only by 39, 43 and 53%, respectively.
Hybrid element meshes share the advantages of unstructured and structured grids: they are flexible enough to capture complex geometry but use the minimum number of elements required to discretise regions without geometric constraints. Fast transitions between highly refined and coarse regions are possible and the better performing hexahedra are used wherever the geometry allows this.

2.4.2 Step 5: A posteriori mesh adaptation guided by finite element discretisation error

During the discretisation phase the mesh is optimised to accurately represent the complex geometry. This match can be quantified, for example, in terms of the largest deviation of an element face from a material interface represented by an NURBS curve. The finite element discretisation error has a similar role as it quantifies the ability of the mesh to represent a specific solution for a specific order of the element interpolation functions. For polynomials of $n$th degree the discretisation error is of the order $O(h^{n+1})$ [e.g. Zienkiewicz and Taylor, 2000], where $h$ is the element size. For linear elements, used in this work, the error is of order $O(h^2)$. They can capture
linear variations in the solution variable exactly, but any nonlinear variations produce a solution curvature dependent error which scales with element size. As the mesh is refined, this error decreases proportionally. While it cannot be eliminated completely, adaptive refinement can be used to distribute the discretisation error uniformly over the mesh. To guide such refinement a measure for the discretisation error is needed. By contrast with first-order accurate linear elements, quadratic elements are second-order accurate ($O(h^2)$) and the normalised difference between quadratic and linear FEM solutions

$$\varepsilon^2(\Psi) = \frac{||\hat{\Psi}^2|| - ||\hat{\Psi}^1||}{||\hat{\Psi}^2||}$$

therefore quantifies the $O(h^2)$ error term. In Equation 2.32 $\hat{\Psi}^1$ and $\hat{\Psi}^2$ are the linear and quadratic FEM solutions, respectively.

To illustrate $\varepsilon^2$ we have computed it for a uniformly refined idealised model of three intersecting disc-shaped fractures (Figure 2.15). This model also allows an assessment of the accuracy of a discretisation error estimate $\tilde{\varepsilon}^2$ made in the absence of a second-order accurate solution which is usually too expensive to compute.

Cook et al. [240] show that a linear FEM solution contains enough information to estimate the discretisation error. Here, we compute an element centred error metric using a Hessian matrix-based a posteriori error estimate that takes into account the size and shape of the finite elements as well as the level of alignment of the mesh with the solution [Pain et al., 2001],

$$\tilde{\varepsilon}^2(\Psi) = \gamma h^T |H| h, \quad H = \nabla N^T \hat{\Psi} \nabla N$$

where $\gamma$ is a scale factor needed to arrive at a target finite element size and $H$ is the characteristic Hessian matrix obtained by an eigen decomposition of the squared interpolation function derivative matrix multiplied with the first-order FEM solution. Each row of $|H|$ contains an eigenvector $\lambda$ of $H$, scaled in length by the corresponding eigenvalue. The vector $h$ contains the length of the analyzed element in the direction of this eigenvector.

Figure 2.16 illustrates how $h$ is computed by measuring element length in the directions of $\lambda_1 - \lambda_3$ in terms of the height of an oriented bounding box around the element. This computation is performed as follows:

$$h_i = \max_{j=1}^{n_n} \left\{ \frac{X_j^{(xyz)} \cdot \lambda_i}{|\lambda_i|} \right\} - \min_{j=1}^{n_n} \left\{ \frac{X_j^{(xyz)} \cdot \lambda_i}{|\lambda_i|} \right\}$$

(2.34)
Figure 2.15: Estimation of spatial discretisation error for a posteriori mesh refinement. (a) Normalised difference between predicted maximum of $O(h^2)$ error $\varepsilon$, and its exact value. Error is plotted as a function of node number for model shown in (b). (b) Model of three intersecting highly permeable discs. Isosurfaces of fluid pressure and anisotropic Hessian-based discretisation error are visualised by tensor glyphs rendered with the visualisation toolkit (VTK). The spatial discretisation error is largest where the curvature in the fluid pressure gradient is maximised.

where $X_j$ is the node coordinate vector for node $j$ of the element in physical space. Figure 2.15A plots the actual and estimated discretisation errors that have a very similar distribution. Their magnitude also matches reasonably well (within a factor of 3). Five peaks are localised in the rock matrix next to up- and downstream fracture terminations (cf. Figure 2.1B). In these locations fluid pressure gradients and their spatial variation are maximised and the error indicator suggests that they should be highly refined ($\geq 5\times$). Within the fractures the error is small because their high permeability minimises the pressure gradient. Thus, large aspect ratio prism or hexahedral elements are a suitable discretisation for these.

Alongside with Pain et al. [2001] and Lipnikov and Vassilevski [2004], we conclude that a Hessian-based error metric provides a good estimate of the discretisation error and can be used to control adaptive mesh refinement. Within the hybrid FE-FVM framework, the computation of $\varepsilon^2$ for the pressure equation is possible because the $N$ functions are differentiable. By contrast, pressure computations based on a finite-volume approach only, as are common in reservoir simulators, preclude this. While the FVM is mass conservative by default, flow velocities computed on a coarse model with large aspect ratio heterogeneities will have little in common with ones obtained from an appropriately refined FE-FVM discretisation.
2.4.3 Finite-volume integration error

To validate the internal consistency of the stencils, mappings and finite volume surface integrations with a hybrid element model we compare the integrated volumetric inflow and outflow from each FV for a prescribed constant velocity with a magnitude of 1 and a computed divergence free velocity field of spatially variable magnitude. For the constant velocity, the measured differences, $\Delta u$, should be numerically zero ($\text{numeric\_limits<double>::epsilon()}$, see C++ standard). For the spatially variable velocity, $\Delta u$ will depend on the divergence of the computed field, i.e. numerical accuracy of the linear FE solution, and on the field’s curl in relation to FE size. In any case, the volume integration error scales with FV flow cross-sectional area, $A_T$:

$$A_T = \frac{1}{2} \sum_{j=1}^{J} A_j \left| n_j \cdot \frac{v}{|v|} \right|$$

(2.35)

To account for this size effect $\Delta u$ is scaled by $A_T$, as follows:

$$\varepsilon_i^{(FV)} = \left| \frac{\Delta u_i}{A_{T,i}} \right|$$

(2.36)

where $\varepsilon_i^{(FV)}$ is the volume integration error for finite volume $i$, reflecting the accuracy of facet area calculations and projections of velocities. For the $10 \times 10 \times 7$ m hybrid

38
element model (Figure 2.15B) with 19% tetrahedra, 18% pyramids, and 63% prisms and finite element volumes ranging from $1.88 \times 10^{-5}$ to 0.0708 m$^3$, we obtain the following results:

1. Prescribed unit velocity: $\varepsilon^{(FV)}_{\text{max}}$ is less than or equal to $9.379 \times 10^{-14}$ m/s. For the entire model the difference between inflow and outflow is $6.82 \times 10^{-13}$ m$^3$/s at a total cross-sectional flow of 70 m$^3$/s.

2. For the computed velocity field with magnitudes ranging from $1.695 \times 10^{-5}$ to $6.454 \times 10^{-3}$ m/s, $\varepsilon^{(FV)}_{\text{max}}$ is less than or equal to $1.16 \times 10^{-14}$ m$^3$/s. The total cross-sectional flow through this model is 0.0038 m$^3$/s.

As the maximum velocity in the computed velocity field is orders of magnitude smaller than the prescribed velocity (case 1), the second set of results implies a much larger $\varepsilon^{(FV)}_{\text{max}}$ for these actual application conditions. However, the maximum $\varepsilon^{(FV)}_{\text{max}}$ value still is so small that the corresponding violation of volume conservation does not lead to a drift in the values of the advected variable. In the development of the method and experimentation with a pyramid stencil with all quadrilateral facets, three orders of magnitude larger $\varepsilon^{(FV)}_{\text{max}}$ occurred wherever pyramid stencils contributed to finite volumes. This result prompted us to replace the warped quadrilateral facets in the apex of the pyramid by planar triangular ones.

### 2.4.4 Application of implicit first-order transport scheme to a discrete 3D fracture model

Figure 2.17 is a plan view of the 1km $\times$ 1km $\times$ 0.2km-sized model FRACS2000 of a fractured sandstone reservoir in the San Andreas formation, southern California, USA [Matthai et al., 2005b]. Two sets of sub-vertical fractures partially intersect at angles between 70 and 80 degrees. The total fracture-rock matrix interface area is 12.3%km$^2$. Sandstone porosity (0.25) and permeability (10 mD) are treated as uniform, and fracture aperture as a linear function of fracture diameter leading to values ranging from 0.5 to 3.5 mm. The parallel plate model [Kranz et al., 1979; Witherspoon et al., 1980] was used to compute fracture permeability for these aperture values. We treat the macro-dispersivity, $D$ as an isotropic diagonal tensor valued $10^{-30}$. This means that any diffusion which arises in the computation is of numerical nature.

Model FRACS2000 is discretised with 1 113 580 hybrid elements (60% tetrahedra, 15% hexahedra, 13% triangles, 11% pyramids, 1% quadrilaterals and 0.01% prisms)
Figure 2.17: Planview of three-dimensional flow and transport model FRACS2000 [Matthai et al., 2005a]. Translucent fractures belong to two approximately orthogonal sets, 1000 fractures per set, each with a power-law diameter-frequency distribution. Flow is from the bottom to the top. After 3.5 months of injection through the lower boundary the tracer front visualised by a dark grey isosurface through a tracer concentration of 1.5 has advanced through two-third of the model. Contour lines on a horizontal cut plane mark the tracer dispersion across the front.

and 223 705 nodes (=degrees of freedom) using a subdimensional, surface mesh representation of the fractures.

Figure 2.17 displays a snapshot of the concentration of a non-reactive tracer injected continuously for 4 months through the lower boundary into a time-invariant flow field arising from a hydrostatic far-field fluid pressure gradient aligned with the model edge. This gradient was created by applying uniform pressures of approximately 20 and 0 MPa at the two opposite model boundaries, respectively.

The pressure equation 2.22 was solved with the FEM and the transport equation 2.24 with the first-order implicit FVM (Equation 2.29). The run on a 2-GHz M-processor notebook took 6.5 min. Most of this time was used to write output files in text format. This result reflects a logarithmic scaling of solution time with number of unknowns as the decisive characteristic of SAMG which can be applied in spite of the non-symmetric shape of the sparse solution matrix.

The snapshot (Figure 2.17) reveals a highly embayed fracture-controlled tracer
front partially penetrating the sandstone. In the fractures, the tracer has advanced over two thirds of the model’s length equating to an average speed of 0.2 m/h. Examination of the finite volumes in the different CAD domains reveals that only 14% of the fractures have come in contact with the tracer thus far. Averaging tracer concentration in front-parallel slabs along the direction of flow yields a long leading edge typical for “anomalous” non-Fickean dispersion seen in actual fractured rock [e.g. Berkowitz and Scher, 1995].

The steep fracture-associated concentration gradients perpendicular to the tracer front show that even for the first-order transport scheme used, numerical diffusion is subordinate to mechanical dispersion caused by the fractures. While no physical importance should be attributed to this specific set of results as they lack any meaningful parameterisation of $D$, the example serves to illustrate the potential capabilities of FE-FVM combined with SAMG.

2.5 Discussion

The characteristics of the geometrically complex simulation with model FRACS2000 highlight the decisive advantages of our hybrid FE-FVM scheme:

- Indirect unstructured hybrid-element meshing facilitates efficient and semi-automatic discretisation of complex geological models, even permitting a realistic representation of the power law length-frequency relationships which typify some natural fractured rocks. For comparison, a structured regular grid model of FRACS2000 with the same amount of detail would have required $4^{11}$ cells ($(1000 \times 1000 \times 200)/0.0005 = \text{aperture of smallest fracture}$), i.e. 5 orders of magnitude more cells than the unstructured model.

- In spite of the cheap linear FEM derived piecewise-constant velocity field, our node-centred FVM conserves the advected tracer during migration across strong discontinuities in permeability and porosity.

- Although it is advocated primarily for elliptic problems, the algebraic multigrid solver SAMG performs fast and reliably on the non-symmetric hyperbolic sets of linear algebraic equations arising from the upstream-weighted advection-dispersion PDE.

- The effort to generate the internally consistent BREP with topology, i.e. the CAD volume decomposition of the geological system also pays off in the analysis
of the results because the model subvolumes distinguished by the CAD can be analyzed individually so that their specific behaviour and interplay can be determined.

2.6 Conclusions

We introduce a new workflow for the simulation of structurally complex geologic models starting with geological interpretation and NURBS BREP of structures followed by indirect unstructured hybrid finite element meshing aimed at producing an unstructured mesh which consists primarily of hexahedra. This second step includes the volumetric discretisation of fractures and faults by multiple layers of large aspect ratio prism and hexahedral elements. We present new isoparametric finite element-based finite volume stencils which eliminate the storage of the finite-volume mesh while facilitating an efficient accumulation of the solution matrix. The resulting discretisation is suited for the efficient solution of flow and transport equations in highly heterogeneous geological models. Due to the application of algebraic multigrid, solution time scales logarithmically with the number of unknowns. We analyze our workflow for models of variable complexity arriving at error estimates for the finite element spatial discretisation as a guide for mesh refinement and the accuracy of the finite volume integrations. We also demonstrate it on a field data-based model of intensely fractured rock.
3 Fracture Propagation

Contents

3.1 Abstract .................................................. 44
3.2 Introduction ............................................. 44
3.3 Governing Equations ................................. 49
  3.3.1 Fracture representations ......................... 50
  3.3.2 Displacements, stresses, and strains ............ 52
  3.3.3 Discretisation ...................................... 53
  3.3.4 Isoparametric quadratic triangles ............... 54
  3.3.5 Failure and propagation criteria ................. 56
  3.3.6 Propagation angle ................................ 59
  3.3.7 Stress intensity factors ........................... 60
3.4 Fracture growth methodology ...................... 62
  3.4.1 Propagation algorithm ............................ 64
  3.4.2 Initial flaws ..................................... 65
  3.4.3 Geometric handling of fracture propagation .... 67
  3.4.4 Fracture tips .................................... 68
  3.4.5 Fracture advance .................................. 68
  3.4.6 Fracture intersection ............................. 70
  3.4.7 Fracture closure .................................. 72
  3.4.8 Algorithm implementation ......................... 73
3.5 Adaptive remeshing ................................. 73
  3.5.1 Element quality-based mesh adaptivity .......... 74
  3.5.2 Fracture-to-mesh and mesh-to-fracture ........... 76
  3.5.3 Mapping between meshes ......................... 79
3.6 Fracture characterisation ........................... 81
3.7 Testing and benchmarks ............................. 82
  3.7.1 Validation of the elasticity kernel .......... 83
  3.7.2 Validation and verification of the crack propagation kernel 83
  3.7.3 Benchmarking eight datasets .................... 85
  3.7.4 Performance analysis ............................. 89
3.8 Fracture growth applications ..................... 91
3.1 Abstract

We describe a finite element-based method to propagate multiple fractures simultaneously. The algorithm is iterative and it simulates sub-critical quasi-static crack propagation. The rock matrix is homogeneous and isotropic, and behaves linear elastically. The algorithmic cornerstones are a failure criterion, a propagation criterion, and a propagation angle. Fracture geometries are kept track of independently of the mesh allowing geometric handling of fracture arrest, closure, and coalescence. Fracture aperture is an emergent property of the model. The mesh is adaptively remeshed to capture variations in the geometry and displacement field. Generated patterns reproduce observed crack paths in physical experiments. Algorithm efficiency scales linearly. The model reproduces en-echelon crack linkage, fracture hooking, and orthogonal tip approximation patterns. Shrinkage simulations produce incipient polygonal cells.

3.2 Introduction

Interest in understanding co-operative behaviour during fracture growth and pattern formation extends across a variety of application fields, including structural analysis in civil engineering [e.g. Bazant and Verdure, 2007], design of composite materials [e.g. Camanho et al., 2006], nuclear waste disposal risk assessment [e.g. Shen et al., 2004], and engineering of naturally fractured reservoirs [cf. Zoback, 2007]. The latter often contain multiple generations of fractures formed during progressive brittle deformation [Pollard and Aydin, 1988]. Figure 3.1 shows examples of real fracture
sets found the Jurassic rocks exposed along the Bristol Channel coast, UK. The different sets interconnect and form complex structures that control the properties of the fractured rock mass [e.g. Belayneh et al., 2006, 2007].

Figure 3.1: Fractures in nature. Pictures of joints (left) and mineralised veins (right) at Kilve beach, Bristol Channel coast, UK [cf. Belayneh and Cosgrove, 2005; Rawnsley et al., 1998].

Stochastic modelling is an approach to generate models that capture the statistical properties of natural fracture patterns. Dershowitz and Einstein [1988] developed a widely used algorithm that creates randomly distributed planar fractures with size and orientation capturing the statistics established by fieldwork. Figure 3.2 shows an example of a stochastically generated fracture dataset. These often have a deficit of physical realism due to the lack of fracture geometric self-organisation during growth [Pollard and Aydin, 1988]. They reflect no cooperative behaviour since fractures are placed instead of being grown. Srivastava et al. [2005] developed a more sophisticated data-informed geostatistic model that pseudo-randomly grows cracks honouring geometric field data. Masihi et al. [2005] described a technique to produce mechanically informed pseudo-random patterns by applying a simulated annealing algorithm to generate select fracture datasets. Rives et al. [1994] published a set of rules to generate fracture sets based, not only on statistics, but also on geological constraints such as strain and curvature attributes. In contrast to geomechanically
generated fractures, pseudo-random models do not capture emergent behaviour, such as alignment and coalescence, that arises from fracture propagation during growth. Mechanically based models have been historically labelled as being prohibitively slow. However, current computational capabilities enable geomechanical modelling of fracture patterns as a viable technique of investigation [Ingraffea and Saouma, 1985; Olson and Pollard, 1989; Olson, 1993; Renshaw and Pollard, 1994b; Belytschko and Black, 1999; Huang et al., 2003].

Figure 3.2: Stochastically generated dataset. Two perpendicular fracture sets mimic perpendicular fracture sets.

Analytical methods for the quantification of deformation, such as the ones in Jaeger et al. [2007], are usually formulated for a set of constrained geometries and boundary conditions. In contrast, numerical methods, such as the finite element method (FEM), provide a geomechanically flexible approach to this problem by discretising the governing equations in their weak form [Courant, 1942; Zienkiewicz and Taylor, 2000]. Interest in the numerical modelling of fracture using the FEM probably initiated with the work of Clough [1962] and Sims et al. [1964]. Their work describes the first numerical fracture propagation technique: the nodal release method. It was used to study the effect of stress of a crack on the wall of a dam. The algorithm splits finite elements along their faces as the crack propagates through them. Therefore, the crack path depends on mesh refinement and topology.

Since then, various numerical methods have become available to model single and multiple crack propagation. Ngo and Scordelis [1967] modelled cracks as discrete entities using the FEM. Nilson [1968] devised a method to propagate a single crack by disconnecting mesh nodes along a predefined path. Ingraffea and Saouma [1985]
introduced remeshing to better capture emerging fracture shapes. Initially remeshing only around crack tips [Wawrzynek and Ingraffea, 1989], later in the entire domain [Bittencourt et al., 1996; Ingraffea and Wawrzynek, 2003]. However, these methods are semi-automatic remeshing procedures as they rely on manual user intervention when fractures intersect or reach a boundary.

The difficulty and computational expense inherent to remeshing procedures motivated the development of sub-grid crack propagation models. Non-geometric methods, originally devised by Rashid [1968], do not represent fractures by subdividing the domain into fractures and matrix. Instead, they focus on the non-linear behaviour of the cracking specimen. Methods that adhere to this approach include: damage mechanics, which emphasises stiffness degradation of the sample [Mazars, 1984]; the microplane model, where the constitutive law is expressed in terms of modified stress and strain vectors as opposed to tensors [Carol et al., 1997]; nonlocal continua, in which the stress at a point is a function of the strain history within its neighbourhood [Pijaudier-Cabot and Bazant, 1987]; and combinations of the above [e.g. Hansen et al., 2001]. These alternative sub-grid crack representations were developed mainly to avoid costly mesh reconstruction procedures. Belytschko and Black’s [1999] extended finite element method (XFEM) avoids remeshing by representing cracks discretely but independently of the mesh. The main advantage of this sub-grid representation is that cracks are not meshed, but only separately tracked avoiding remeshing of the domain. Disadvantages include poor definition of the internal fracture domain, complexity introduced by intra-element fracture tips, and fracture intersection [Asferg et al., 2007]. Importantly, in sub-grid representations elements are only designed to handle a single displacement discontinuity per element. Thus, two very proximal crack walls would either have to be contained in two very small neighbouring elements, or would have to partition the element in more than two domains. A coalescing fracture tip further introduces intra-element three-way partition of unit complexities. An additional problem is introduced by the degree of refinement that the initial, static mesh requires in order to resolve stress concentrations at any place of the grid during the simulation. Huang et al. [2003] showed that the required static mesh needs a high degree of refinement to capture arbitrary variations in the stress field. de Borst et al. [2004] and Rots [1991] thoroughly review discrete and continuum models used to represent fractures over the past three decades.

Kachanov [1987] avoided meshing altogether by modelling discrete cracks as polylines juxtaposing analytically computed stresses. Discrete fracture models usually
assume pointy fracture tips, which determine the exponential decay of the surrounding stress field. The stress concentration around the tip is approximated by the stress intensity factors, $K_I$, $K_{II}$, and $K_{III}$, for crack opening modes I, II, and III respectively. In the numerical model, these are calculated from the displacement field. Growth occurs when $K$ surpasses the material toughness. For sub-critical crack growth, propagation occurs earlier, when $K$ reaches a degraded toughness given by the accumulation of damage and other corroding processes around the tip [Atkinson, 1984]. Renshaw and Pollard [1994b] developed a multiple fracture growth model combining a failure criterion and a propagation velocity exponent. They analyzed the effect of crack speed on the concentration of elastic energy stored around fracture tips in the final pattern and validated their model by predicting fracture sets grown in a coated acrylic sheet. They also observed that generated patterns exhibit the same statistics as in fracture outcrops. However, their analytical model is only applicable to cracks with straight paths. Olson and Pollard [1989] later extended it to arbitrarily shaped cracks by using the boundary element method (BEM) to compute stress intensity factors. Olson [2004] further incorporated the effect of mechanical layer thickness by modifying the BEM fracture interaction function using an empirical relationship.

As these important methods continue to evolve the alternative presented here relies on recent advances of continual remeshing methods. In this chapter, we describe a discrete fracture propagation methodology to model quasi-static crack growth. The building blocks of the algorithm are: a fracture criterion, a propagation criterion, and a propagation angle. We combine the following approaches from the literature: a sub-critical crack failure criterion; a propagation criterion based on the weighing of velocities with fracture length [Broek, 1986; Kachanov, 1987; Atkinson and Meredith, 1987a], which has been proved to yield realistic fracture patterns by Olson et al. [2001b] and Renshaw and Pollard [1994b]; and a propagation angle determined by the maximum circumferential stress [Cotterell and Rice, 1980]. Deformation is solved numerically by means of a finite element-based incremental remeshing fracture propagation technique, based on the original single-crack propagation studies of Wawrzynek and Ingraffea [1989] and Bittencourt et al. [1996]. The mesh automatically adapts to the evolving geometry. The elliptic Griffith [1921] fracture tip shape is approximated by a wedge tip. As Barenblatt [1959] and Dugdale [1960], we assume that cohesive forces due to progressive weakening accumulate in a small zone surrounding the tip. We assume, as in the Barenblatt model, that this cohesive zone is small relative to the size of the crack and that, therefore, behaviour can be
described within the limits of the elasticity theory. The algorithms are embedded in a methodology that rigorously computes stress fields of growing discrete fracture representations suitable for \textit{a posteriori} fluid flow simulations. Figure 3.3 shows a set of geomechanically propagated fractures using this methodology.

![Figure 3.3: Geomechanically generated fractures. Two hundred fractures are grown by applying tensile stresses at the top and bottom boundary of the model. The matrix is rainbow colour coded with the mean stress values. Red values around tips reflect high stress concentrations.](image)

The remainder of this chapter is organised into the following sections. Section 3.3 describes the governing equations of the deformation including how we compute the displacement field with the FEM, and derive the stress intensity factors. Section 3.4 describes the fracture growth algorithm. It is followed by a description of the geometric representation of the fractures in Section 3.4.3. Then, Section 3.5 lays out the details of the adaptive remeshing procedure required at every growth iteration. Section 3.6 describes the concepts used to characterise the geomechanically generated datasets. Section 3.7 presents a set of selected tests used to validate and verify our model, as well as results of diverse benchmarks that qualify and quantify the performance of the implementation. Section 3.8 shows diverse application examples of the algorithm. Finally, Section 3.9.3 discusses the extension of the algorithm to 3D.

### 3.3 Governing Equations

This section is an overview of the implemented equations that govern the mechanics of elastic deformation and multi-fracture sub-critical crack growth. It includes a review of the material model and finite-element based deformation kernel as well as a description of the failure and propagation criteria. It also describes the numerical computation of the stress intensity factors, and concludes with a summary of the underpinning assumptions of the methodology.
3.3.1 Fracture representations

Numerical models for fracture propagation rely on materials. These can be classified into three basic groups [Jirasek, 2006]: continuous, discrete, and mixed (see Figure 3.4). In the continuous model, discontinuities are represented by a change in the local constitutive relations. In the case of fracture, the material is locally weakened. Continuous models include elastic, plastic, and, in general, constitutive inelastic models such as the damage model. In a discrete model, the mesh domain is cut along the fracture interface. Discrete models include finite element ones with remeshing [Ingraffea and Wawrzynek, 2003], distinct element [Cundall and Strack, 1979], granular [e.g. Kuhl et al., 2000], particle [Bazant and Kazemi, 1990; Plesha and Aifantis, 1983], lattice [Schlangen, 1993], and probabilistic models [Rossi and Wu, 1992]. Mixed models combine continuous and discontinuous quantities to represent inelastic deformation. In the presented discrete material model the matrix is represented by a continuum of elastic constitutive relations, and fractures are represented as displacement discontinuities in the FEM domain. Crack faces are assumed to be traction free and cohesion is not taken into account. Small displacement, quasi-static crack growth is modelled within an isotropic, homogeneous, and linear elastic material.

Alternative discrete fracture methods include the boundary element method, which is a discrete approach based on Green’s analysis of stress fields around crack tips only requiring the definition of fracture planes and model boundaries in the simulations [Olson, 1993; Renshaw and Pollard, 1994b; Renshaw, 1996]. There are several caveats to this method. Each fracture is represented by a set of equal length segments [Pollard and Olson, 1989], causing fracture growth to be constrained to a minimum capped by the initial discretisation of the boundary domain. The matrix domain is assumed to be homogeneous, and subsequently, variable material properties cannot be represented. Furthermore, the method is not readily extensible to non-linear problems.

The extended FEM is based on the partition of unity method that enriches the FEM shape functions to capture the displacement discontinuities in the domain [Babuska and Melenk, 1997; Melenk and Babuska, 1996; Belytschko and Black, 1999]. The main advantage of the XFEM is that it does not require remeshing, and multiple material domains can be handled by one static mesh. However, in order to resolve stresses at emerging fracture tips, the mesh needs to be well refined; therefore, it requires an initial overall refined mesh or the implementation of an adaptive meshing algorithm. This, however, contradicts the initial advantage of avoiding remeshing.
Mesh-free methods, initially introduced by Lucy [1977], do not require an *a priori* definition of the connectivity of the nodes. They are popular because they do not require an initial mesh to capture emergent behaviour. Belytschko and Tabbara [1996] devised probably the first mesh-free discrete fracture propagation method. As in XFEM, mesh-free methods represent fractures as jumps in the displacement field. Ventura et al. [2002] extended the method to support kinked and curved cracks. More recently, Bordas et al. [2008] extended it to 3D and combined with non-linear constitutive relations, and Rabczuk et al. [2007] applied it to multiple crack initiation and growth in 3D. However, cracks are represented as a surface continuum derived from background cells. The method requires some local background meshing procedures. Moreover, the complex nature of their shape functions and complexity associated to neighbour searches make them considerably more time-consuming than their mesh-based counterparts [Fries and Matthies, 2004]. Additionally, mesh-free methods do not scale well to distributed systems [Griebel and Schweitzer, 2002].

By combining the FEM with continual remeshing and the sub-critical crack growth method, we formulate a finite element-based mixed material model for
the simultaneous propagation of multiple discrete fractures. It tackles emerging geometry and dynamic mesh refinement requirements by automatically remeshing the domain at each growth step. The matrix domain constitutive law models an isotropic, homogeneous, linear elastic medium. However, the fracture propagation model is inherently non-linear because growing fractures represent irreversible brittle deformation. This method not only generates accurate displacement and stress fields, but it is also extensible to material anisotropies and nonlinearity, permits the visualisation of material domains and computed stress fields, and can be integrated to our existing multi-physics framework CSMP++ [Matthai et al., 2001].

3.3.2 Displacements, stresses, and strains

The elastic deformation of a body in two-dimensions is governed by the following set of equations [Cook et al., 1989]. The linear elastic stress-strain constitutive equations for homogeneous and isotropic media relate stress build-up to deformation [Timoshenko and Goodier, 1934]

\[ \sigma = D (\varepsilon - \varepsilon_0) + \sigma_0 \]  

(3.1)

where \( \varepsilon = \{\varepsilon_{xx}, \varepsilon_{yy}, \varepsilon_{xy}\}^T \) is the strain vector, \( \sigma = \{\sigma_{xx}, \sigma_{yy}, \sigma_{xy}\}^T \) is the stress vector, \( \sigma_0 \) and \( \varepsilon_0 \) are the initial stress and strain vectors respectively, and \( D \) is the linear elastic material stiffness matrix.

The strain-displacement equations relate deformation to displacement

\[ \varepsilon = \partial u \]  

(3.2)

where \( u = \{u, v\}^T \) is the displacement vector, and the kinematic operator \( \partial \) is defined as

\[
\partial = \begin{bmatrix}
\frac{\partial}{\partial x} & 0 \\
0 & \frac{\partial}{\partial y} \\
\frac{\partial}{\partial y} & \frac{\partial}{\partial x}
\end{bmatrix}.
\]  

(3.3)

At force equilibrium,

\[ \partial \sigma + F = 0. \]  

(3.4)

This means that stresses are in equilibrium with body forces \( F = \{f_{xx}, f_{yy}, f_{xy}\}^T \), such as the ones produced by gravity, dilatation, acceleration, and others.

Using the FEM we simulate deformation of objects that obey these relations with arbitrary geometry and boundary conditions. We solve \( u \) for a set of initial conditions and fixed displacements at the boundaries. The model can be calibrated to specific
material properties by defining a $D$ that captures its constitutive relation. The following section summarises the discretisation of this model.

### 3.3.3 Discretisation

An unstructured grid subdivides the geometric domain into discrete elements. The principal of virtual displacements for a deformable body, also known as principle of complementary virtual work, can be derived from Equations 3.2, 3.4, and 3.1. It states that a body subjected to an infinitesimal virtual displacement remains in force equilibrium, expressed as

$$\int \{\delta \varepsilon\}^T \{\sigma\} dV = \int \{\delta u\}^T \{F\} dV + \int \{\delta u\}^T \{\Phi\} dS \quad (3.5)$$

where $\delta$ is a virtual differential, and $\delta \varepsilon = \{\delta \varepsilon_x, \delta \varepsilon_y, \delta \varepsilon_{xy}\}$ is the vector of virtual strains, and $\delta u = \{\delta u, \delta v\}$ is the vector of virtual displacements. Thus, for a quasi-static $\delta u$, in equilibrium, the increment of strain energy that is stored is equivalent to the sum of the work by body forces $\{F\}$ and surface tractions $\{\Phi\}$.

For a set of nodal displacements $\{d\}$ we interpolate $\{u\}$ as follows

$$\{u\} = [N] \{d\} \quad (3.6)$$

it follows

$$\{\varepsilon\} = [\partial] \{u\} = [\partial] [N] \{d\} = [B] \{d\} \quad (3.7)$$

where $[B]$ are the shape function derivatives. It follows that

$$\{\delta u\}^T = \{\partial d\}^T [N]^T \quad (3.8)$$

and

$$\{\delta \varepsilon\}^T = \{\partial d\}^T [B]^T. \quad (3.9)$$

By substituting Equations 3.1, 3.8, and 3.9 into 3.5 we obtain the integral form

$$\int [B]^T [D] [B] dV \{d\} + \int [B]^T \{\sigma_0\} dV - \int [B]^T [D] \{\varepsilon_0\} dV - \int [B]^T \{F\} dV - \int [B]^T \{\Phi\} dS = 0 \quad (3.10)$$

This system of equations is summarised as

$$[k] \{d\} = \{r\} \quad (3.11)$$

53
where $[k]$ and $\{r\}$ are the left- and right-hand sides of the system of equations respectively. It follows

$$[k] = \int [B]^T [D] [B] dV.$$  \hspace{1cm} (3.12)

The material constitutive relation is defined by the stiffness matrix, $D$. In this case, $D$ is defined as a linear elastic matrix. In two dimensions, for the plane strain assumption $D$ is

$$D = \frac{E}{(1 + \nu)(1-2\nu)} \begin{bmatrix} 1 - \nu & \nu & 0 \\ \nu & 1 - \nu & 0 \\ 0 & 0 & 1 - 2\nu \end{bmatrix}$$ \hspace{1cm} (3.13)

and for plane stress, its defined as

$$D = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1 - \nu \end{bmatrix}$$ \hspace{1cm} (3.14)

where $E$ is the elasticity modulus and $\nu$ is Poisson’s ratio.

Applying the FEM these equations are computed on an element by element basis and accumulated into a large $[K]$ matrix of dimensions equal to the number of total degrees of freedom ($=$ number of nodes $\times$ number of dimensions of the solution variable). For further details on how to perform the accumulation of the global matrices we refer the reader to [Cook et al., 1989]. We apply the algebraic multigrid solver (SAMG) to solve the ensuing linear algebraic equations [Stüben, 2001; Stüben et al., 2003]. Once the solution vector $\{d\}$ is computed, we derive the stresses and strains at each integration point by substituting $\{d\}$ into Equations 3.6 and 3.7.

### 3.3.4 Isoparametric quadratic triangles

The domain is discretised using higher order six-node isoparametric quadratic triangles [Taig, 1961]. Isoparametric elements are those whose shape functions are the same as their interpolation functions (see Figure 3.5). Interpolations are made in parametric space and values are mapped back to global space using Jacobian transformations (see Section 2.3.9). These elements yield a highly accurate quadratic interpolation field which captures the variable behaviour of the displacement field.

The nodes of the isoparametric quadratic element in parametric space $(r, s)$, are

$$(0, 0), (1, 0), (0, 1), \left(\frac{1}{2}, 0\right), \left(\frac{1}{2}, \frac{1}{2}\right), \left(0, \frac{1}{2}\right).$$ \hspace{1cm} (3.15)
displacements are computed at the nodes
displacements are interpolated to the ips
strains and stresses are computed at each ip

Figure 3.5: Triangular isoparametric quadratic element. The shape functions are defined on three corner nodes and three mid-side nodes yielding a quadratic interpolation field. Displacements and material properties are defined at the nodes and integration points respectively.

Their quadratic shape functions are defined as

\[
N_1 = (1 - r - s)(1 - 2r - 2s) \quad (3.16)
\]

\[
N_2 = r(2r - 1)
\]

\[
N_3 = s(2s - 1)
\]

\[
N_4 = 4r(1 - r - s)
\]

\[
N_5 = 4rs
\]

\[
N_6 = 4s(1 - r - s).
\]

For quadratic shape functions the gradient field, from which we derive stress and strain, is linear as opposed to linear shape functions which yield constant gradient fields.

In parametric space, integrals in Equation 3.10 can be approximated by using the Gaussian quadrature rule. It states that the integral of an element can be exactly approximated by the sum of the values at the element’s integration points. It follows

\[
\int \psi dA = \sum_{i}^{ip} w_i \psi_i J_i \quad (3.17)
\]

where \(\psi\) is a property value, \(A\) is the area of the triangle, \(ip\) are the Gaussian integration points, \(\psi_i\) is the value of the property at integration point \(i\), and \(w_i\)
is its respective weight. For three integration points, their coordinates in parametric space are

\[(2/3, 1/6) \quad (1/6, 1/6) \quad (1/6, 2/3)\]

and their weights are 1/3.

Thus, Equation 3.12 becomes

\[k = \sum_{i}^{ip} w_{i} [B_{(i)}]^{T} [D_{(i)}] [B_{i}] J_{i}.\]  

(3.19)

Notice that the material stiffness matrix \([D_{i}]\) is an integration point property. Material properties are defined at three different locations for each triangle allowing material heterogeneities to be incorporated into the model. Section 2.3.6 describes with detail the mapping between the parametric and global space.

Triangles around tips are defined as isoparametric quarter point elements to attain a more accurate displacement field resolution [Barsoum, 1976]. Quarter-point elements are quadratic elements that have their mid-side nodes shifted by one quarter toward a predefined tip node. Thus, elements are better suited to capture exponential decaying field around the tips. Section 3.3.7 discusses how we exploit these to obtain highly accurate approximation of stress intensity factors.

### 3.3.5 Failure and propagation criteria

Modelling simultaneous growth of multiple cracks relies on three locally determined criteria: failure, propagation, and angle. Failure criteria examine if a sample will fail, and control whether a fracture continues to propagate at a specific tip. The sub-critical crack growth failure criterion prescribes that a tip will propagate even though the energy around it may not overcome the material toughness by supposing that fatigue and corrosive processes around have progressively weakened it. For fractures, such a criterion is based on the evaluation of the magnitude of the stress intensity factor at each tip [Atkinson, 1984]

\[K_{IO} \leq K_{I} \leq K_{IC}\]  

(3.20)

where \(K_{I}\) the stress intensity factor at the tip, and \(K_{IO}\) and \(K_{IC}\) are the material corrosion limit and toughness, respectively. We assume a lower threshold stress intensity factor of \(K_{IO} = 0.1K_{IC}\) [Olson, 1993; Segall, 1984a; Atkinson and Meredith, 1987b]. Failure criteria are often material property and in-situ conditions dependent. The scheme presented here offers the flexibility to implement failure criteria to capture
more complex material behavior. All simulations presented use the sub-critical failure criterion to determine tip advance.

The propagation of a single crack is simulated by only applying a failure criterion. Every time a tip fails, it advances by a fixed distance. This simulates a crack propagating at a fixed speed, governed by the local stress field around its tips. Figure 3.6 shows an example of a single crack numerically propagating in a homogeneous medium.

However, when more than one crack propagates simultaneously, it is no longer clear if they are all propagating at the same speed or not. Numerical studies show that within a group of cracks, and in order to reproduce patterns found in the field, cracks must grow at different speeds. In fact, in the 60’s Gurney and Hunt demonstrated that for quasi-static growth, the energy required to propagate a fracture is inversely proportional to the root of its size. In physical experiments, maximum speed is related to the maximum fracture length by means of a power law [Broek, 1986; Kachanov, 1987; Atkinson and Meredith, 1987b]. The weighing of the speeds, and therefore, the weighing of the length advances per tip is formalised by a propagation criterion.

![Figure 3.6: Single crack propagation. Single crack propagates governed by a local failure criterion. The matrix domain is discretised by triangles. Scalar field depicts the mean stress at each element.](image)

For each crack we measure the energy release rate, $G$, associated with growth
defined for plain strain as

\[ G = \frac{(1 - \nu^2)K_I^2}{E} \]  

(3.21)

and for plane stress as

\[ G = \frac{K_I^2}{E}. \]  

(3.22)

We then monitor the tip with the maximum energy concentration

\[ G_{\text{max}} = \|G\| \]  

(3.23)

where \( G_{\text{max}} \) is the \( L \) infinity norm of \( G \). \( G_{\text{max}} \) applies to the tip that is growing the fastest.

We choose a well-established propagation criterion originally defined by Charles [1958], and extended by Renshaw and Pollard [1994b], to compute the distance a crack tip will extend at any propagation step. This criterion relates the energy accumulated around a specific tip with the maximum energy of all tips and restricts growth by weighing it with an empirical velocity index, \( \alpha \). This implies that tips with the highest energy in the fracture set advance significantly faster than the rest. It follows that

\[ l_{\text{adv}} = l_{\text{max}} \left( \frac{G}{G_{\text{max}}} \right)^{\alpha=0.35} \]  

(3.24)

where \( l_{\text{adv}} \) is the propagation length, and \( l_{\text{max}} \) is the maximum length increase at any propagation iteration. Renshaw and Pollard [1994b] identified by exhaustive experimentation that a velocity index of 0.35 yields realistic fracture patterns. BEM implementations of this method, such as [Olson and Pollard, 1989], can only add fix sized elements at the extremes of fractures. Therefore, they artificially accumulate the advance until it becomes as large as the minimal grid size. In contrast, in our approach, we apply growth increments immediately.

In the original formulation, \( l_{\text{max}} \) is defined in terms of the initial flaw size, \( l_{\text{max}} = 2a_0 \), mainly because it is assumed that the initial flaw size is representative of the fracture process zone, where

\[ 2a_0 \geq l_{\text{max}} \geq h_{fpz} \]  

(3.25)

and \( h_{fpz} \), also known as characteristic length, is the radius of the fracture process zone [Irwin, 1958]. Under this assumption, and since \( G \leq G_{\text{max}} \), it follows that \( l_{\text{adv}} \leq l_{\text{max}} \). If \( a_0 \) would be much larger than this characteristic length, \( l_{\text{adv}} \) would become much larger than the estimated \( h_{fpz} \) violating the linear elastic mechanics assumption of scale separation. Initial fracture length, in our formulation, is a separate parameter, an upper bound to the maximum propagation length.
The failure criterion relies on the nature of the stress concentration at the tip. In the most basic case, if the amount of tensile stress exceeds the strength of the material, the element at the tip yields to propagate the crack. However, in order to model this stress peak, the mesh must be sufficiently resolved to capture the steep gradient in the solution field. At the same time, the mesh size around the tip should not be smaller than the material-specific Irwin characteristic length [Irwin, 1958; Bazant, 2000], which approximates the local fracture process zone. Thus, we can define a range of triangle sizes required for the correct evaluation of the failure criterion around a fracture tip. For isoparametric quadratic elements, material points are defined at locations close to the tips. This, added to the definition of highly accurate quarter-point elements reduces greatly the need for large refinements around the tips. Therefore, it can be used to avoid over refinement of the tip area. However, the arbitrary nature of crack growth often tends to create small segments around the tip that yield an automatic refinement of the region. This collateral refinement, given by an underlying dynamic and detailed geometry, generates as a by-product a more exact solution around the tip, and therefore, augments the accuracy of the evaluated failure criterion.

3.3.6 Propagation angle

The stress field around the tips determines the angle at which the tip extends [Cotterell and Rice, 1980]. We define the propagation angle, $\theta$, as the angle between the fracture plane near the tip and the maximum principal stress. Thus, the crack is guided by the locally measured maximum circumferential stress ahead of its tip.

Figure 3.7 is an example of two propagating fractures that approximate, their tips react to the perturbation in the stress field caused by each others’ presence. Thus, they curve first away form each other, until they overcome the compressive field around the tip, then toward each other at a 90-degree angle seeking termination.

In order to evaluate the stress state surrounding the tip, we compute a weighted average of the stress tensors at the closest integration points to the tip. We obtain a single stress value approximation at the tip that captures the trend of the field around it. The propagation angle is approximated by the direction perpendicular the maximum stress eigenvector.
Figure 3.7: Numerical simulation of hooking fracture tips. Stress fields around the tips are perturbed by the neighbouring cracks causing paths to deviate from a straight course.

### 3.3.7 Stress intensity factors

Stress intensity factors characterise the local stress field around a crack tip. We implement the quarter point displacement technique (QPDT) [Henshell and Shaw, 1975] to compute stress intensity factors $K_I$ and $K_{II}$ (Pa m$^{1/2}$). The QPDT has been shown to yield accurate approximations of $K$ around the tip as compared to its counterparts: the displacement correlation technique and the displacement extrapolation technique [Lim et al., 1992; Fehl and Truman, 1999]. The alternative J-Integral method has been shown to be even more exact [Rice, 1968], but also considerable more computationally expensive.

The theoretical description of the displacements at the tip for mode I is approximated by [Sih and DiTommaso, 1985] as

\[
\begin{align*}
    u &= \frac{K_I}{4G_s} \sqrt{\frac{r}{2\Pi}} \left( (2\kappa - 1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right), \\
    v &= \frac{K_I}{4G_s} \sqrt{\frac{r}{2\Pi}} \left( (2\kappa - 1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2} \right)
\end{align*}
\]  

(3.26)

where $u$, $v$, $r$, $\theta$, and $G_s$ are the displacements parallel and perpendicular to the fracture axis (m), the distance from the tip (m), the angle measured from the fracture axis, the shear modulus

\[
G_s = \frac{E}{2(1 + \nu)},
\]  

(3.27)

and

\[
\kappa = \begin{cases} 
2 - 4\nu & \text{for plane strain} \\
(3 - \nu)/(1 + \nu) & \text{for plane stress}
\end{cases}
\]  

(3.28)

Figure 3.8 illustrates these quantities around a tip. Similarly, for pure mode II,
displacements near the tip are approximated by

\[
\begin{align*}
u &= \frac{K_{II}}{4G_s} \sqrt{\frac{r}{2\Pi}} \left( (2\kappa + 3) \sin \frac{\theta}{2} + \sin \frac{3\theta}{2} \right) \\
v &= -\frac{K_{II}}{4G_s} \sqrt{\frac{r}{2\Pi}} \left( (2\kappa - 3) \cos \frac{\theta}{2} + \cos \frac{3\theta}{2} \right)
\end{align*}
\] (3.29)

\[
\text{plane of the fracture}
\]

Figure 3.8: Quarter point displacement technique. The stress intensity factor is approximated by assuming an exponential decay of the stresses around the tip. A and B are the closest mid-side nodes to the tip, \(\theta\) is the angle measured from the axis of the fracture tip, and \(r\) is a distance measured from the tip.

The QPDT relies on the accurate modelling of the \(\sqrt{r}\) displacement variation near the tip. As described in Section 3.3.4, elements at the tip are quarter point triangles in which the mid-side nodes are displaced from their original position toward the tip. Thus, a single element can capture the strong displacement variation occurring around the tip. Figure 3.9 illustrates the rosetta of quarter point elements around it.

For mixed mode loading, Equations 3.26 and 3.29 uncouple when \(\theta = 180^\circ\) and \(r = r_A\), where \(r_A\) is the distance between the tip of the element and the closest node, and A and B are equidistant from the tip. Thus, \(r_A = r_B\).

The QPDT method relies on the correlation between the theoretical values and the values of the displacement field at the two closest nodes to the tip, A and B. It approximates the stress intensity factors as follows

\[
\begin{align*}
K_I &= \frac{2G_s}{\kappa + 1} \sqrt{\frac{2\Pi}{r_A}} (v_A - v_B) \\
K_{II} &= \frac{2G_s}{\kappa + 1} \sqrt{\frac{2\Pi}{r_A}} (u_A - u_B)
\end{align*}
\] (3.30)

where \(u_A\) and \(u_B\) are the displacement vectors at A and B projected onto the fracture axis. Similarly, \(v_A\) and \(v_B\) are the velocity vectors at A and B projected onto the plane perpendicular to the fracture axis. This method yields accurate approximations.
of $K_I$ within 4% of the theoretical values for tension-only experiments and within 2% of theoretical values for shear-only experiments [Fehl and Truman, 1999]. Earlier, Lim et al. [1992] showed that this fast approximation yields results within 1% of the theoretical values, when the mesh was refined around the tip so that $r_A/l \sim 0.1$ where $l$ is the crack length. Mesh refinement around the crack tips contributes to reducing the error associated with this calculation.

Finally, the underpinning assumptions of FEM formulation are

- the material is brittle, homogeneous, isotropic, and linear elastic;
- deformation is simulated in 2D assuming plane strain;
- fracture tips are initially “V” shaped, linearly approximating an elliptical shape;
- damage zone is significantly smaller than the size of the fracture;
- propagation is quasi-static and strain rate independent;
- and, there is no cohesion/traction between fracture walls.

### 3.4 Fracture growth methodology

As fractures grow, they interact by creating complex stress perturbations that eventually start to overlap. Figure 3.10 (b) is an example of interacting fracture
tips found in the diatomite cap rock shale of the Orcutt field in California. Fractures are stained with oil, highlighting their paths. Figure 3.10 (a) shows similar crack interactions in concrete. A model that can capture these evolving patterns must allow cracks to grow as directed by the local stress surrounding their tips.

![Image](a)

![Image](b)

**Figure 3.10**: Fracture tip interaction in diatomite and concrete. (a) Fractures interact on a concrete structure. We observe hooking between fracture tips. (b) Two cracks stained with oil interact as they approximate. The tiny fractures arise in one of the shale outcrops that cap the shallow Orcutt oil field in California, USA.

In this section, we discuss the algorithm used to model crack propagation. Paluszny and Matthai [2008b] describe an earlier version of this crack propagation methodology. During the simulation, the model is sub-divided into the propagation and flaw area. The model region where flaws are initially placed is the **flaw area**. The model region in which fractures can grow is the **propagation area**. The objective of the latter is to avoid artificial boundary effects. It is equivalent to the total model shrunk by 1-5%. In the case where a fracture reaches the limits of the propagation area, its growth is stopped to avoid violation of the FEM continuity assumption. Thus, for
uni-axial tension models, elongated propagation area shapes are preferred because fractures can continue to grow without reaching any boundary.

3.4.1 Propagation algorithm

The development of algorithms for growing multiple cracks using a non-linear finite element method faces three major challenges [Yang and Deeks, 2007]: a well validated crack propagation criterion; a robust solver that supports vector variables, is scalable, and can handle non-linear numerical systems; and efficient and accurate mapping of nodal properties between meshes. Although the work presented in this section focuses on the development of a solution feasible for linear elastic fracture mechanics, it addresses all the latter issues. In realistic subsurface models, fractures coexist with other inelastic features such as compaction or shear bands and other damage localisation regions which should ideally be incorporated into the deformation model. A discrete fracture approach allows the inelastic behaviour to be modelled at the sub-grid level, while cracking is captured as a dynamic property of the grid. We present a geometric-based fracture propagation algorithm that simultaneously grows multiple fractures that populate a model in response to displacements applied at the boundaries.

Fractures yield a dynamic geometry that is kept track of in the form of two-dimensional polygons. The mesh is adapted to capture the emerging fracture geometry. For a fixed set of boundary conditions, the model is iteratively deformed until no more growth is registered. This is equivalent to a high-level Picard iteration that allows fractures to advance until the energy at the tips is not large enough to induce more propagation. While this equilibrium state is not reached, the mesh nodes are not moved. However, every time the geometry changes, the previous stress state is invalidated and new updated stresses are recomputed. Once fractures cease to grow, the model nodes are moved to capture deformation.

In summary, the simulation of fracture propagation involves the following steps

1. Generating a set of randomly positioned flaws with a uniform distribution and lengths with a Gaussian distribution.


3. Application of displacement boundary conditions.

4. Solution of the deformation partial differential equations using a linear elastic constitutive law.
5. Computation of displacement, strain, and stress fields.

6. Computation of stress intensity factors, propagation lengths, and directions at each tip of each fracture.

7. Extension of all propagating fractures.

8. Regeneration of the mesh and mapping of the displacement field onto the new mesh.

9. Re-computation of stresses and their accumulation as the stress state determined by the previous deformation step.

10. Repeating the previous steps until there is growth equilibrium. Initial flaws develop into fractures until the energy accumulated around their tips ceases to induce propagation.

11. Equilibrium occurs when no growth is recorded for a fixed boundary displacement.

12. At this stage, we increase the displacement at the boundaries and repeat all steps.

### 3.4.2 Initial flaws

Our model does not nucleate cracks, it only propagates them. Therefore, we place seeds in the form of flaws or pre-existing cracks that grow when tip stresses exceed a critical value. Initially, we populate each model with a set of tiny flaws. The size of these flaws depends on the experiment. For example, for a 1m × 4m specimen, we define flaws of ~0.003m length. They represent pre-existing weaknesses in the intact rock. These flaws not only have random locations, as they would have in a physical specimen, but also have random sizes. However, it is known that for a quasi-heterogeneous material, these approximately follow a Gaussian size distribution [Underwood, 1970].

We use diamond shaped flaws because they are simple, and for high aspect ratios (e.g. 10, 10²), they concentrate stress at their tips. They mimic thin penny-shaped microcavities present in brittle rocks [Herrmann, 1990]. Flaws can have fixed or random orientations. Figure 3.11 illustrates three different scenarios: (a) flaws are randomly distributed and randomly oriented, (b) flaws are randomly distributed, but
all parallel, and (c) flaws align in specific regions of the model. This scenario resembles damage planes that concentrate inelastic deformation during compression.

Flaws are not placed on pre-defined grid locations. They are randomly positioned within the entire flaw area. When generating these flaws, their location and lengths are an input of the model. Consistency checks make sure that these do not overlap, and that they honour numerical proximity constraints. The placement of crack centres is not entirely random. Two flaw centres placed too close to each other would artificially generate one larger flaw, with a length outside of the initial desired distribution. Extreme proximity of two or more flaws also over-constrains the mesher by imposing tiny distances to the geometric model, which coerces the mesher to create minuscule triangles between flaws. This produces an over-refined mesh at these difficult locations. In the worst case, the distance between two cracks is so small that the mesher cannot resolve the space between them, and fails.

The methodology to position the flaws is as follows. Using a uniform random number generator, we compute position candidates for each new flaw:

$$f_{ij} = \text{random()},$$

where $f_{ij}$ is the position of the flaw. If any point in the new flaw is closer to another flaw than a minimal distance the insertion fails, else, it succeeds. It follows that

$$\forall f_{ij} \forall f_{ij} \ | f_{ij} - f_{ij} | > d_{\text{min}},$$

where $f_{ij}$ and $f_{ij}$ are the positions of any two flaws and $d_{\text{min}}$ is a minimum separating
where \( l_{f_{ij}} \) and \( l'_{f_{ij}} \) are the lengths of \( f_{ij} \) and \( f'_{ij} \), respectively, and \( s_{\text{max}} \) is the maximum spacing between two cracks. The edge length of the minimum triangle representable in the grid sets an upper limit to \( s_{\text{max}} \). The initial density of the flaws is decreased by assigning a larger \( s_{\text{max}} \). This creates more evenly spaced flaws, while a smaller \( s_{\text{max}} \) induces smaller flaw clusters to spontaneously form.

Initial flaw aperture is set to a tenth of the flaw’s initial length

\[
a_{f_{ij}} = \max \left\{ a_{\text{min}}, 0.1l_{f_{ij}} \right\}
\]

where the aperture value, \( a_{f_{ij}} \), is bound by a numerical minimum, \( a_{\text{min}} > 10^{-7} \text{m} \), for single-precision calculations.

In order to accurately generate the uniformly distributed random numbers we use the extremely fast and reliable Mersenne Twister algorithm [Matsumoto and Nishimura, 1998], implemented in the open-source Altruist C++ library [Fog, 2000]. Flaw sizes are generated using the stochastic random number generator of the Altruist library. Flaw size distribution is Gaussian.

### 3.4.3 Geometric handling of fracture propagation

Each fracture is represented by a two-dimensional closed polygon using the boundary representation technique (BREP) [Paluszny and Matthai, 2008b]. Thus, it is a set of polylines that enclose a finite region that corresponds to the inside of the fracture. Figure 3.12 illustrates the body of a crack with three tips. A fracture cluster is a set of multiple cracks that have intersected with one another. Each fracture is stored as an independent object in the model. Tips are tracked individually during growth, any node in the BREP may become, at any stage of growth, a new crack. Tips are determined by small angles in the BREP. The small angle hints at the fact that the tip will propagate. Fracture aperture is an emergent property given by the separation of its walls.

Boolean operations, such as intersection and merge, handle the geometric coalescence of fractures. Geometric housekeeping of the dataset is key to swift and robust automatic remeshing. This includes the identification and removal of small overlapping or proximal segments that, if remnant would cause terminal errors during the creation of the mesh. After each fracture propagation step the fracture BREP is updated using the new deformed FEM mesh. The simulator adjusts the nodes of the
Figure 3.12: Polygonal representation of a crack. Fractures are represented by 2D polygons. Tips are tracked during the crack growth by finding the corners of the polygon that resemble a crack tip as found in nature. Nodes that lie on the tips of the star-shaped polygon, on segments that form very large continuity angle $\beta$ such as $t_{n-1}$, are fracture tips.

Mesh to represent the new position of the fractures caused by applying forces to the specimen. This allows tracking wall movements and monitoring of fracture aperture. At some locations aperture reduces to the point of local fracture closure.

3.4.4 Fracture tips

In numerical simulations, tips are emergent properties of the fractures. They are not predefined in shape, position, or quantity at any time. A flaw initially has two tips. Node tips advance during growth, and their intersection with other fractures triggers coalescence forming multi-tip fracture clusters. If the angle $\beta$ between any two adjacent segments of the fracture polygon is larger than minimum $\beta_{\text{min}}$, the point between the two segments is dynamically considered a fracture tip. Figure 3.12 illustrates the difference in magnitude between the angles at each fracture kink. The value of $\beta_{\text{min}}$ is set to $355^\circ$. This means that the maximum aperture of a tip wedge is capped to $5^\circ$. Tips at the boundaries are at a dormant state, and do not grow.

3.4.5 Fracture advance

Fractures grow after each iteration step by adding new wedge tips. After each step, $l_{\text{adv}}$ and $\theta$ determine the extension of the fracture BREP. If $l_{\text{adv}} < l_{\text{min}}$, where $l_{\text{min}}$ is a given numerical tolerance, the tip position is moved. In our case, $l_{\text{min}} = 10^{-7}$ m, as we use a mesher with single precision, floating point arithmetic (7 digits of numeric accuracy). Else, if $l_{\text{adv}} \geq l_{\text{min}}$ we extend the fracture representation by adding a new tip at the extreme of the BREP. Figure 3.13 illustrates the geometric handling of an advancing tip. Adding a new tip, $t_n$, includes splitting the previous tip, $t_{n0}$, into two
nodes, \( t_{n-1} \) and \( t_{n+1} \), which are at a minimal distance \( l_{\text{min}} \), and the segment \( t_n \overline{t_{n+1}} \) is perpendicular to the plane inclined at \( \theta/2 \) degrees from the fracture axis.

![Figure 3.13: Extension of the fracture polygon.](image)

Fracture tips deactivate when they reach the growth area boundary. In this case, fractures will cease to grow through the passive tip. We assume that fractures that arrest at the border of the propagation area continue to grow, and are not taken into account to compute crack propagation velocity in further iterations. In this way, the fracture set can continue to propagate even though some of the fractures have traversed the propagation area. In theory, the sample would have to be infinitely long to allow continuous propagation. The opening mode fractures formed by this process continue to cast a compressive shadow upon their neighbours and are subsequently included in all mechanical simulation computations except in the determination of \( G_{\text{max}} \).

During growth, the fracture centreline is stored. The centreline captures the skeleton of the fracture shape. The centreline is important because it allows us to generate high quality flow meshes where fractures consist of sub-dimensional line elements. Apertures can be sampled at any point along the centreline in order to assign thickness to the sub-dimensional elements. The centreline polyline can also be interpolated using parametric curves to obtain a smooth fracture representation.

Due to the geometric complexity of the fracture shapes (e.g. contains holes and is non-convex), the automatic extraction of such a skeleton results expensive. Instead, we keep track of the centreline of the growing star-shaped polygon by adding the new
nodes, $t_n$, to the centreline of the previous step

$$c = \langle t_n^A, t_n^{1A}, ..., t_n^{0A}, t_n^{0B}, ..., t_n^{1B}, t_n^B \rangle$$ (3.35)

where the fracture has initially two tips: $t_n^{0A}$ and $t_n^{0B}$, which have extended $i$ and $j$ times respectively.

Automatic tracking during growth is crucial to maintain performance. Every time the mesh is deformed, the new centreline location is interpolated using the displacement field. When fractures intersect, their centrelines also merge.

### 3.4.6 Fracture intersection

We assume traction between surface walls to be zero. Therefore, in our model, a fracture always terminates propagation at intersection with any other fracture, and never propagates through an existing fracture [Dyer, 1988]. When a fracture tip becomes very close to a wall of another, the tip snaps onto the crack wall. If two fractures intersect, their shapes combine by merging their polygonal representations. As more and more fractures coalesce, they form complex cluster polygons with multiple tips. Figure 3.14 illustrates how a fracture $A$ terminates against a proximal free surface of a secondary fracture $B$. This behaviour mimics how an opening mode crack, at intersection with another open crack, ceases to grow and connects both paths. There are three distinct cases: (a) the new fracture tip, $t_n$, is at a distance from the wall which is smaller than the numerical tolerance $l_{\text{min}}$. The tip snaps onto the wall and the fractures merge. If $t_n$ is not snapped, the small distance causes the mesher to produce minuscule, ill-formed elements that diminish the overall quality of the discretisation. (b) The length advance is over predicted and $t_n$ extends beyond the walls of a proximal fracture. $A$ is arrested at intersection by shifting $t_n$ to the closest position inside the secondary fracture $B$. The third and less common case, (c), is when the new $t_n$ is located exactly within the walls of the secondary crack. Cases (a) and (b) are geometrically projected to case (c) by moving $t_n$ into the fracture walls. Two nodes describe the intersection between fracture $A$ and $B$: $t_{n-1}$ and $t_{n+1}$. These are separated by a minimum distance larger than $l_{\text{min}}$. In (c), the original intersection points between the fracture tip wedge and the secondary crack are at a distance smaller than $l_{\text{min}}$. During advance, their positions adjust so that they are separated, and are considered as two separate points for a specified numerical tolerance. Finally, (d), shows the new fracture $k$ which results from the intersection. After the intersection is identified and $t_n$ is adjusted, the bodies of the two cracks are merged.
Figure 3.14: Fracture intersection detail. Fracture A arrests at intersection with a free surface (wall of fracture B). $t_n$ is the predicted new tip position, $t_{n-1}$ and $t_{n+1}$ intersect the free surface. Fracture B is cropped at intersection simulating arrest. In (a), $t_n$ is at a distance smaller than the tolerance $l_{\text{min}}$. It is snapped onto fracture B’s wall. (b) Fracture advance overestimates the distance to the next free surface, $t_n$ is adjusted to a position within fracture B. (c) The fracture tip lies within fracture B but the intersection points, $t_{n-1}$ and $t_{n+1}$, are too close to one another. (d) After the tip and intersection points are adjusted fractures are merged.

Coalescence of fractures is handled by merging their bodies using constructive solid geometry. Thus, the geometry of a new fracture is defined as the union of two previously existing shapes. Namely, the union of the shapes of fractures A and B results in a shape C that encloses all space delimited by A and by B. We rely on the General Polygon Clipper Library (GPC) by Alan Murta, to compute the Boolean operations on our fracture shapes. The GPC is a fast and robust engine designed to perform Boolean operations on non-convex 2D polygons.
Complex fracture intersections may trigger the formation of detached matrix regions. Figure 3.15 illustrates the mechanism by which a double intersection may conduce to the formation of a disconnected mesh region, referred to as blocks. They are of special interest because they introduce a further mechanical complexity to the simulation. As separate blocks, they might, for example, rotate and intrinsically change the aperture distribution of the fractures that initially formed them. This is best captured using the discrete element approach [Zhang and Sanderson, 2001]. Handling of these blocks requires identification of contact points and transduction of stresses. For mechanical simulations, the mesh inside of the fractures is never created; therefore, no forces are directly transduced to emerging disconnected regions. This issue can be addressed by creating a connective mesh between the blocks and the rest of the matrix. An alternative is to implement a discrete method approach to handle deformation due to contact of these separate bodies. In this formulation, block formation is captured by an inner loop in the fracture geometric representation. However, due to the FEM mesh continuity constraint, these blocks cannot be meshed. Thus, we do not handle block rotation and assume that once formed, these become permanently disconnected from the matrix and fixed at their initial position.

![Figure 3.15: Formation of a block. (a) Two fractures approximate. They deflect prior to intersection. (b) Tips start an orthogonal approach toward each other before the double collapse. (c) The cavity in the mesh denotes a disconnected block mesh.](image)

3.4.7 Fracture closure

After deformation, the fracture BREP is updated according to the new deformed mesh. Nodes adjust to represent the new position of the fractures caused by displacing the boundaries. Thus, fracture shapes track wall movements and fracture aperture evolution. At some locations, fracture aperture is extensively reduced indicating local fracture closure.
Fracture closure occurs when the fracture shape is self-intersecting after deformation. This situation, often referred to as mesh tangling, is caused by walls that are over-displaced toward each other due to compressive forces. The tangled mesh can be re-approximated into a closed crack by tracking the wall nodes and reconstructing the closed walls with an interpolated centreline. A Non-Uniform Rational B-Spline (NURBS) approximates the centreline of the fracture by interpolating a centreline through the tangled fracture nodes. We choose to fit a NURBS because it provides local control of the shape, and its variation diminishing property. The tessellation of the approximated parametric curve becomes the centreline of the new closed fracture. Finally, we extract the non-convex hull of this polyline and generate an approximate polygonal representation of the closed crack. This approach allows the fracture to re-activate at any posterior iteration, allowing to simulate growth of a population of fractures where some are active, some have never been activated, and some are closed.

The main disadvantage of this approach is that it does not capture partial fracture closure. It assumes that if the fracture shape becomes invalid due to boundary overlap, it is a sign of closure, and therefore closes it entirely. This shortcoming can be overcome by defining stiff elements inside the fractures that do not allow for full closure. However, this approach requires a costly internal mesh. An alternative method splits the fracture BREP into several independent sub-regions allowing for some of them to remain open while others close. Figure 3.16 illustrates how the five-step closure algorithm works.

3.4.8 Algorithm implementation

All methods presented in this work have been implemented using the C++ programming language. They have been integrated as the mechanics module of the Complex Systems Modelling Platform (CSMP++), an object-oriented finite-element based library for multi-physics modelling developed at Imperial College and the ETH Zurich [Matthai et al., 2001]. A detailed reference guide to the implementation is presented in Appendix C.

3.5 Adaptive remeshing

In order to cope with the emerging geometry we implement a range of mesh housekeeping techniques. This section discusses these solutions.
Figure 3.16: Fracture closure. Handling of fracture wall overlapping due to excessive compression: (a) fracture walls before deformation; (b) overlapping fracture walls after deformation and interpolation of NURBS between fracture nodes to estimate fracture centreline; (c) NURBS tessellation; (d) extraction of non-convex fracture hull from tessellation, reformation of fracture polygon with minimal aperture.

### 3.5.1 Element quality-based mesh adaptivity

As propagation advances, fractures change the boundaries of the matrix domain, and the mesh adapts to the emerging geometry. The produced high quality, geometry conforming mesh has a minimal amount of nodes and elements. The quality of an element within a mesh can be formulated in terms of the suitability of its shape and size to represent a specific scalar field [Pain et al., 2001]. For a given scalar field an \textit{a posteriori} error can be approximated as

$$
\tilde{\varepsilon}^2 = \gamma h^T |H| h, 
$$

(3.36)
where $\varepsilon$ is the error estimate associated to a triangle, $\gamma$ is a scale factor, and

$$H = \nabla N^T \hat{\Psi} \nabla N$$

(3.37)

$H$ is the characteristic Hessian matrix of the eigenvectors of the squared interpolation function derivative matrix multiplied with the value $\hat{\Psi}$ of the FEM solution at that triangle. Each row of $|H|$ is an eigenvector, scaled by its corresponding eigenvalue. The vector $h$ contains the length of the element in the direction of each eigenvector [Pain et al., 2001; Paluszny et al., 2007] (see Figure 2.16).

This element-by-element quality score is used to control adaptive mesh refinement [Pain et al., 2001; Lipnikov and Vassilevski, 2004]. The error associated to the mean strain and pressure fields of a fractured specimen is systematically higher around the tips of the fractures where flow and deformation concentrates. Thus, large elements are ill-suited to populate the vicinity of discretised fracture tips.

We formulate a geometry-based \textit{a priori} refinement criterion which relies on automatic mesh tip tracking to generate a mesh refined around the tips and coarse elsewhere. At each step, the algorithm adapts an initial coarse mesh by constraining the size of elements with nodes at any fracture tip. These have a maximum area so that their edge length is equal to the material’s characteristic length. Elements with nodes at a fracture tip are identified and tagged for refinement. Then, a second meshing event produces a mesh that is further refined at the target elements.

After every remeshing procedure, variables are mapped between old and new meshes using the shape function projection transfer technique [Patzak and Jirasek, 2004]. Boroomand and Zienkiewicz [1999] give details on alternative methods.

The inside of the fractures is not meshed because we assume fractures to be opening mode only with no traction, nor cohesion between their walls. Therefore, refinement around the tips does not significantly influence the number of nodes of the final mesh. However, node density increases naturally as cracks propagate because fractures need a larger amount of nodes to represent their growing shape. For example, for a fracture set with 100 initial flaws the entire mesh has initially $\sim$40k nodes and $\sim$76k triangles. For the same set of flaws, grown over 110 iterations, the mesh has $\sim$370k nodes and 673k elements, including mid-side nodes. Section 3.7.4 discusses this in detail.

Theoretically, meshes can include a variety of element sizes that range from very small-sized triangles at areas of stark refinement and large elements at less busy areas. However, numerical restrictions imposed by size of the random access memory (RAM max $\sim$4GB for a 32-bit machine) of the computer and limitations introduced
by the meshing algorithms only allow for a restricted range of element sizes to be created. Particularly, meshing becomes unstable once the size of the minimum triangle approaches the floating-point precision. Therefore, as a part of the meshing strategy, meshes are generated at a larger scale, and then scaled back down to normal size for the mechanical simulation. This approach has proved to be very stable.

### 3.5.2 Fracture-to-mesh and mesh-to-fracture

Additional to their discrete representation within the mesh, fractures are kept track of as independent entities. Geometry is synchronised every time any of the two changes. We identify two cases: first, when the mesh is deformed the discrete fracture representation changes and, thus, must be updated in the fracture object. Second, when the fracture grows and its polygonal representation is extended, the mesh must be updated. The first is a fracture-to-mesh mapping while the second is a mesh-to-fracture mapping. Figure 3.17 illustrates the two parallel representations of the fractures held during the simulation.

![Fracture-to-mesh and mesh-to-fracture](image)

Figure 3.17: Fracture-to-mesh. Every time the fracture geometry changes, the mesh adapts to fit its new shape. (a) Illustrates the geometric representation of a fracture. (b) Is the equivalent mesh. Notice that the mesh has more nodes on the fracture than the original shape. Extra nodes close to the tip are inserted to induce refinement. Nodes in the centre walls are automatically inserted by the mesher.

**Fracture-to-mesh** mapping is an automatic procedure for the meshing of a domain constrained by watertight boundaries and a set of internal holes. In this case, each cavity corresponds to a fracture. We generate these triangular meshes using a C++ robust Delauney-based triangulator [Shewchuk, 1996, 2002]. Using this library, we automatically generate high quality two-dimensional meshes by defining the outer boundaries and geometric domains using Planar Straight
Line Graphs (PSLG). Each fracture is internally stored using a BREP of its geometry. Thus, in 2D, the domain is defined by a set of segments, which capture the discontinuity in the rock. These shapes are directly used as the PSLG that defines meshing. Each domain is marked by a seed, also known as material point. Thus, we must define one seed point per material (in this case one material for the matrix), and one seed per cavity. This requires the automatic identification of a point inside each fracture at each iteration step, $t_i$. Furthermore, in order to constrain element sizes around the tip we prescribe two points, $t_{n-1}$ and $t_{n+1}$, around each $t_n$, that subdivide the adjacent polygonal segments at a distance of $h_t$ from the tip, where $l_{min} << h_t < a_0$. These prescribed points introduce an $h$-refinement constraint around the fracture tip. Size of elements away from the tips is unconstrained. All elements are forced to have internal angles of at least 30°. Figure 3.18 shows an example of mesh refinement around the tip. Figure 3.19 shows the detail of a mesh generated around a tip.

Mesh-to-fracture mapping updates fracture shapes during the simulation by extracting the fracture polygon from the mesh in three steps. First, the mesher is given a marker for each fracture segment with a corresponding fracture number. Second, nodes are grouped by tag, which identifies the nodes of the fracture but still not the polygon. Third, the nodes are ordered based on mesh connectivity to yield the final fracture polygon. Once the polygon is extracted,
Figure 3.19: Detail of the mesh at a fracture tip. The tip node $t_n$ is surrounded by a rosetta of isoparametric quarter point elements. $t_{n-1}$ and $t_{n+1}$ lie on the fracture boundary, they are prescribed to the mesher and constrain minimum element sizes around the tip. Minimum element size at the tip is defined by $h_t$. $t_i$ is an internal point of the fracture which is used as a seed for the fracture domain.

the corresponding node tags are stored. Subsequently, after each deformation step, new node coordinates can be efficiently reloaded. Figure 3.20 illustrates the fracture shape extraction process performed on the mesh.

Fracture-to-mesh and mesh-to-fracture mappings are necessary in order to preserve the shape of fracture BREPs during full remeshing. This synchronisation ensures that the fracture geometric objects that live outside the mesh capture the simulated deformation and allow for a lightweight tracking of the fracture set geometry during growth. Other algorithms do not separately store these shapes in fracture objects. Instead, they perform all computations on the entire mesh. This increases the overall complexity of the system. Thus, methods such as the selectively remeshing of areas surrounding the tips and oversimplification of fracture shapes are common practice. In Section 3.7.4 we show that this process only introduces an efficiency overhead of $\sim 0.24\%$. 

78
3.5.3 Mapping between meshes

After remeshing, stresses from the original “old” mesh must be mapped onto the consecutive “new” mesh. Only by mapping these stresses, we take into account the current stress state of the object at each iteration. Figure 3.21 is an example stress field around a set of fractures. Figure 3.22 shows an old mesh (back) and a new mesh (front), to which stress tensors are mapped. Taking into account that stress is a tensorial property defined at each integration point, mapping consists of the following steps:

1. extrapolate stress tensors to nodes;
2. for each interpolation point of the new mesh, find the element in the old mesh that contains it;
3. interpolate the value of the stress tensor onto the new mesh.

Mapping is straightforward when the mesh is being refined because the new integration point represents a constrained area of the old mesh. However, when
Figure 3.21: Stress tensor field around fractures. The principal components of the stress tensor are represented by two axes, representing $\sigma_3$ and $\sigma_1$. Elements are coloured by the average mean stress: red is compressive and blue is tensile.

Figure 3.22: Mesh mapping. Shows two juxtaposed meshes: an old mesh of the previous step is colour-coded with the element mean stress field. The new mesh, to which properties are mapped, is in black. Mesh refinement is exaggerated for visualisation purposes.

The mesh is being locally coarsened, the mapping is done by proximity and not by area approximation (no rigorous upscaling of the stresses is done). This means, that the new integration point in the new mesh will be influenced by the sole element where it was contained in the old mesh, and not by its vicinity. A more rigorous approach would account for stress gradients to better approximate the variations of
the field. The error introduced by the mapping is $\sim 3\%$ (relative error), and the error introduced by the iterative process is $\sim 10^{-7}\%$ (relative error).

3.6 Fracture characterisation

Discrete fracture datasets are characterised using various concepts. This section describes the techniques used to study spatial organisation of the generated fractured rock analogues.

**Density** The spatial density ($m^{-1}$) of a 2D fracture set as originally defined by Underwood [1970] is

$$d = \frac{1}{A} \sum_{i=1}^{n} \left( \frac{l_i}{2} \right)^2$$

where $n$ is the amount of fractures in the set, $A$ is the flaw area, and $l_i$ is the length of fracture $i$. We measure how density increases throughout growth and later use it to analyze how it relates to other properties.

**Spacing** The area method defines spacing (m) as [Wu and Pollard, 1995]

$$s = \frac{A}{l_0 + \sum_{i=1}^{n} l_i}$$

where $l_0$ is the height of the specimen. This method yields a good estimation of the saturation of poorly- and well-developed fracture sets.

**Connectivity** The connectivity of a developing fracture set can be defined in relation to the initial fracture set as

$$c = 1 - \frac{n_i}{n_0}$$

where $n_0$ is the initial number of fractures, and $n_i$ is the number of clusters (intersected fractures) at the $i$th growth step. Initially $n_i = n_0$, and $c = 0$. When fractures are all connected to each other, they are part of one large cluster, and $c = 1$. Connectivity is, for growing datasets, a measure of the fracture development stage.
**Length** Two kinds of lengths are measured: the fracture length and the cluster length. The first is the length of a single fracture, even if it belongs to a larger cluster. The second is the sum of the lengths of the fractures that belong to a specific cluster. We generate histograms of fracture set length distributions to study the distribution of fracture sizes as a function of density. In the two-dimensional case fracture length is proportional to the fracture surface area.

**Extension** It refers to the length of a fracture or cluster in the $x$ direction. For straight cracks, the extension is equivalent to the length. For curved cracks, extension is a better approximation of the geometric footprint of the crack.

**Aperture** Is an emergent property of the discrete fracture model. Measured on an element-by-element basis, multiple apertures correspond to a single fracture. They are stress-dependent features; therefore, they are not stored, but computed as the closest distance between two fracture walls at any point along its centreline.

### 3.7 Testing and benchmarks

Numerical models, such as the fracture propagation model presented here, are tested in order to assess they accuracy and validity. Three main processes can be identified: validation, verification and confirmation [Thacker et al., 2004; Oreskes et al., 1994]. Validation refers to the process of determining if a numerical model correctly represents a conceptual model that captures some behaviour of the real world. Verification refers to the process of determining if the implementation of the numerical model accurately captures the model’s conceptual description. Confirmation is the process whereby we determine if there is agreement between observation and prediction.

In earth sciences, this process is particularly complex, as the boundary conditions at the field, local material heterogeneities, and other perturbations cannot be readily quantified and integrated into numerical models [Oreskes et al., 1994]. However, individual system components can be tested in order to partially validate and verify the model.

This section discusses various study cases used to validate and verify the fracture propagation algorithm. It presents a sensibility analysis of the effect of the initial random flaw distribution and analyzes the overall performance their implementation.
3.7.1 Validation of the elasticity kernel

Elasticity is a path independent constitutive law. Thus, if a body is subjected to a deformation $\rho$, the deformation is equivalent to the one attained by applying $n \times \rho/n$. In order to achieve this equivalence we must numerically accumulate the stress state between an iteration and the next. In order to test this, we compare the displacement fields of two equal experimental setups that apply a tensile deformation on a squared dataset over one and ten iterations respectively. A square dataset of $1m \times 1m$ is deformed in two different setups. For both cases, the bottom boundary is fixed. The first experiment deforms the model by applying a single top tension displacement of $10^{-3}m$. The second deformation applies 100 consecutive iterations of $10^{-5}m$ each. At the end, we compare the displacement fields. The final displacement field values range between $4.97 \times 10^{-18}$ and $1.01 \times 10^{-03}$. The stress values range between 0.0 and 5.04Pa. We measure an absolute error of $9.98 \times 10^{-7}m$ and a relative error of $1.98 \times 10^{-21}m$ in the displacement field. For the final stress field we measure an absolute error of $9.90 \times 10^{-4}Pa$ and a relative error of $1.96 \times 10^{-18}Pa$.

3.7.2 Validation and verification of the crack propagation kernel

The sub-critical failure criterion has been confirmed by multiple authors [e.g. Anderson and Grew, 1977; Atkinson, 1984; Kirby, 1984; Segall, 1984b]. The propagation criterion was validated by Renshaw and Pollard [1994b] against an acrylic coated brittle specimen. Additionally, the statistics emerging from the produced patterns were confirmed against multiple field observations [Olson and Pollard, 1989, 1991; Olson, 1993; Olson et al., 2001b; Olson, 2004]. The propagation angle was validated against experimental data early on by Cotterell and Rice [1980]. The stress intensity factor computations were validated against analytical formulas by multiple authors [e.g. Banks-Sills and Sherman, 1986; Lim et al., 1992; Alshoaibi and Ariffin, 2006].

It remains to verify that our implementation accurately represents the crack propagation conceptual model. In order to test the stress distribution along the nodes of a crack, we plot the maximum principal stresses at each node. Figure 3.23 illustrates the exponential increase of the stress toward the tip. In another test, stress intensity factors were found to agree up to 4.3% with the analytical methods for large elements around the tips, and up to 1.3% accurate for refined tips.
Figure 3.23: Maximum principal stress at the nodes. Stresses increase exponentially toward the crack tips. In this case, all corner nodes are equidistant.

The overall propagation methodology was verified by comparing our fracture propagation results to available experimental data. Figure 3.24 illustrates this comparison. Thomas and Pollard [1993] experiments compare crack paths resulting from laboratory experiments to their BEM forward simulations. They obtain a very good approximation of the crack interaction. Superposed, we observe the results of numerical crack propagation experiments using our approach.

In this experiment, two cracks of 6 cm each are initially separated by 12 cm × 6 cm. Thus, their initial interaction is minimal. During the simulation, only the lower crack propagates. The study evaluates the effect of three different boundary conditions on the final crack path: (a) all-around tension, (b) uni-axial tension, and (c) crack-parallel compression. In (a), crack interaction is strongest, causing the crack path to curve and eventually to coalesce against the stagnant fracture. In (b) and (c) the boundary conditions do not favour interaction, and the final crack path is quasi-linear. For the numerical experiments, results indicate that curvature responds to the remote stresses, and are in good agreement with the experimental data. The location of crack intersection follows the experiment closely. These numerical results partially confirm that our model reproduces experimentally generated crack paths.

In this section, we have verified that the stress intensity factors and propagation criteria agree with expected values. For more than two fractures, tests become more complex. We focus on quantifying the influence of the initial random flaw distribution on the characteristics of the final pattern. In the next section, we
3.7.3 Benchmarking eight datasets

In order to test the reproducibility of the patterns we generate eight datasets of $4\,\text{m} \times 1\,\text{m}$ with 150 flaws each. Each dataset is initially a different instance of the random flaw distribution. The material properties are elastic modulus of 20 GPa, Poisson’s ratio of 0.2, and $K_{IC}$ of 15 GPa m$^{1/2}$. The flaws have an initial size of 0.003 m and a minimum spacing of 0.012 m. These are subjected to large differential stress boundary conditions by applying tension in the $y$ direction and compression in the $x$ direction. Boundary conditions are enforced as uniform displacements of $10^{-3}$ m at the respective boundaries.

We perform between 90 and 110 iterations and systematically measure the aperture, length, density, and connectivity attributes of the datasets. Figures 3.25 and 3.26 show the fracture patterns for the eight datasets on the 50th iteration. At
this point, the density has reached an average of $0.35 \text{m}^{-1}$ for all datasets. This visual comparison illustrates the similarity of the structures that form within the datasets.

Initially, flaws grow following straight paths. Once they become larger, they start to interact with proximal fractures by influencing the stress distributions around their tips. By the 50th iteration the pattern is not yet well developed. Arrays have started to form in agreement with field observations [Olson and Pollard, 1991]. In some cases, fractures in these arrays have coalesced to form larger, more influential structures.

Figure 3.27 shows the density and connectivity trends. All datasets exhibit similar quasi-linear trends as a function of iteration and density respectively. As the datasets become more developed, measurements start to have more variation. By the 100th iteration, datasets reach a density of $\sim 0.8 \text{m}^{-1}$. Connectivity does not increase smoothly as a function of density. Instead, it increases in a steep step-wise manner. For various consecutive iterations, the connectivity remains stagnant, and increases steeply in an event-like manner. Connectivity values can soar significantly without triggering increase in density. However, throughout growth, connectivity and density exhibit an overall linear relationship.

Length distributions are also in agreement. Figure 3.28 (a) is a histogram of the fracture lengths at the 50th iteration for all datasets. As stress halos overlap and the initial Gaussian flaw distribution is shifted toward a log-normal distribution. The long leading edge represents a minority of fractures that have significantly extended their length by coalescing with others. We observe for all eight datasets, at the 50th iteration, fracture lengths follow an overall lognormal distribution.

Spacing of a fracture set is approximated with one number by using the Wu and Pollard [1995] area method. Usually the area method is applied to study the relationship between bedding thickness and fracture saturation. It is superior to the scan-line method because it produces one quantity per fractured specimen, and it can be applied to poorly and well developed fracture sets [Wu and Pollard, 1995]. We measure spacing in order to evaluate the saturation of the eight different benchmark datasets. Figure 3.28 (b) plots the spacing of each dataset as a function of density. We observe an exponential decrease in all cases.

For all datasets we obtain similar spacing, connectivity, density, and length distributions. The patterns that form exhibit the formation of fracture arrays, intersection of fractures, and formation of clusters. After the 50th iteration, the number of nodes and memory consumption per dataset raise to $\sim 480 \text{k}$ and 50GB respectively. These datasets were generated using the high-performance SGI Altix 650 shared memory system, which can run jobs that require up to 128GB of memory.
Figure 3.25: Eight geomechanically generated fracture datasets (Part I). Snapshot of the fracture shapes at the 50th iteration of growth. Initially, they grow as straight fractures. Once they become proximal, arrays begin to form. Array fractures tend to link together and form larger structures.
Figure 3.26: Eight geomechanically generated fracture datasets (Part II). Snapshot of the fracture shapes at the 50th iteration of growth.
Figure 3.27: Density and connectivity of eight datasets. Density increases linearly as the propagation advances. Similarly, all datasets exhibit a similar, quasi-linear trend in connectivity.

Figure 3.28: Spacing and fracture lengths. The left plot is a histogram of the length distribution of the eight benchmark datasets at the 50th iteration. In all cases, lengths follow an approximate lognormal distribution. The long-leading edge represents few fractures that have coalesced and significantly increased their length. The right sub-figure plots spacing as a function of density. Spacing is an average value measured at each iteration using the area method [Wu and Pollard, 1995]. It exhibits exponential decay for all datasets as density increases.

In the next section, we describe with more detail the performance of the algorithm implementation.

3.7.4 Performance analysis

The performance scaling of the crack propagation algorithm is important because it refers to the increase in resource consumption, such as memory and CPU time, as the model size increases. Ideally, scaling should be linear or sub-linear. However,
algorithms involving searching and comparing are often of exponential or sub-exponential complexity order. In this section, we analyze how our algorithm scales as a function of fracture and node density. All performance measurements were conducted on an Intel Dell Precision M65 Dual Core each 2.16GHz 2GB RAM.

**Initial flaws to nodes** The mesh that discretises a set of initial flaws is a lower bound to the total number of nodes required to simulate their growth. As a first step, we examine how the amount of nodes scales with the amount of initial flaws. Results indicate that the number of nodes and time required for the initial propagation step increment linearly and quadratically, respectively, as a function of the initial number of flaws (see Figure 3.29). The amount of nodes is the minimum required to propagate the fractures. As fractures grow, geometry becomes more complex and induces more refined areas that significantly increase the amount of nodes. These not only determine how fast the FEM linear equation system will be solved, but also influence the amount of memory required for computation. Measurements are performed on a 1m × 4m model with initial mean flaw sizes of 0.003m and standard deviation of 0.04m². For 100 flaws, we generate an initial mesh with 17k nodes which requires 0.049s to compute the first propagation step. In contrast, for 5000 initial flaws, we produce 613k nodes and spend almost a minute during the first propagation step. Thus, although the scaling is linear, simulations with >5000 flaws are very time consuming.

![Figure 3.29: Initial flaws: nodes versus time. The number of nodes required to discretise the initial set of flaws varies quadratically with the amount of flaws, while the time to compute one propagation step increments linearly as a function of flaws.](image-url)
Efficiency The algorithm is subdivided into the following main steps: building the objects, mapping data between fracture and mesh, mapping the stresses, propagating the fracture objects, and solving the displacements and stresses.

Building the objects, including the generation of the mesh (fracture-to-mesh) and the creation of the internal model data-structures (e.g. SuperGroup, explained in more detail in Appendix 1) consumes an average of 7.3% of the total time. Synchronising fractures with the mesh is the least time consuming, with an average of 0.24% of the total step time. Mapping the stresses consumes 2.98% of the total time. The propagation, including the computation of the stress intensity factors, automatic tip-tracking, merging, intersection, and extension of the fracture, requires an average 13.5% of the total iteration time. Finally, computing the displacement, stresses and strains, requires 75.95% of the total time. Thus, it represents the main bottleneck of the performance. In detail, the time for accumulation of algebraic system of equations is negligible, the time to compute solve these equations using the SAMG solver is $\sim 70\%$, and the time to compute the stresses and strains is approximately $3.2\%$ of the total time.

The greatest performance hit is given by the solution of the algebraic system of equations implemented using the SAMG solver [Stüben, 2001]. The SAMG is a system algebraic multigrid solver highly optimised for ground water flow and oil reservoir simulations. It is a scalable solver with complexity in the order of $O(\log(\text{nodes}))$. It supersedes other solvers by a factor of up to ten times in terms of speed [Stüben et al., 2003], and it is available for distributed systems. Thus, by running the same experiments in a faster serial machine, or in a distributed cluster, the amount of time required to solve the system can be reduced as a function of increase in resources.

3.8 Fracture growth applications

The fracture propagation module can be used to generate a number of fracture patterns. In this section, we review some examples of applicability of the dataset. For all datasets, we assume elasticity modulus of 20 GPa and Poisson’s ratio of 0.2.

3.8.1 Tension: 10 fractures

We setup a model of $1\text{m} \times 0.5\text{m}$ with ten initial flaws of mean size 0.01m and standard deviation of 0.07m$^2$. We fix the lower boundary and apply an extensional displacement boundary condition at the top of $10^{-5}\text{m}$. We apply 50 deformation steps. The final
patterns can be examined in Figure 3.30. We examine the geometry and stress states at iterations 1, 20, 30, and 48. During the first 20 iterations, growth is focused only in three main fractures. Others exhibit minimal growth. By the 30th iteration four of the initial flaws have become full sized fractures. We observe that fractures initially propagate on straight paths. As they become larger, they inhibit the growth of the smaller surrounding cracks. The two proximal cracks at the lower right are the first to interact. Their tips hook as they approximate each other. The left side of the figure shows the adaptive mesh that forms during growth. The refinement around fractures is a result of the pre-defined fracture resolution set by the nodes of the fracture BREP. On the right, the images show the mean stress distribution in the dataset. The cracks accumulate compressive stresses around their tips and cast tensile stress halos around them. Intersecting cracks yield a single stress halo. Tension is accumulated only at the out-most tips. Proximal cracks cast a single, larger stress shadow around them.

3.8.2 Tension: 200 fractures

The dimensions of the model are 2m × 0.5m. It has 200 initial flaws of mean length 0.01m and standard deviation of 0.07m². These have an initial minimum spacing of 0.03m. We fix the model at the bottom and extend it at the top by 10⁻⁵m. Figure 3.31 shows (a) the displacement field on the initial flaws, and (b) on the propagated fracture set. Once fractures grow, they affect the displacement field by shielding deformation. Aligned fractures cast a single stress shadow. Figure 3.32 shows the stress intensity factors, $K$, measured at the crack tips. Stress intensity values are higher for larger fractures. Smaller, inhibited cracks accumulate less stress. The larger values at the tips indicate that fractures, although unconnected, express a combined stress halo.

Figure 3.33 shows the detail of the stress field around the tips of interacting fractures. In (a) fractures exhibit interaction by hooking at the tips. Stresses at the extreme of the array are higher than around internal tips. (b) Proximal fractures act as a unit and cast a combined, larger stress shadow around them. Stresses are stored as tensorial properties. For visualisation, the average of the principal stresses, a scalar value, is projected onto the mesh.

3.8.3 Effect of local material heterogeneities

To measure the effect of the matrix heterogeneities of the material on the fracture paths and patterns, we re-generate the crack patterns for increasing perturbed
Figure 3.30: Growing 10 fractures. Model at iterations 1, 20, 30, and 48. On the left, the mesh at the different growth stages adapts to the new geometry. On the right, the mean stress field that drives propagation. Fractures grow initially in a straight path. The growth of most of the flaws is inhibited by their larger neighbours. Once these get closer, their fields interact and for more complex curved shapes.

As opposed to previous experiments, the matrix has a variable Young’s modulus, $E$. We perturb $E$ by 10% and 50% and examine the effects on fracture shape tortuosity and overall pattern characteristics. Figure 3.34 illustrates the variations in the elasticity moduli. As expected, for 10% variation the perturbation is more homogeneous as in the case where $E$ varies by 50%. For 10% variation the elastic modulus ranges from 18GPa up to 20GPa. For 50% perturbation, $E$ ranges from 10GPa up to 20GPa. To make the heterogeneity effect stronger, $\nu$ and $K_{IC}$ are varied proportionally to $E$. The paths obtained are compared against the homogeneous case. Since the three propagate from the same initial set of flaws, we can compare fractures one-to-one.
Figure 3.31: Growing 200 fractures. (a) Displacement field caused by a uniform extensional deformation at the top of the model. Arrows are coloured by the length of the vector. (b) Shows the displacement field after growth. Notice that fractures induce heterogeneities in the deformation. (c) Mean stress contours are over the mean stress scalar field of the same dataset. Contours mark regions of the model with equal mean stress values. A higher concentration of contours indicates a faster change of the stress values.

Experiments from the civil engineering literature indicate that crack paths are independent of material properties for a specific geometry and set of boundary conditions [e.g. Arrea and Ingraffea, 1982]. Moreover, they show that for brittle porous media, variations may occur due to grain size differences as coarser materials usually produce more tortuous paths than their counterpart finer versions. Figure 3.35 shows the detailed paths of three interacting fractures. Paths become more tortuous as perturbation increases. Interaction between the cracks agrees in the three cases.

Figure 3.36 shows the final patterns obtained for the three cases. These exhibit
Figure 3.32: Stress intensity factors at fracture tips. Glyphs represent $K$. Refinement around the cracks is structured. $K$ is larger at the fracture array tips. Isolated fractures measure significantly lower $K$.

Figure 3.33: Detail of fracture interaction. (a) Fracture array casts combined stress halo. (b) Larger fractures inhibit the growth of neighbouring smaller fractures. The stresses at the array tips are higher.

strong similarities. Although local heterogeneities in the material cause curvature differences, these do not affect the final crack paths. Results agree with Arrea and Ingraffea [1982] findings. For various mixed loading experiments of concrete and mortar beams of various grain sizes, they observe that resulting cracks have comparable paths.

In nature, these local heterogeneities may induce stronger effects on crack patterns due to the stark contrast between the material properties of the different components of the matrix. Material-dependent relationships between toughness and elasticity
modulus, disregarded here, may influence the overall effect of these heterogeneities on the final fracture paths and patterns. This experiment demonstrates the ability of the FEM-based methodology to handle complex material-property distributions that affect the formation of fracture patterns. In contrast, BEM-based models do not define properties on an element-by-element basis, and therefore, are better suited to model homogeneous media.

### 3.8.4 Effect of initial random flaws: orientations and spacing

The initial distribution of the flaws plays an important role in determining the final pattern mainly because it determines where fractures can start to propagate and
where not. This has a quantitative impact on the final pattern. Large areas without flaws act as stronger areas of the model which exhibit less fracturing. This is a primarily quantitative effect. The orientation of the flaws does not affect the overall qualitative characteristics of the final pattern.

Figure 3.37 shows the patterns generated with a set of 100 initial flaws with different initial spacing. All flaws have a mean size of 0.01m and a standard deviation of 0.07m$^2$. In (a), the maximum distance between any node of any flaw with any other flaw node is the size of the crack. In (b), this minimum distance is $3\times$ the size, and in (c) $10\times$. The spacing of the cracks induces density variations in the initial flaw pattern. For low spacing values a more evenly distributed flaw set is generated, while for larger spacing, small clusters form larger void areas. Results indicate that crack path curvature is affected by spacing. For the clustered datasets (a) and (b) we observe higher amounts of crack arrays which, during growth, interact and eventually coalesce. In (c), fractures are evenly spaced and grow in overall straighter paths. In (a) and (b) more fractures exhibit curved paths as compared to (a).

Figure 3.38 shows the initial flaw distribution and fracture pattern resulting from propagation: (a) a set of randomly oriented flaws, and (b) a set of initially parallel
Figure 3.36: Effect of local material heterogeneities. Patterns result from growing the same initial set of cracks with perturbations in the Young’s moduli of each integration point of: (a) no perturbation, (b) 10%, and (c) 50%.

cracks. In both cases, the set has initially 100 flaws. Both datasets were propagated during 14 iterations of growth. The resulting density is lower for (a) as compared to (b). Only 50% of the flaws grow. In (b), most of the cracks exhibit some growth. In both cases, the majority of the fractures have straight paths. When examined closely, fracture shapes in (a) exhibit a small kink, which preserves the original orientation.
Figure 3.37: Effect of spacing. Patterns for initial flaw spacing of: (a) 1×, (b) 3×, and (c) 10× the maximum flaw size. Smaller spacing induces flaw clustering and promotes increased connectivity. Larger spacing produced evenly distributed flaws and results in straighter final paths.

of their seed. We also observe, that in (a), those flaws that were favourably aligned with remote stresses grew, while others remained at their original size. Figure 3.39 shows these crack paths in detail.

For the flaws with random orientations, the resulting pattern is less populated
Figure 3.38: Effect of flaw random orientations. (a) and (c) are the initial flaw distributions for randomly and horizontally oriented flaws respectively. (b) and (d) are the developed patterns. Vertically oriented flaws in (b) do not propagate, thus, the final pattern exhibits less density. In (c) all flaws propagate. The overall fracture orientation in both datasets is horizontal.

because many of the fractures are not aligned with the remote stresses. To grow more fractures we place the majority of seeds at an orientation aligned with the principal deformation stresses. In some cases, such as pure tension, we place all cracks oriented perpendicular to the maximum principal stress. In other cases, where the principal stress is not defined beforehand it is best to define randomly oriented flaws. The next section describes the case in which we shrink a model by applying a volumetric strain. In this case, the stress field is highly variable and fractures may open in all directions.

3.8.5 Shrinkage

Shrinkage experiments mimic phenomena of loss of volume, such as drying of mud, cement, and other materials. This process can be simulated by applying isotropic shrinkage to a model by specifying an initial volumetric strain. This is set to a value lesser than one. Simultaneously, displacement at the boundaries is restricted. Thus,
the material at the boundaries is dragged toward the centre and torn away from the boundary. If the material has flaws (as in our models), these concentrate the stresses generated during this process and propagate fractures. This process creates a unique overlap of stress halos that induce the formation of enclosed polygonal regions. Due to the continuity restrictions of our model, the fracture sets is only propagated until the first block forms. However, well before closure, polygonal regions are identifiable.

Figure 3.40 shows a set of initial fractures that grow while exposed to shrinkage. The model is 1m × 1m. It has 20 randomly oriented flaws, initially of mean size 0.01m. We apply isotropic shrinkage of 0.1% while fixing displacement at all boundaries.

Figures 3.41 (a) and (b) show the displacement field around the propagating fractures. Initially, displacements are more pronounced around the fractures. These absorb most of the deformation energy by propagating. At a latter development stage, the displacement field is significantly more variable. Visualisation indicates that the centre block is rapidly moving to toward the upper right. The blank area corresponds to a matrix block that has become unattached from the rest. It is a region of the matrix that has become completely isolated from the rest due to the linkage of all its surrounding cracks. Tips at the boundaries do not grow beyond the propagation area. Therefore, they continue to accumulate stress. This can be observed in Figure 3.41 (c).
3.9 Extensions of the method

The presented method can be extended to include a range of physical phenomena as dynamic fracturing, non-linear material properties, multiple domain simulation, and three dimensional domains. This section briefly discusses the previous.

3.9.1 Including dynamic effects

Dynamic systems, as opposed to static crack systems, acquire kinetic energy due to the unbalanced forces they are subjected to. Under these conditions, the static energy release rate conditions discussed earlier no longer apply as they disregard inertial forces of the system [cf. Lawn, 1993]. In the case where stable cracks grow slowly, kinetic energy is insignificant as compared to the mechanical energy of the system. Those quasi-static systems can be described by the well-established Griffith-Irwin static solutions. In contrast, the numerical simulation of a dynamic system requires the model to incorporate notions of kinetic energy, and therefore time. However, these only affect failure criteria, energy balance, and speed computations of growing cracks while geometric handling challenges remain the same. Shahani and Fasakhodi [2009] describe a similar geometric generation technique for dynamic fracture propagation which relies on the continuous remeshing of the domain and computes dynamic parameters of the system to control the growth of a discrete crack. In order to extend our method to handle dynamic fracturing, it would suffice to adopt a dynamic failure and propagation criterion, e.g. dynamic stress intensity factor [Kanninen and Papelar, 1985], while preserving the same geometric handling described in this chapter.
Figure 3.41: Shrinkage cracks: displacements and stresses. The displacement field is visualised using thick arrows for the (a) first, and (b) last iteration. The empty area corresponds to a block that is detached from the rest of the matrix. Stress contours (c) and (d) illustrate the areas of the model with equal mean stresses. Higher density of contours indicates strong changes in the field. These are measured at the tips of the cracks around the boundaries.

### 3.9.2 Non-linear material constitutive laws

The process zone is a plastic region ahead of a fracture tip in which stresses and irreversible strains concentrate. In general, materials where the process zone size is comparable to the cross sectional size of the studied body are considered to be quasibrittle [Bazant, 2004]. Rocks and concrete are two obvious examples of quasibrittle materials. Fracture models discussed within this report indirectly model this zone and capture the degrading behavior that precedes fracture formation.

This region can be modeled with more detail by defining a plastic constitutive relationship which takes into account local irreversible deformation. Non-linear
constitutive laws require the implementation of a costly iterative solver for the yielding non-linear equation system. This results orders of magnitude more computationally expensive than the linear elastic alternative, and its stability decreases proportionally to the increase of nodes. Kojic and Bathe [2004] describe how these methods are implemented. Furthermore, plastic strain accumulation leads to local loss of material integrity. Damaged material becomes weaker and eventually fails. In such case, it is necessary to formulate a transition mechanism for damage from the sub-grid level to the discrete fracture level, modelling crack nucleation.

3.9.3 Crack propagation in 3D

Fractures in nature develop in 3D rock masses. Ideally, fracture propagation algorithms should aim to represent these in their true three dimensions. However, the complexities related to the geometric handling of their evolving shapes have pointed research from the formulation of a sound propagation algorithm thus far. Two-dimensional models are developed as a starting point for the study of fracture propagation. Earlier on, we defined the three cornerstones of a growth algorithm: a failure criterion, a propagation criterion, and propagation angle. Additionally, we identified computational-geometry hurdles that must be overcome in order to numerically simulate growth: shape housekeeping (Section 3.4.3), intersections (Section 3.4.6), and adaptive remeshing (Section 3.5). In 3D, fracture shapes are not polygons, instead, they are complex multi-faceted non-convex polyhedra. Intersections must be handled with complex 3D Boolean techniques. The fracture-sweeping algorithm for fracture-to-mesh and mesh-to-fracture mapping becomes an intricate flood-fill algorithm. Property mapping between meshing becomes more expensive, requiring the implementation of optimisation techniques to avoid a performance hit. Finally, in 3D, automatic and adaptive remeshing relies on less available cutting-edge algorithms that handle three-dimensional adaptation of geometry.

Ingraffea and Wawrzynek [2003] presented an algorithm that grows a single 3D fracture as a function of the stress at its curve tip. This process is semi-automatic, and designed for the growth of one or two fractures only. It avoids geometric handling of 3D merging and intersection. Object oriented programming allows to create code that is extensible to the third dimension. Extending the finite element discretisation of the deformation equations is straightforward. In fact, in CSMP++, we compute stress and strain distributions of 3D models due to external deformation by feeding a 3D mesh and set of boundary conditions to the deformation kernel. Thus, the main
bottleneck to extend the crack propagation framework to 3D is the lack of readily available 3D meshing modules able to handle complex domain characterisation while supporting geometric-based adaptive meshing.

3.10 Conclusions

We have presented a discrete crack model that can be used to geomechanically generate 2D fracture patterns. Fracture shapes are kept track of independently of the mesh. Thus, geometric techniques to handle propagation, intersection, and closure do not involve operations with the mesh. The mesh is adaptively refined to capture the emerging geometry and the high stress curl at the tips. We apply tension and volumetric strain boundary conditions to generate fracture patterns. This method reproduces crack paths from physical experiments. It proves as an efficient way to model discrete fracture propagation and yields self-organising, evolving set of fractures that grow as a response to a deforming boundary. The methodology allows us to create a realistic heterogeneous medium analogous to a fractured rock mass.
4 Application to layered media

Contents

4.1 Abstract ......................................................... 106
4.2 Introduction .................................................... 106
4.3 Boundary condition effect on crack paths .................. 109
4.4 Study of stress fields around fractures ...................... 112
  4.4.1 Two-dimensional plane stress versus plane strain ...... 114
  4.4.2 Three-dimensional single layer models .................. 116
  4.4.3 Three-dimensional layer bound models .................. 117
4.5 Discussion ....................................................... 120
4.6 Conclusions ..................................................... 123

4.1 Abstract

Fractures in layered media form patterns that depend on depth, layer thickness, friction with neighbouring layers, and material properties. Over the past four decades two-dimensional fracture propagation algorithms have been used to model threedimensional behaviour inherent to nature. However, sub-dimensional models do not capture stress variations in the third dimension and disregard the influence of upper and lower confining layers on deformation. We investigate the effect of boundary conditions on the curvature of fractures. Models display curvature for large, intermediate, and zero differential stresses. By comparing stress fields of unbound 2D and 3D fracture models with their bound counterparts we identify friction as an important factor in the generation of straight fracture patterns. Results indicate that two-dimensional models are suitable to model crack growth in layers where interaction does not dominate deformation.

4.2 Introduction

Layer restricted fracture growth is of great importance for the accommodation of lithology dependent strains during burial and exhumation of sedimentary sequences
(see Figure 4.1). However, available two-dimensional models neglect such strain and stress variations in the third dimension, and are, therefore, not directly applicable to layered rocks. From the numerical point of view, 2D models are lighter and faster because they require less nodes than their 3D counterparts. From the physical point of view, they have the advantage of allowing to discern effects due to in-plane self-organisation and crack interaction only [e.g. Du and Aydin, 1991], as opposed to complex 3D models that inherently blend the effects of three-dimensional crack surface interaction and non-trivial boundary conditions into the final pattern.

In the field, fracture orientations in layered restricted patterns range from straight to curved. Their curvature is important because it not only holds clues as of how these patterns were formed, but also influences the overall rock flow properties. Olson and Pollard [1989] presented a method to infer the relative magnitude of differential stresses from fracture patterns. They concluded that straight, as opposed to curved fractures, form in the presence of “large” differential stresses. Renshaw and Pollard [1994a] extended this analysis and concluded that straight paths could also originate due to anisotropic material weakness and the locking effect of fracture surface roughness. Based on this model, numerous forward crack propagation simulations were performed excluding the possibility for fractures to curve [e.g. Olson, 1993; Renshaw and Pollard, 1994b; Olson, 2004]. Later, Olson [2004] incorporated the effect of layer thickness on the evolving patterns by introducing an empirical weighting factor for crack interaction. These investigations, however, did not consider frictional coupling with potentially plastic neighbouring layers.

Often, layer deformation is triggered by the movement of neighbouring layers. Figure 4.2 illustrates an example in which a middle layer, bound by a top and bottom plastic layer is deformed due to stresses transduced through these layers. Figure 4.1 shows an example of this behaviour in nature. Depending on the material property contrast between these layers, and the slip and friction between them, peripheral deformation induces strains in the brittle layer and causing fractures to form. The characteristics of the fracture patterns that arise in the brittle layer depend on the competence contrast between the layers, the nature of the contact between the layers (frictional, slipping, consonant), and the boundary conditions of the entire system. It is of interest to investigate if the stress fields that arise under these circumstances are adequate for the formation of curving cracks, as opposed to straight cracks only.

In nature, the resulting crack path arises as a combination of physical processes. Two important factors that have been recognised to influence crack paths are boundary conditions and layer interaction. In this chapter, we study the difference
between applying small and large differential stresses on the system and on the question of when fracture stress field interaction dominates over the remote stress. By studying the qualitative properties of these stress fields we obtain insights as to how the fractures would propagate when subjected to these conditions, as propagation is known to follow a path aligned with the maximum circumferential stress at the tip [Cotterell and Rice, 1980]. Additionally, we study the effect of friction between bounding layers on the stress fields around a growing fracture tip as a dominating mechanism for straight crack growth.

This chapter is organised as follows. Section 4.3 introduces the effect of boundary conditions on fracture paths. Section 4.4 discusses the stress fields around fractures due to layering effect. Section 4.6 concludes this chapter with a summary of the applicability of the two-dimensional crack propagation method to layered media.
4.3 Boundary condition effect on crack paths

In the physical experiment presented in Section 3.7.2 [Thomas and Pollard, 1993], a fracture propagates toward another proximal, but dormant fracture. Under sufficient deformation, the fracture path curves toward or away from its neighbour. For all-around tension - also known as zero differential stress, the crack curves toward the first one until they coalesce. For uni-axial stress - also known as intermediate differential stress, a lesser interaction between the two cracks occurs. Finally, for the crack-parallel compression case promoting a large differential stress, fractures exhibit almost no interaction at all yielding a quasi-straight path. Figure 4.3 illustrates the boundary conditions for these three cases. These 2D results, generalised to the vast amount of parallel, non-interacting fractures found in the field [Segall and Pollard, 1983], have motivated scientists to regard straight cracks in layered media as having formed under large differential stress boundary conditions [Olson and Pollard, 1989].

Figures 4.4 and 4.5 show the effect of applying zero, intermediate, and large differential stresses to a dataset with 100 initial flaws. The three datasets start out from the same set of flaws with Gaussian size distribution. The final crack length variation is minimal in this model. Although they originate from the same set of flaws,
the final datasets differ significantly in terms of fracture curvature and connectivity. For the large differential stress early paths are predominantly straight lending support to the straight-only crack propagation algorithms described earlier. However, once cracks reach a critical size, where the length of the largest fracture starts to exceed their spacing, interaction triggers path curving. For intermediate and zero differential stresses, curving begins earlier.

Figure 4.6 shows the displacement field around a fracture dataset subjected to zero and large differential stresses respectively. In both, we notice a stark shielding effect of the fractures on the overall deformation induced at the boundaries. In (a), the displacement field is more perturbed toward the centre of the dataset as compared to (b). Local variations of displacement induce the formation of curved crack paths. This fracture pattern develops a significantly more complex stress field, as opposed to a two-fracture model, due to the local dominance of neighbouring cracks over the remote stress field. Thus, differential stresses do not appear to play a critical role in determining the fracture crack path. Instead, its position within the set appears to control the type of path, straight or curved, as opposed to remote stress.

These results indicate that although boundary conditions control fracture curvature in single crack experiments, for multiple-cracks this might no longer be the case. Instead, we observe curving of fractures for all applied boundary conditions, including large differential stress. This behaviour is best explained by interaction processes between heterogeneities that locally override the remote stress field. Propagation, driven by local processes around the tip, is no longer controlled by the remote stresses. Therefore, in 2D models, a high degree of differential remote stress does not completely suppress curvature.
Figure 4.4: Boundary conditions: displacement field. Displacement field generated by: (a) zero, (b) intermediate, and (c) large differential remote stresses.

Other mechanisms, such as friction processes with neighbouring stratigraphic layers, and locking due to surface roughness, which control the variability of stress and strain in the third dimension may reduce the effect of crack interaction on the resulting paths. In the next section we explore the effects of boundary conditions on the stress fields and investigate bounding layers as a possible alternative explanation to straight path growth.
4.4 Study of stress fields around fractures

Two-dimensional models provide insights into the interaction processes that arise during growth. They are capable of forming intricate patterns that reproduce the length and aperture distribution statistics measured in the field. In 2D, the deformation algorithm makes assumptions about the stress and strain fields in the
third dimension. For plane strain, strain in the $z$ direction is assumed to be zero. For plane stress, stress in the $z$ direction is assumed to be zero. In general, plane strain is applied for the modelling of thick plates while plane stress is more appropriate for the simulation of thin plates [Timoshenko and Goodier, 1934].

This section investigates the effect of layer interaction on crack paths by performing a variety of simple simulations designed to compare the stress field around two proximal fractures as given by various modelling scenarios. We study 2D plane stress and plane strain stress fields and compare them against the 3D stress field of
actual 3D plates of different thickness. We also study the shape of the stress field around a crack in a finite thickness layer bound by one or two layers.

All datasets are based on the same initial 2D fracture model as proposed originally by Thomas and Pollard [1993]. In all experiments, the specimen is fixed at the bottom. For all, we apply a tensile (pulling) displacement boundary condition of $10^{-4}$ m at the top boundary. The nodes of the mesh at the fracture walls are disconnected; thus, nodes can deform away from each other under stress. Results are expressed as effective stress contours and stress tensors. Effective stress is defined as the average of the stress tensor eigenvalues. In the illustrations, stress tensors are represented using tensor glyphs: ellipses in 2D and ellipsoids in 3D. Their axes are aligned with the eigenvectors of the original stress tensor. Similarly, their radii match the eigenvalues of the original stress tensor. For better visualisation, glyphs are coloured with the effective stress of the tensor. For all experiments, we assume a Young’s modulus of 40 MPa and Poisson’s ratio of 0.23.

4.4.1 Two-dimensional plane stress versus plane strain

For the 2D analysis of the system, we create a model with two fractures of 0.06m diameter each. These are embedded in a 0.36m × 0.56m rectangle. We mesh the dataset using the Ansys ICEM Tetra Surface mesher. The initial hybrid mesh contains a mix of triangles and quadrilaterals. The closest crack tips are at a distance of 0.11m in the $x$ direction and 0.06m in the $y$ direction. The distance between them is $\sim 0.125$m. Thus, we expect little interaction between the stress fields of the cracks. The objective of generating this mesh using the external mesher is that it can later on be extruded into well-formed hybrid 3D meshes for the rest of the experiments.

Figure 4.7 shows the effective stress contours for the: (a) plane strain, and (b) plane stress assumptions. We observe similar formed stress contours for both assumptions. However, in the plane stress case we observe that fractures cast a larger stress shadow. We also measure a higher maximum compressive and tensile mean stresses as compared to the plane strain case. For plane strain, fractures exhibit a larger degree of interaction.

Figure 4.8 shows the stress field of the 2D specimen assuming: (a) plane strain, and (b) plane stress. In both cases, fractures significantly influence the stress field around them. In the figure, ellipses appear thinner in the plane stress case. For plane stress, the displacement at the top causes a large extension in the $y$ direction, but very little deformation in the $x$ direction. Thus, the remote stresses exhibit a more
Figure 4.7: Plane strain versus plane stress: stress contours. Each line represents a region of the dataset with equal effective stress for: (a) plane strain, and (b) plane stress.

anisotropic shape than for the plane strain mode. In this case, the Poisson effect, strain perpendicular to the stress direction, is better appreciated.

Figure 4.8: Plane stress versus plane strain: stress field. Stress tensors are represented using 3D ellipsoidal glyphs. The axes of the ellipses are aligned with the eigenvectors of the tensor. The glyphs are coloured by the effective stress: blue represents extension and red compression. Results are for deformation assuming: (a) plane strain, and (b) plane stress.

We observe that plane stress and plane strain stress shadows are different. The isosurfaces display a stronger interaction between the fractures, while plotting the stress tensor field shows that, in both cases, fractures exert significant influence on the field around them. Thus, at propagation, and unless this interaction is overridden by some local large differential stress, we expect crack paths to exhibit curvature. In the next section, we explore the effects of these fractures on the stress field of a three-dimensional plate.
4.4.2 Three-dimensional single layer models

To quantify the effect of the specimen thickness on the stress field of the fractures we generate two specimens which are extrusions of the original 2D dataset produced for 4.4.1. Their thickness is $10^{-3}$m and 0.03m, respectively. Figure 4.9 shows the stress contours for the two cases.

![Stress contours for two cases](image)

(a) (b)

Figure 4.9: Stress contours around fractures embedded in a three-dimensional layer. 3D isosurfaces mark regions of the datasets of equal effective stress. Two datasets are depicted: (a) thickness $10^{-3}$m and (b) thickness 0.03m.

Figure 4.10 shows the stress tensors for the two cases. The two experiments yield similar stress contours. Therefore, it is difficult to explain an effect of the thickness on the stress field around the interacting cracks. This is different to the spacing discussion in [Bai and Pollard, 2000] which provides evidence that fracture spacing is a function of layer thickness due to the maximum size of the stress shadow that a layer bound fracture can cast. The stress fields generated are also comparable to the previous two-dimensional plane strain case in that they both exhibit strong interaction between the cracks. However, for the same boundary conditions, the maximum tensile and compressive stresses are smaller as compared to the 2D case. The important difference between the two lies in the shape of the remote differential stress away from the cracks. We observe that stress tensors away from the cracks are more anisotropic for the thin specimen as compared to the thick specimen. However, we do not observe a great qualitative difference in the shape of the stress tensors around the cracks. Results are qualitative analogous to the two-dimensional results obtained previously. In all cases, fractures cast a stress shadow that manifests itself as a heterogeneous stress field around it. Thus, cracks that propagate and approach this area will curve their paths due to the ensuing interaction.
Single layer models also do not exhibit stress fields that would cause straight crack propagation. In order to understand straight fracture patterns formed in nature we draw our attention away from the thickness of the layer toward the interaction with neighbouring layers. In the next section, we investigate how a lower bounding layer affects the studied stress field.

![Figure 4.10](image)

(a) (b)

Figure 4.10: Stress field around fractures in a three-dimensional single layer. 3D glyphs are used to visualise the stress tensor distribution. Two datasets are depicted: (a) thickness $10^{-3}$m and (b) thickness 0.03m.

### 4.4.3 Three-dimensional layer bound models

We repeat the same experiment for a thin layer with two fractures of 0.06m of diameter at the same distance as before. The specimen has a thickness of $2 \times 10^{-3}$m. We take the previous $10^{-3}$m width specimen containing two cracks and glue it to a layer without fractures. This second layer is underneath the first. The idea is to mimic the effect of a layer deforming on top of another. We expect to see, reflected in the stress field around the tips, the conditions that induce formation of parallel joint sets. Figure 4.11 shows the resulting stress contours and tensors.

The stress field in Figure 4.11 shows the strength reduced effect of the fractures on the overall stress state of the thin plate. We observe that the fractures cast a very mild influence on their proximity by inducing loss of anisotropy in the stresses around them. However, as compared to the two previous experiments, the influence of the fractures on their vicinity is much less pronounced. In fact, the fractures display almost no interaction nor effect on the surrounding stress field. In this case, stress tensors close and far from the tip are highly anisotropic. They are aligned with the far field displacement boundary condition. This example models the extreme case in
Figure 4.11: Stress field around fractures in a three-dimensional double layer. The dataset is $2 \times 10^{-3}$m thick in total. It is composed of two layers. The upper $10^{-3}$m thick layer contains two fractures, while the lower layer does not contain any fractures. In (a) the isosurfaces represent regions of the dataset of equal effective stress values. In (b) 3D glyphs represent the stress tensors at each integration point.

which the friction between the upper and lower layer is high, there is no competence contrast between the layers, and the deformation is controlled by the more flexible, less brittle lower layer. This configuration of stresses will generate straight fractures which do not interact, even if closely spaced. It is expected that for a weaker coupling between the layers we would see a more pronounced effect of the cracks on the brittle layer.

When the layer is bound by a lower and an upper layer, the region of influence of the fracture is also greatly affected by the neighbouring bodies. When the two exterior layers are pressed together and simultaneously pulled apart, they crush the middle layer generating extensional deformation. Figure 4.2 illustrates this case. Figure 4.12 shows the variation of the stress field as the a function of the increase of the bounding layer weakness. The stronger the bounding layers, the more pronounced the stiffening effect of the neighbour layers, and the more localized the effect of the fractures on the field. When the two bounding layers are considerably weaker than the middle layer, fractures cast a compressive shadow around them. When the neighbouring layers are of similar strength to the middle layer, the deformation herein is not as significant as in the previous case, and the compressive fracture halos are significantly weaker. For the sandwich layer we observe a less pronounced interaction between the fractures as compared to the unbound case presented in Section 4.4.2.

The three-dimensional layer bound experiments show that traction against peripheric bodies may induce stress states that inhibit crack interaction and therefore
Figure 4.12: Stress field around fracture in a three-dimensional sandwich layer with differential stress. 3D glyphs are used to visualise the stress tensor distribution of a middle layer bound by an upper and lower neighbour layers. Tensors are coloured with the value of the effective stress. Boundary layer elasticity modulus is (a) $1\times$, (b) $0.1\times$, and (c) $0.01\times$ the elasticity modulus of the middle layer.

promote crack growth in straighter paths. We showed that large differential stresses are dominated, not only by local fracture interaction processes, but also but structural layer deforming processes. Thus, we showed that two-dimensional numerical crack propagation is suitable for modelling layer restricted fracture propagation in which interaction with neighbouring layers does not dominate deformation.
4.5 Discussion

Previous work on multiple crack propagation disregards fracture curving. In some cases, this assumption is valid. For example, when fractures occur on a coating layer of a bending plate (see analogous experiment in Section 4.4.3). The plate controls deformation, and the maximum principal stress is governed by the bending and not by local fracture interactions. Another case is the generation of fractures in an inherently anisotropic medium that exhibits weakness in a preferred direction.

Figure 4.13 shows two simulations performed on the same initial flaw distribution: fractures grow due to tensile displacements in the $y$ direction. In (a), fractures were propagated on a straight path only. This is achieved by defining a constant propagation angle, equal to the original crack orientation, during growth. In (b), the local stress field controls fracture curving. The resulting patterns have distinct geometrical qualities. In isolation, fractures propagate in linear paths in both cases. The curved pattern exhibits a higher connectivity of the fractures and yield significantly different flow properties (see Chapter 5).

Another example that highlights the impact of modelling curvature is fracturing due to shrinkage. We apply shrinkage by defining an isotropic dilatation factor of 0.1% while fixing displacement at the boundaries. At the onset of the simulation, the model contracts and cracks propagate to accommodate loss of volume. Figure 4.14 shows two simulations performed on the same initial set of flaws: (a) shows the initial flaw distribution, and (b) and (c) show advanced stages of growth. Fractures grow as a function of stress concentration at their tips. In (b), fractures are propagated on a straight path. In (c), crack paths are updated at every iteration. Cracks approximate each other orthogonally. In (b), the simulation does not capture this emergent effect. Curving has a significative impact on the final pattern: (b) results unrealistic, while (c) captures the self-organisation of the fractures that creates internal polygonal regions.

We showed two examples for which fracture curving yields significantly different fracture patterns. With 2D fracture propagation simulations the underlying assumption is that, in the $z$ direction, the fracture extends uniformly and that either stress or strain are assumed to be zero. Thus, the applicability is restricted to layered media. However, layered media comprises a wide range of deformation scenarios involving bounding neighbour layers, material property contrast, compressive and differential stresses, among others. In this chapter, we compared the effects of boundary conditions and interaction with other layers on the ensuing stress fields.
Figure 4.13: Straight versus curving paths: tensile stress. Comparison between two datasets generated from the same initial set of flaws, subjected to tensile stress boundary conditions. In (a), fractures propagate on a straight path only. In (b), fracture propagation angle is updated at every growth step. Notice that fractures in isolation exhibit similar shapes in both cases.

around cracks in full 3D fractured layer media representations. These results can be used to predict under which conditions cracks will propagate on straight or curved paths.

Layer bound fracture patterns are often of parallel or semi-parallel nature. Friction with neighbouring layers moderates some of the differential stresses that cause fractures to curve, inducing cracks to propagate along a straighter path. Two-dimensional algorithms for crack propagation are not well suited to reproduce this kind of patterns. In numerical experiments, curving can be turned off [e.g. Olson and Pollard, 1989; Renshaw and Pollard, 1994b]. This is somewhat arbitrary, but such patterns compare well to physical experiments of bending thin coated acrylic sheets [e.g. Renshaw and Pollard, 1994b]. In this case, the layer is constrained by a lower body that controls deformation. Thus, the principal stresses, and the strain anisotropy is local to each fracture tip. The plasticity of the lower layer allows it
to bend without fracturing, while the upper layer cannot cope with the strain, and breaks forming parallel fracture sets with an overall orientation perpendicular to the remote stress. In 2D, the layer is assumed to have continuous material properties in the $z$ direction. In fact, the only assumption is that stress or strain are zero along the third dimension. Thus, such bending problems result much more difficult to reproduce.

Although curving is important and it significantly influences the final characteristics of a simulated pattern, it is also a strong indicator of the local stress conditions at the time of crack formation. Unfortunately, two-dimensional models cannot always capture, from first principle, some of the crack propagation scenarios found in nature. An approach that attempts to model this behavior weighs the influence of cracks on each other using empirical parameters [Olson, 2004]. In this case, the elastic constitutive relationships are modified by using a relationship which relies on a layer thickness, and two experimentally fitted parameters. However, only full three-dimensional simulations can truly capture the emerging behaviour of thickness effect on fracture spacing. Our observations indicate that 2D experiments are not well suited to reproduce this kind of layer bound patterns. Instead, they are good proxies for fractures forming on thick layers of a quasi-brittle material.
4.6 Conclusions

Straight crack patterns ensue due to a combination of boundary and layer interaction effects. Curving is a manifestation of fracture interaction that cannot be suppressed by applying high differential stresses to a fracture model. Friction with binding layers homogeneously transduces stresses, causing the inhibition of crack interaction and contributing to straight crack formation.

Two dimensional fracture propagation simulations do not capture the stress fields that arise during layer interaction processes. Therefore, they are appropriate to simulate fractures in layered media when friction with stratigraphic neighbours does not dominate deformation.

Large differential stresses ensuing from the deformation of bending layers do not uniquely constrain the layer’s stress field. For rocks with low fracture density, these propagate in straight paths when subjected to large differential stresses. However, once fractures grow, their stress halos locally dominate over the remote stress field and induce curved propagation. We conclude that layer friction plays a dominant role in the deformation process, as opposed to the effects of remote differential stresses, that generates parallel fractures in layered media.
5 Impact of fracture growth on effective permeability

Contents

5.1 Abstract .......................................................... 125
5.2 Introduction ....................................................... 125
5.3 Governing equations ............................................. 128
  5.3.1 Single-phase steady state flow ......................... 128
  5.3.2 Fracture permeability ................................. 128
  5.3.3 Effective permeability of the fractured porous medium 129
  5.3.4 Fracture-matrix flux ratio ........................... 131
5.4 Domain discretisation ............................................. 132
5.5 Experimental setup .............................................. 132
  5.5.1 Model dimensions and initial flaw distribution .... 133
  5.5.2 Boundary conditions ................................ 134
5.6 Results ............................................................ 135
  5.6.1 Fracture patterns and connectivity .................. 135
  5.6.2 Fracture set characterisation ......................... 137
  5.6.3 Fracture aperture distribution ....................... 139
  5.6.4 Measured flow properties ............................. 141
  5.6.5 Flaw area versus observation area ................. 151
  5.6.6 Curved versus straight pattern: effect on flow .... 152
  5.6.7 Pattern anisotropy ................................ 155
  5.6.8 Effective permeability estimation ................... 158
5.7 Discussion ........................................................ 160
  5.7.1 Conductivity as a function of density ............... 162
  5.7.2 Impact of matrix permeability and relationship to percolation theory ............................ 163
  5.7.3 Topology and overestimation of conductivity .... 165
5.8 Conclusions ....................................................... 167
5.1 Abstract

Fracture networks exert a strong influence on flow patterns in the subsurface. We study the impact of a set of geomechanically propagating fractures on fluid flow throughout growth. The apertures of these discrete fractures are controlled by an ambient stress state. For fracture geometries at incremental steps, and under simultaneous consideration of rock matrix properties, we measure the variation of the effective permeability of the fractured mass. As fractures arrest, close, and coalesce the permeability structure dramatically changes as a function of increasing density and connectivity. Results depend on the aperture distribution and the matrix permeability. Fixed apertures overestimate effective permeabilities as they disregard possible path blockages due to fracture interaction. For a reduced region of the model, we measure a gradual increase in the conductivity up to the percolation threshold, thereafter, a pronounced increase when a path connects the flow boundaries. The simplified straight-crack version of the model underestimates the conductivity of the model. Results indicate that fracture percolation, as well as stress-dependent aperture distribution due to mechanical interactions, control the evolution of the effective permeability of the system.

5.2 Introduction

Recent numerical studies show that fracture patterns can be realistically recreated by approximating mechanical behaviour using 2D simulations [Ingraffea and Saouma, 1985; Belytschko and Black, 1999; Olson, 1993; Renshaw and Pollard, 1994b; Huang et al., 2003]. Interest in simulating fracture growth extends across a variety of application fields including: hydraulic fracturing [e.g. Boone and Ingraffea, 1990], structural analysis for civil engineering [e.g. Bazant and Verdure, 2007], composite material design for aeronautics [e.g. Camanho et al., 2006], nuclear waste disposal risk assessment [e.g. Shen et al., 2004], and analysis of flow and mechanical properties of fractured reservoirs [cf. Zoback, 2007]. Fractures not only damage rocks making them weaker and causing fragmentation, they also influence their flow properties changing the speed at which they conduct liquids, gases, transport contaminants, among others. The flow properties of these rock masses change as they undergo such irreversible geometric alterations.

Effective permeability is a flow property of special interest to the reservoir modelling community because it measures the hydraulic conductivity of a system at a
larger scale. In the case of growing fracture sets, we evaluate the mechanical transition that occurs at percolation, when the system becomes a thoroughly interconnected network that bridges two boundaries together. We examine the trend of the effective permeability as a function of fracture aperture, density, and growth stage.

Previous percolation-theory driven results highlight the lack of geometrically realistic and sufficiently resolved simulations [Bogdanov et al., 2007]. They often rely on stochastically generated fracture datasets that mimic reality but do not reproduce the internal structure arising due to mechanical interaction during growth. Huseby et al. [1997] studied the effect of crack density, as well as topological and geometric interconnectivity, on the flow properties of the fractured mass. Initial experiments included measuring effective permeability of fracture sets with different connectivity levels. The experiments were conducted on randomly generated datasets consisting of equal length 3D polygonal fractures. Later, Bogdanov et al. [2007], conducted similar numerical experiments, this time with a fracture size power-law length distribution. Both authors found a systematic distribution of the effective permeability of the fractured system in which effective permeability increases linearly as a function of density. The trend can be approximated by two linear segments: before percolation permeability increases linearly with density, after percolation permeability values are several orders of magnitude higher, but continue to increase linearly with density. The authors report a stark dependence on the fracture to matrix permeability ratio: the larger the ratio, the stronger the jump of effective permeability at percolation onset. They also observe a slightly higher effective permeability for percolated models as opposed to their well interconnected counterparts. Apart from the uncertainty in the fracture geometry, fluid flow models for fractured porous media usually make simplistic assumptions about the relationship between fracture length and aperture or even collapse its range into single values. Matthai and Belayneh [2004] have shown that such models yield highly unrealistic results.

Geomechanical fracture set generation poses an interesting alternative to stochastically generated fracture sets. Its main disadvantage with respect to the latter is that it is more computationally expensive. However, there is a strong tendency to fall back to this thorough approach due to the level of geomechanical detail it provides. The simplicity of randomly generated datasets does not capture complex mechanical interactions between fractures that occur during growth. Therefore, they often contain patterns that do not exhibit fracture self-organisation properties and reproduce the statistics of original measured data, such as spacing and aperture distributions, without capturing the underlying mechanics that govern their
development. These patterns depend on the rock’s deformation history and its material properties and have a stark influence on the flow behaviour through the fractured rock mass. One example is density: in randomly generated datasets, density is created by adding more or larger fractures, while in geomechanical datasets it is a by-product of growth and coalescence. Furthermore, randomly generated datasets do not model fracture curving. Thus, a fracture is always planar and its orientation predefines its connectivity. In mechanically informed simulations, this is not the case: fractures can grow in any shape and pattern, and thus, can greatly enhance connectivity without significantly increasing density.

Related work on fracture pattern generation relies on a sub-grid representation of fractures and on the extension of the finite element method to capture them as discontinuities in the displacement field [Belytschko and Black, 1999]. The advantage of the finite element-based modelling of deformation is the simplicity of the numerical discretisation of the solved equations. The idea is that by retaining an accurate representation of topology and material interfaces the numerical method is relieved of a sub-mesh representation of the geometry and there is more room to capture complex behaviour, such as compaction, damage, and inelastic deformation. An example that couples flow and deformation is hydraulic fracture propagation by Boone and Ingraffea [1990] and Secchi et al. [2007]. Using an approach where fracture and matrix domains are discretely represented allows for its swift integration into compatible flow codes to measure multiphase flow and other transport properties.

We quantify the change of the conductive properties by measuring effective permeability and fracture-matrix flux ratio as a function of increasing density. We numerically measure flow properties of fractured rock analogues by conducting a steady state single-phase flow analysis taking into account full fracture geometry as well as flow through both matrix and fractures [Matthai and Belayneh, 2004]. Hence, results are applicable to fractured porous media as opposed to crystalline rocks only.

This chapter is organised as follows. Section 2 discusses the methodology to propagate cracks and compute effective permeability of a fractured rock. Section 3 presents the simulation results including fracture pattern characterisation, aperture distributions, and effective permeability trends of the system. In Section 4 we discuss formation of fracture patterns, their statistics, and the behaviour of the upscaled permeability around the percolation threshold. Finally, Section 5 concludes this work.
5.3 Governing equations

We compute the effective permeability and fracture-matrix flux ratio to evaluate the effect of the fractures on the hydraulic conductivity of a porous medium. Additionally, we measure the effects of density, connectivity, and spacing on the conductivity of a growing fracture set. All datasets are generated using the geomechanical fracture growth engine presented in Chapter 3.

5.3.1 Single-phase steady state flow

We consider steady-state pressure driven fluid flow that, in absence of sources and sinks, obeys conservation of volume

\[ \nabla \cdot q = 0, \]  

where \( q \) (m/s) is the specific discharge given by Darcy’s law,

\[ q = \frac{k}{\mu} \nabla p, \]  \hspace{1cm} (5.1)

and \( p \) (Pa) and \( \mu \) (Pa s) are the fluid pressure and dynamic viscosity respectively. For a porous medium, \( k \) refers to the matrix permeability. The discretisation of this equation is described in Section 2.3.3. It follows that each element of the discretised domain has a permeability associated to it. For matrix elements we define a constant matrix permeability, \( k_m \). In the fractures we derive an equivalent porous medium fracture permeability, \( k_f \), using a piecewise parallel plate approximation. This is explained with detail in the next section.

The derived velocities and fluxes, \( v_D \) and \( q_D \), correspond to the direction and magnitude of the Darcy velocity respectively. The interstitial velocity is \( v_I = v_D / \phi \), where \( \phi \) is the porosity. For lower-dimensional elements \( \phi \) is scaled with the fracture aperture on an element-by-element basis [Matthai and Belayneh, 2004].

5.3.2 Fracture permeability

Fracture permeability, \( k_f \), is defined as a piecewise constant value along the fracture centreline. At any given location, we compute permeability from local aperture using the parallel plate law. This assumes that the flow is laminar and the fracture has smooth, step-wise parallel walls with a local separation of \( h \) [Kranz et al., 1979; Witherspoon et al., 1980]

\[ k_f = \frac{h^2}{12}, \]  \hspace{1cm} (5.3)
where the height \( h \) is the local fracture aperture. The fracture centreline is the skeleton of the fracture BREP.

Sisavath et al. [2003] showed that the parallel plate approach overpredicts flow in fractures by a factor less than two, because it disregards the influence of fracture roughness on fluid flow and requires the definition of a unique representative aperture for the entire fracture. However, we apply the parallel plate law in a stepped, local manner and assume channel flow through parallel walls of the fracture at each element of its centreline, as opposed to attempting to identify a single hydraulic aperture representative of the flow capacity of the entire fracture. Figure 5.2 shows how the aperture of a fracture is captured in an element-by-element basis yielding a local approximation of its hydraulic properties.

By extracting the centreline, we can conduct a flow simulation on a body that has triangle elements representing the matrix and sub-dimensional line elements representing the fracture. We capture the varying aperture of the fracture by assigning a different conductivity to each of its elements. At any given point, we can read off the mesh the local aperture of the fracture, and compute a local permeability. Therefore, we define for a single fracture a variable \( k_f \) along its centreline

\[
k_f = \langle k_{f0} | \ldots | k_{fi} | \ldots | k_{fn} \rangle,
\]

(5.4)

where fracture \( f \) has \( n \) points along its centreline. Due to the lower-dimensional representation of the fracture, we weigh properties defined at the lines, such as the permeability and the porosity, with the local aperture to capture the actual thickness of the reduced element. By using line elements we can generate meshes with significantly less elements and obtain equivalent results [Juanes et al., 2002] as if they would have been volumetrically discretised. Figure 5.1 illustrates the piecewise mapping of the apertures along the fracture onto its centreline.

### 5.3.3 Effective permeability of the fractured porous medium

We integrate the boundary fluxes to compute the total model throughput \( q \). For a given macroscopic fluid flow gradient we approximate \( k_{eff} \) as

\[
k_{eff} = \frac{q \mu L}{A(p(u) - p(d))},
\]

(5.5)

where \( L \) (m) is the length of the model in the direction of flow, \( A \) (\( m^2 \)) is the area of a cross section perpendicular to the flow, and \( p(u) \) and \( p(d) \) (Pa) are the fluid pressures applied to the upstream and downstream model boundaries. The modelled
Figure 5.1: Snapshots of fracture growth. Fracture growth illustrated by snapshots of the fracture geometry at different growth stages. Initially fractures are straight and show little interaction. As they grow, and start to connect, they inhibit growth of proximal fractures. Insets highlight the observation areas A and B where we compute $k_{eff}$ at selected growth stages.

Fluid is water. Figure 5.3 illustrates these quantities. Here, $k_{eff}$ is a measure of the conductivity of the fractured porous medium as it takes into consideration flow through fractures and matrix. Matthai and Belayneh [2004] present the validation and other details of this approach.
Figure 5.2: Fracture centreline and aperture approximation. The centreline tracks the skeleton of the fracture shape. Black dots denote the nodes on the fracture wall (such as $f_n$, $f_{n+1}$ and $f_{n-1}$). White dots represent the corresponding centreline nodes, $c_i$ and $c_{i+1}$. $f_n$ is a fracture tip. $a_i$ is the aperture of the fracture at $c_i$; in this case $a_i$ equals the distance between $f_{n+1}$ and $f_{n-1}$. The thick line represents a line element of the centreline. The aperture assigned to the local line element, $a_{fi}$, is the average between $a_i$ and $a_{i+1}$.

Figure 5.3: Effective permeability boundary conditions. A fluid pressure gradient $(p(u), p(d))$ is applied to the observation area of length $L$, no flow is allowed on the upper and lower boundaries. The fluxes $q_{in}$ and $q_{out}$ are the total in and out flux, respectively. These are equal.

### 5.3.4 Fracture-matrix flux ratio

The fracture-matrix flux ratio [Matthai et al., 2007] is a measure of the importance of the fractures as compared to that of the role matrix in conducting fluid flow

$$\frac{q_f}{q_m} = \frac{q_t - q_m}{q_m}, \quad (5.6)$$
where $q_t$ is the flux integrated over any model cross-section perpendicular to the far-field gradient, and $q_m$ is the total flux through an equivalent matrix with no fractures

$$q_m = \phi_x \frac{k_m p(u) - p(d)}{L}$$

(5.7)

where $\phi_x$ is the cross-sectional porosity defined as

$$\phi_x = \frac{V_m}{V_m + V_f}$$

(5.8)

and $V_m$ is the total volume of the matrix and $V_f$ is the volume occupied by the fractures.

### 5.4 Domain discretisation

The optimal discretisation for flow does not coincide with the discretisation of the domain for mechanical computations because, in this case, the inside of the fractures must be meshed. The straightforward approach is to use volumetric elements such as triangles and quadrilaterals. For isotropic meshers, the quality of these triangles is a function of their aspect ratio and size. Thus, meshers create a large amount of minuscule and/or ill-formed triangular elements inside of thin fractures. In contrast, elongated elements aligned with the flow are better suited to capture flow behaviour in high aspect ratio fractures [Matthai and Belayneh, 2004; Paluszny et al., 2007]. Moreover, the fracture volumetric representation must capture its variant aperture which is often close to the numerical limits of the model. In order to avoid these discretisation complexities, fractures are represented using sub-dimensional quadratic line elements. These incorporate their piecewise thickness at each element by storing a local aperture property used to weigh their conductivity [Juanes et al., 2002].

### 5.5 Experimental setup

A total of 110 growth iterations were performed by applying large differential displacement conditions at the boundaries. After the 100th iteration, growing was dominated by fractures that had transcended the flaw area. The propagation simulator stops after the 110th iteration. At this point, the fracture set is saturated because cracks only grow at boundaries and stop growing elsewhere. Figure 5.4 shows an example of the initial state and fully saturated state.
Figure 5.4: Initial and final steps of a growth simulation. Flaws are initially randomly placed in the flaw area and grow within the limits of the model’s propagation area. \( k_{\text{eff}} \) is measured in the observation area. Upon reaching saturation, growth only occurs at the left and right extremes of the propagation area.

### 5.5.1 Model dimensions and initial flaw distribution

We assume a Young’s Modulus of 20GPa, Poisson’s ratio 0.25, and \( K_{IC} = 15 \times 10^6 \text{Pa m}^{1/2} \) – average limestone material properties. In all cases, total model dimensions are \( 1m \times 4m \). The specimen has three areas of interest: flaw, propagation, and observation area. The flaw area, where the simulator initially places uniformly distributed flaws, is \( 0.8m \times 2m \). The propagation area, where fractures propagate, is \( 0.9m \times 3.9m \). The observation area, where we measure \( k_{\text{eff}} \), is a sub-region of the propagation area and its dimensions are approximately \( 0.5m \times 0.5m \). Figure 5.1 shows how the flaws distribute in the flaw area. The propagation area is a delimiting and somewhat longer bounding area. Figure 5.4 shows these areas for the deformed specimen.

Fracture sets grow from an initial population of 100 flaws. Flaws have an initial horizontal orientation. Their position has a uniform random distribution, and their lengths follow a Gaussian distribution with mean 0.004m and standard deviation 0.045m² (approximate range of: 0.004m ± 0.002m). Flaw density, total flaw length divided by total propagation area [Underwood, 1970], is \( \sim 0.0058m^{-1} \),
and this coincides with values reported by Renshaw [1996]. Although initial flaw
density should be representative of the initial state of the modelled rock mass, it
does not seem to play a dominant role in fracture pattern formation [Renshaw, 1996;
Renshaw and Park, 1997].

Spacing, as defined in Section 3.6, allows adjusting the initial density of the
flaws. The larger the minimum spacing, the less initial flaw clusters will be formed.
Minimum spacing between two flaw centres is set to three times the initial flaw length.
Thus, in the process of populating the flaw area with flaws, no flaw centre is at a
distance lesser than 0.012m of any node of a previously placed flaw.

The described experimental setup allows for significant fracture growth beyond
the observation area. Although units are in meters, from the mechanical point of
view, the model is completely scale independent as neither failure nor propagation
criteria are size dependent. The method only assumes the medium to be a continuum
and that the fracture process zone is smaller than the initial flaw size. Nevertheless,
the parallel plate law used for the computation of the permeability of the fractures is
scale dependent. Therefore, the results presented in this work are applicable to the
meter scale only.

5.5.2 Boundary conditions

We define two different sets of boundary conditions. The mechanical boundary
conditions constrain the deformation of the model area. Specifically, we apply
large differential stresses [Thomas and Pollard, 1993]. This corresponds to applying
tension at the upper and lower boundaries of the model, and compression on the left
and right boundaries on the horizontal direction. This setup allows for significant
fracture growth beyond the percolation threshold within at least two observation
areas. Observation areas are a subset of the model that is 42% of the height and
10.5% of the original specimen width. Observation areas A and B are samples of
the set close to the centre (see Figure 5.1). For the \( k_{eff} \) computation, we only use
the observation areas of the original model, to which we apply a hydrostatic pressure
gradient using fixed pressures on opposing boundaries oriented perpendicularly to the
fractures. Top and bottom are no flow boundaries.

We simulate two geometric cases: fixed apertures for all fractures, which are
artificially set by assigning a fixed permeability to all centrelines, and geomechanical
generated fractures due to extensional displacement of 0.01 and 0.001m in the vertical
direction. We measure flow properties for a matrix with permeability of \( 10^{-15} \text{m}^2 \) (1
mD) and $10^{-12} \text{m}^2$ (1 D) - which are representative of carbonate reservoir $k_m$ [Dullien, 1992].

5.6 Results

In this section, we present the mechanical and flow results of all conducted experiments. Figure 5.1 shows observation areas A and B. We measure the flow properties of these two sub-regions at consecutive growth stages to monitor influence of growth and density on the permeability of the system. Specifically, we measure $k_{\text{eff}}$ and the $q_f/q_m$ ratio in the direction perpendicular to the fluid pressure gradient, at every tenth growth step. Figure 5.1 shows the fracture dataset at iterations 0, 10, 30, and 110. The mesh has initially $\sim 40k$ nodes and $\sim 76k$ triangles. As cracks propagate, node density increases. By the 110th iteration the mesh has $\sim 370k$ nodes, including mid-side nodes, and 673k elements.

5.6.1 Fracture patterns and connectivity

Fractures initially propagate following straight paths. At this stage, remote, large differential stresses dominate local stress fields. As they grow, they create stress shadows around them that influence, and even inhibit, growth of neighbouring cracks. Once proximal, they engage in complex inter-dependent growth behaviour, such as coalescence and formation of arrays. *En echelon* cracks connect as they curve toward each other. This linked pattern, also frequently observed in nature [Nicholson and Pollard, 1985], plays an important role in increasing the permeability of the rock. At double connection points, where both tips coalesce on each other, matrix blocks disconnect from the matrix and enable a double flow path between two fractures. At later stages of growth, we observe a widespread occurrence of mechanically interacting fractures in all generated geometries.

When two fractures intersect, their bodies merge and subsequently act as one object. Therefore, the dataset connectivity is derived from the ratio of connected fractures at a growth step to the number of initial disconnected flaws. We observe a reduction in the total number of fractures from 100 to 82 throughout deformation. This reduction of fractures implies the gain of approximately 20% in the connectivity of the system. Figure 5.5 shows how connectivity increases as a function of density. Between the 20th and 30th iteration there is a pronounced increase of 10% in the connectivity, by then, most fractures reach their neighbours, some intersect, while others are inhibited. Thereafter, connectivity increases almost linearly. We
identify no correlation between connectivity and percolation. Fracture clusters form as connectivity increases. Figure 5.6 shows how the increase in connectivity relates to their geometric characteristics. Clusters sizes range between 2-5 fractures. Cluster extension (in the $x$ direction) ranges between $\sim0.019m$ and $\sim1m$, which corresponds to half of the total flaw area extension. We observe that both maximum cluster size and maximum cluster extension increase linearly as a function of density.

![Figure 5.5: Fracture connectivity. The number of clusters increases as growth progresses. There is a $\sim20\%$ gain in connectivity throughout growth. There is a large increase in connectivity between the 20th and 30th iteration due to a large amount of small-scale fractures that intersect their neighbours.](image)

We observe no tip-to-tip coalescence, but mainly connections through intersection of tips on fracture walls. As predicted by Melin [1982], fractures tend to avoid each other as their tips approach due to the influence they exert on each other’s stress fields. We observe that most fractures intersect at straight angles. These results match expected behaviour because we do not model cohesion nor traction between the fracture walls.

Although we apply large differential stresses to the sample throughout deformation, we do not observe predominant straight paths. Fractures shield one another from the remote stresses and perturb the field locally, originating emergent patterns governed by local stress constellations. We only observe predominant straight path formation, due to the applied large differential stresses, at initial stages of growth as reported in the literature [Olson and Pollard, 1989; Olson, 1993].
5.6.2 Fracture set characterisation

Density increments linearly throughout deformation. For the propagation area, it ranges from \( \sim 0.0026 \text{m}^{-1} \) to \( \sim 0.6085 \text{m}^{-1} \). For the flaw area, it ranges between \( \sim 0.0058 \text{m}^{-1} \) and \( \sim 1.33 \text{m}^{-1} \). Observation area density values range from \( \sim 0.0073 \text{m}^{-1} \) to \( \sim 1.2118 \text{m}^{-1} \). The flaw area is denser than the propagation area because it is where most of the growth takes place. The difference is pronounced because the propagation area has empty space that pushes density values down. The density of both observation areas is quite high for all iterations, except for the last 20, in which the flaw area becomes saturated. Observation areas are particularly dense as compared to the rest of the model. Figure 5.7 shows the trends of the density values.

We measure spacing of fractures within the flaw area. During growth, it decreases by a factor of six, from \( \sim 0.4069 \text{m} \) to \( \sim 0.0721 \text{m} \). The large number of small non-interacting fractures growing independently causes an initial sharp decrease of 62.5\% in the spacing around the 10th iteration. Thereafter, there is a further reduction in spacing of 20\% distributed along the remaining 100 iterations. This indicates that the ten-fold increase in density reflects the generation of large amounts of new fracture surface area, enhanced by fracture curving. Fractures that propagate following a
Figure 5.7: Fracture density. Fracture density increases almost ten fold throughout growth. The density depicted by black and white dots is only computed for observation areas A and B, respectively. Diamonds show the density of the propagation area. Squares show the density of the flaw area.

curved path generate more surface area and increase density while not affecting the spacing of the set. See Figure 5.8 for a plot of the spacing trend during growth.

Figure 5.8 shows how mean and standard deviation of the lengths develop during growth. They increase from an initial value of $\mu = \sim 0.019 \text{m}$ ($\sigma^2 = 5.92 \times 10^{-5} \text{ m}^2$) up to $\mu = \sim 0.24 \text{m}$ ($\sigma^2 = 0.2 \text{ m}^2$). High standard deviation values are indicative that a normal distribution does not adequately capture length distribution. Instead, as shown in Figure 5.9, lengths display a lognormal distribution expressing the skew distribution of sizes resulting from growth. Initially, fracture lengths follow a normal distribution. A 10% connectivity increase, between the 20th and 30th growth iteration, causes the length distribution to shift to lognormal. As the dataset grows, a significant amount of cracks arrest due to the effect of proximal fracture arrays. When fractures align, even though not connected, they express a combined stress shadow that inhibits growth in the surrounding area. The result is a skewed fracture length distribution, with a bulk of cracks of size between 0.1 and 0.2m and a long leading edge of a minority of cracks between 0.3 and 0.65m.
### 5.6.3 Fracture aperture distribution

We compute apertures for two macroscopic stress states induced by displacement boundary conditions of $10^{-3}$m and $10^{-2}$m respectively. Deformation is applied to the entire model, not only to the observation or flaw areas. Mechanical interactions between fractures of the entire dataset determine the aperture distribution. Therefore, geometries that might appear equal at simple sight, such as iteration 100 and 110, have a different aperture distribution due to the deformation behaviour of the sample. For a displacement of $10^{-2}$m, we obtain a mean aperture of $\sim 0.0008$m, for the first iteration, and up to $\sim 0.007$m for the last. For the first iteration, we measure a minimum aperture of $\sim 0.0011$m and a maximum of $\sim 0.0013$m. For the last iteration, we measure a significantly lower minimum of $\sim 1.76 \times 10^{-5}$m and a considerably higher maximum of $\sim 0.033$m. We obtain correspondingly smaller quantities for smaller boundary displacements. Figure 5.10 shows an increase of the fracture aperture by an order of magnitude during the first ten iterations.

For 100 fractures the apertures at the 10th iteration exhibit an uneven distribution (see Figure 5.11). Clustered, aligned fractures exhibit the largest apertures. At this stage, there are two predominant regions in the model. The lower left cluster with
a significantly higher maximum aperture is the one to deform the most. *En echelon* cracks that populate the upper right belt of the dataset all exhibit similar maxima.

Figure 5.12 shows in detail three stages of sampled apertures. When fractures have similar lengths, cracks exhibit an aperture distribution that is higher toward the centre and lower toward the tips. In this particular case, aligned cracks eventually coalesce into one larger fracture. At this point, the maximum aperture jumps by an order of magnitude. In the last case, interaction between the stress halos of the two larger fractures causes their apertures distribution to diminish in proximal regions.

Figure 5.13 groups apertures sampled at all centreline nodes for an external displacement of $10^{-3}$m. These exhibit a lognormal distribution throughout growth. Around 50% of the sampled points retain their initial aperture throughout growth. As propagation advances, a small amount of sample points measure significantly higher apertures.

Fracture volume displays the same behaviour. We observe that aperture mean continues to rise above the percolation threshold due to growth that occurring beyond the observation area: fractures become longer and their aperture increases. In the next section, we investigate the effect of the aperture distribution on the flow properties of the system.
Figure 5.10: Flaw area and apertures as a function of growth. The plot depicts the decadic logarithm of mean geomechanical apertures for all stages of growth (observation area A). Apertures increase by an order of magnitude during the first 10 iterations. Then, they remain stagnant at a value that is slightly lower than the remote displacement. We observe a similar aperture behavior for all applied displacements.

5.6.4 Measured flow properties

We compute the $k_{\text{eff}}$ and $q_f/q_m$ ratio of a developing fracture set taking into account aperture variability due to mechanical effects. Displacements applied at the boundaries and deformation history control fracture apertures as shown in Figure 5.14. We sample this aperture distribution to compute an element-based variable $k_f$ for each fracture using the parallel plate law.

We measure $k_{\text{eff}}$ and $q_f/q_m$ as density increases. We compare measurements to a system that does not take into account the effects of realistic apertures, but instead assumes a fixed aperture value for all fractures. As expected, the distribution of stress-dependent apertures is far from homogeneous. Results indicate that fracture size is not the only factor that influences its maximum aperture. Its position within the set is key to determining how open it is and how much blockage its path has.

As fractures grow, their shapes organise forming connected patterns. We observe that the total permeability of the system increases proportionally to the available fracture volume and its conductivity. For fixed apertures, the trend is clear, the more growth and the higher density, the higher the permeability. Conductivity after percolation continues to increase. For geomechanical apertures, permeability is lower than in the previous case, although the specimen has undergone a purely tensile
Figure 5.11: Flaw area initial apertures. After 10 iterations of growth, apertures are smaller toward the tips, and larger toward the centre. Apertures correspond to a tensile displacement of 0.01m, they are exaggerated $\times 100$.

deformation. Before percolation, $k_{eff}$ steadily increases until the fracture network percolates. Thereafter, $k_{eff}$ fluctuates as some of the paths partially close due to further growth.

**Effective permeability**

The following summarises the pre-percolation $k_{eff}$ measurements, for iterations 0 to 70. See Figure 5.5 for the definition of the areas. See Figure 5.15 and 5.16 for the $k_{eff}$ plot.

**Initial behavior - low $k_m$** Figures 5.15 (a) and 5.16 (a) are plots of the measured $k_{eff}$ before percolation for $k_m = 10^{-15}$m$^2$. For fixed aperture datasets $k_{eff}$ increases by an order of magnitude for both observation areas. For mechanically generated apertures, larger boundary displacements exhibit a larger increase in $k_{eff}$ with density. For observation area A, large displacements create a rather well connected network, and behaviour is similar to the fixed aperture case. Small displacements only trigger a slight increase in the permeability of the rock mass. For observation area B, even large boundary displacements do not accomplish a well-interconnected network, yielding a consistently lower $k_{eff}$ than fixed apertures. For observation area B, small boundary displacements hardly increase $k_{eff}$ at all.
Figure 5.12: Aperture distribution. Aperture distribution at three stages of growth. Glyphs exaggerated $\times 1000$. The size of each glyph is proportional to the aperture of the fracture at the line element. When fractures connect, they behave as one and the aperture distribution along the centreline changes.

Initial behaviour - high $k_m$  Figures 5.15 (b) and 5.16 (b) are plots of the measured $k_{eff}$ before percolation for $k_m = 10^{12}$ m$^2$. The increase in permeability before percolation is not pronounced because the matrix channels much of the flow regardless of flaw density. Thus, the porous medium bridges small gaps between unconnected cracks efficiently, increasing total flux through the specimen. Results indicate a quasi-steady $k_{eff}$ for both small and large displacements.

The following summarises overall $k_{eff}$ measurements, for all iterations (0 to 110). See Figure 5.5 for the definition of the area. See Figures 5.17 and 5.18 for the $k_{eff}$
Figure 5.13: Flaw area aperture bins. For a displacement of $10^{-3}$m at the top boundary we measure the aperture of the fractures at all nodes of their centrelines. The apertures exhibit a lognormal distribution with a bulk of sample points retaining their value throughout growth, and a minority increasing to up to 0.004m.

**Fixed apertures** We measure $k_{eff}$ for low matrix permeability of $1.0 \times 10^{-15}$m$^2$ only. For apertures of 0.01m, $k_{eff}$ increases by over an order of magnitude before reaching percolation, and then jumps by seven orders of magnitude to $\sim 2 \times 10^{-7}$ m$^2$ for both observation areas. For apertures of 0.001m, $k_{eff}$ rises from $1.0 \times 10^{-15}$m$^2$ to $1.93 \times 10^{-10}$m$^2$ and $1.94 \times 10^{-10}$ m$^2$ for A and B respectively. B’s slightly higher permeability agrees with its more dense geometry. The jump in the $k_{eff}$ is due to the attainment of a topological connection of the flow boundaries, which has a fixed aperture along its path.

**Large displacement induced apertures - 0.01m** For fracture apertures of 0.01m, $k_{eff}$ values measured for both observation areas are in partial agreement. For high matrix permeability, both observation areas yield $k_{eff}$ values in the order of $10^{-12}$m$^2$. For low matrix permeability, observation area A measures a jump in $k_{eff}$ of two orders of magnitude at the attainment of percolation. At this stage, flow focuses on the central fractures. In contrast, observation area B does not exhibit a drastic jump. Instead, it slightly increases its conductivity. Observation area B has two dominant clusters that accommodate deformation simultaneously and cast
Figure 5.14: Aperture distribution. Depicts a sequence of aperture distributions for iterations (a) 0, (b) 10, (c) 20, and (d) 30 (observation area A). Apertures are initially controlled by fracture size. Fracture interaction leads to an uneven distribution of the apertures. In this example, the central fracture cluster accommodates most deformation.

Small displacement induced apertures - 0.001m For smaller fracture apertures, $k_{eff}$ values measured for both observation areas agree. For low matrix permeability, we measure a maximum $k_{eff}$ of $4.62 \times 10^{-15} \text{m}^2$ for A, and $2.07 \times 10^{-15} \text{m}^2$ for B. Although density soars, permeability does not significantly increase. This is due to the uneven aperture distribution and partial blockage of the percolating network due to mechanical effects of stress shadowing. For high matrix permeability, we measure a $k_{eff}$ around $10^{-12} \text{m}^2$ at all stages, for both observation areas. In this case, the contribution of the matrix is significantly higher than the fractures, which only contribute minimally to the total $k_{eff}$ of the system. At the 110th iteration, we measure a $k_{eff}$ of $\sim 1.0046 \times 10^{-12} \text{m}^2$ for A, and $\sim 1.0027 \times 10^{-12} \text{m}^2$ for B.
Figure 5.15: Observation area A: effective permeability before percolation. Detail of the early time behaviour of the effective permeability. Only seven iterations are showed here (Figure 5.17 shows data for all iterations). In (a), matrix permeability is low \( k_m = 10^{-15}\text{m}^2 \), while in (b) its high \( k_m = 10^{-12}\text{m}^2 \). Effective \( k_{\text{eff}} \) increases by almost two orders of magnitude for the fixed aperture datasets. For large mechanical apertures the behaviour is similar. For small mechanical apertures effective permeability increases by almost an order of magnitude. Plots are shown up to the point of local percolation.

Figure 5.16: Observation area B: effective permeability before percolation. Detail of the early time behaviour of the effective permeability. In (a), matrix permeability is low \( k_m = 10^{-15}\text{m}^2 \), while in (b) its high \( k_m = 10^{-12}\text{m}^2 \). Only seven iterations are showed here (Figure 5.18 shows data for all iterations). Effective \( k_{\text{eff}} \) increases by almost two orders of magnitude before reaching the percolation threshold. For fixed apertures \( k_{\text{eff}} \) increases by almost two orders of magnitude. For the stress-dependent aperture case \( k_{\text{eff}} \) increases by around an order of magnitude for the low permeable matrix, and minimally for the high permeability matrix. Plots are shown up to the point of local percolation.
Figure 5.17: Observation area A: effective permeability. The \( k_{eff} \) is plotted in decadic logarithmic scale. Flat lines are for fixed aperture values: 0.01m and 0.001m for (a) and (b) respectively. Dashed lines are for stress dependent apertures by applying a displacement boundary condition of 0.01m and 0.001m for (a) and (b) respectively, and for a matrix permeability of \( 10^{-12} \) m\(^2\). Stippled lines are for matrix permeability of \( 10^{-15} \) m\(^2\) respectively. For fixed apertures of 0.01m we results exhibit a jump of 7 orders of magnitude at local percolation. For smaller fixed apertures of 0.001m the jump is less pronounced at 4 orders of magnitude. For mechanical apertures we observe a steady increase.

Figure 5.18: Observation area B: effective permeability. The \( k_{eff} \) is plotted in decadic logarithmic scale. Flat lines are for fixed aperture values: 0.01m and 0.001m for (a) and (b) respectively. Dashed lines are for stress dependent apertures by applying a displacement boundary condition of 0.01m and 0.001m for (a) and (b) respectively, and for a matrix permeability of \( 10^{-12} \) m\(^2\). We observe a jump increase for fixed apertures. For stress dependent apertures, the increase in conductivity is quasi-linear. Stippled lines are for matrix permeability of \( 10^{-15} \) m\(^2\) respectively.
Fracture-matrix flux ratio

The $q_f/q_m$ ratio expresses the effect of the fractures on the total flux through the fractured porous medium in comparison to the flow through the matrix. Figure 5.19 shows the $q_f/q_m$ ratio measurements of observation areas A and B. These are plotted as a function of growth iteration, density and connectivity. For A and B, density increases to $0.6\,m^{-1}$ within the first 30 iterations after which it increases linearly up to $1.2\,m^{-1}$ and $0.9\,m^{-1}$ respectively. Fracture-matrix flux ratios increase differently with respect to these three factors. As a function of the growth iteration, the increase is smoother for B as compared to A. In some cases, an small increase in connectivity causes a sharp increase of the $q_f/q_m$ as for low matrix permeability and large fracture apertures. In contrast, for low fracture permeability fracture apertures the $q_f/q_m$ ratio seems to increase as a function of density. For fixed apertures, we measure similar $q_f/q_m$ ratios which increase up to three orders of magnitude until percolation is attained. Subsequently, we observe a jump of up to six orders of magnitude after a connecting path is formed. In contrast, stress dependent apertures exhibit a steady increasing trend of the $q_f/q_m$ ratio throughout percolation. For higher values of $k_m$ we measure lower $q_f/q_m$ ratios. The results compare to the effective permeability measurements. We observe that $k_{eff}$ measurements do not fully capture the initial steep increase in conductivity due to a connectivity increase of 10%. In contrast, $q_f/q_m$ captures this initial behaviour by varying several orders of magnitude within this first phase.

Velocity and flux trends

Fluid flow velocities in fractures vary up to six orders of magnitude (see Figure 5.20, Figure 5.21 for averages). For a low matrix permeability of $10^{-15}\,m^2$ and a fixed aperture distribution ranging from $10^{-3}\,m$ to $10^{-2}\,m$, we observe the following behaviour of the velocity and flux profiles: (a) initially fractures channel some of the flow, and the matrix contributes significantly to the total flow. (b) Before percolation, connective bridges between neighbouring fractures form clusters that locally speed up flow. (c) After percolation, the flow is channelled through a preferred path, and the matrix does not make a significant contribution to the flow. For geomechanical apertures generated by applying a remote tensile stress, we observe: (a) flux is channelled through some fractures at a greater speed than others, proportional to the local fracture aperture. (b) Preferred paths are not consistent throughout growth stages, some paths close after intense flux contributions, while others open up at final
Figure 5.19: Fracture-matrix flux ratio. The $q_f/q_m$ ratio increases for observation area A and observation area B as a function of (a) and (b) growth iteration, (c) and (d) density, and (e) and (f) connectivity. Black and white dots depict the normalised fracture-matrix flux ratios for a low permeable matrix, while squares and diamonds depict high permeability values. In all cases apertures have been mechanically computed.
stages of development. (c) As compared to the fixed aperture case, flow is significantly higher at the larger fractures, and overall velocities peak at higher values. (d) Smaller fractures that lie on stress shadows of larger ones become closed and channel little to no flow. For high matrix permeability, $k_m = 10^{-12} \text{m}^2$, flow is primarily conducted through the matrix. However, as the dataset grows, we observe partial channelling of flow through fractures with large apertures. This is concordant with the measured $k_{eff}$ indicating that, in this case, attainment of percolation does not affect the overall conductivity of the system. For fixed apertures we assume a constant $k_f$. In this case, a large $q_f/q_m$ ratio hints at more flow focus and, therefore, a significant larger increase of the $k_{eff}$ of the fractured mass. Figures 5.22 (a) and (b) depict velocities at different growth stages for fixed and mechanical apertures for observation areas A and B respectively.

Figure 5.20: Velocity variation within fractures. The histogram plots the fraction of nodes with specific volume fluxes. Velocities vary up to six orders of magnitude. They exhibit a parabola profile proportional to the aperture distribution of the dataset.

Velocities are higher at the fracture tips where there is more flux and apertures are smaller. Fluxes are correspondingly higher around isolated tips and at areas of the matrix that bridge flow through disconnected but proximal fractures. This intensive flow rate would probably further reduce tensile strength of the connecting rock and
accelerate fracture coalescence. However, this phenomenon is not modelled within this work. Furthermore, due to flow channelling through fractures, diverse regions of the matrix become isolated from the flow. Some are of a blocky nature, for example, in the case where two tips coalesce against each other in a hooking manner.

5.6.5 Flaw area versus observation area

Measurements of permeability within the observation area only sample two sub-regions of the dataset. Observation areas cover 15.6% of the total flaw area and sample the model at regions where crack paths eventually subdivide the observation area into two or more domains. Figure 5.23 graphs the difference between the measurements. Results indicate that the observation area measurements predict the original flaw area effective permeability within an order of magnitude, with several exceptions. For fixed apertures, once a path connects the flow boundaries, $k_{eff}$ soars, overpredicting the permeability by up to 9 orders of magnitude. For stress-dependent apertures, effective permeability values are consistent with the flaw area measurements with one exception with large apertures that overpredicts $k_{eff}$ by 3 orders of magnitude.
Figure 5.22: Velocities. (a) Observation area A and (b) observation area B. (i) Geometry of the observation area at iterations 0, 10, 20, 30, 70 and 110. (ii) Interstitial velocity fields for fixed apertures at 0.001m. Bigger arrows imply higher velocities. (iii) Interstitial velocity fields for mechanical apertures generated by a tensile deformation of 0.001m in the vertical direction. Matrix permeability is fixed at $10^{-15}$ m$^2$, fracture permeability is determined by the parallel plate law. (iv) Same as (iii), except matrix permeability is fixed at $10^{-12}$ m$^2$.

### 5.6.6 Curved versus straight pattern: effect on flow

The boundary conditions applied to the initial set of flaws generate a crack pattern that exhibits a high degree of curvature. By growing the same set of flaws, but restricting curvature, we obtain an “equivalent” fracture pattern in which all fractures
Figure 5.23: Flaw area versus observation area. Compares the decadic logarithm of the effective permeability between the flaw area and the observation areas for (a) fixed and (b) stress-dependent apertures. In (a), where apertures are constant, the observation area significantly overestimates the permeability of the total flaw area after percolation. In (b), permeability measurements span over an order of magnitude around the $k_{eff}$ of the flaw area, except for one case that overestimates it by approximately three orders of magnitude. In all cases matrix permeability is $10^{-15}$ m$^2$. Notice that (a) and (b) exhibit maxima that differ by over four orders of magnitude.

have straight paths. This secondary dataset is depicted in Figure 5.24. From the geometrical point of view, straight cracks are less complex. Intersections occur only between coplanar fractures, shapes can be optimised due to their monotonicity, and polygonal extension is trivial. Both datasets are propagated until they reach similar density levels. As cracks propagate, fracture interaction inhibits growth of surrounding cracks, but does not trigger curving. We compare the flow properties of this dataset with the flow properties of our full-geomechanical model. We measure flow through the entire flaw area instead of the observation areas.

The straight path pattern exhibits clustering and alignment of proximal cracks in response to stress perturbations caused by local heterogeneities. Figure 5.25 shows the measured effective permeability values as a function of density. For the straight cracks, $k_{eff}$ trends for (a) high and (b) low permeability matrix increase up to an order of magnitude, from $10^{-12}$ m$^2$ to $10^{-11.9}$ m$^2$ and from $10^{-15}$ m$^2$ to $10^{-14.9}$ m$^2$ respectively, as fractures develop. The small fluctuations in the permeability are due to the variability of the aperture field at the onset of deformation.

Results indicate that for fixed apertures, the trend of $k_{eff}$ for curved and straight datasets is similar. Figures 5.26 (a) and (b) exhibit linear increase as a function of density. However, the straight model consistently measures lower $k_{effs}$ throughout.
Figure 5.24: Straight fractures. Fractures are grown suppressing the computation of the propagation angle. Straight fracture patterns at incremental growth stages, (a), (b), and (c), display the formation of preferentially aligned cracks due to clustering. Fracture set maturity is attained when growth focuses only on limiting fractures that transcend the flaw area.
Low \( k_m = 10^{-12} \text{m}^2 \)

High \( k_m = 10^{-15} \text{m}^2 \)

Figure 5.25: Flaw area straight fracture pattern: effective permeability. Plots of the decadic logarithm of the effective permeability of the “equivalent” flaw area for (a) low and (b) high matrix permeability. In both cases apertures are stress dependent. For smaller displacements effective permeability is consistently lower. In both cases, values fluctuate due to emerging aperture distributions and increase up to an order of magnitude.

For small apertures and high matrix permeabilities, the contribution of the straight cracks to flow is negligible. For stress-dependent apertures, the straight model underestimates \( k_{\text{eff}} \) as the difference becomes more pronounced. Due to the alignment of the fractures, many of the paths become occluded by interacting with neighbouring stress halos. Thus, apertures formed by the mechanical model reach higher maxima and conduct, in average, more flow. However, in this case, the difference between the permeabilities is less than an order of magnitude. Fracture-matrix flux ratio measurements exhibit a clearer difference between the flow capacity of the models. Notice that for the curved model, there is a peak in the \( k_{\text{eff}} \) and \( q_f/q_m \) measurements in all cases. This peak can also be appreciated in Figure 5.23. Examination of fracture geometry and topology shows that conductivity soars because of a peak in the aperture distribution. Fractures coalesce and form a larger permeability structure that conducts significantly more flow than in its previous disconnected constellation. However, the fracture pattern continues to develop partially closing the path that originated the stark increase in \( k_{\text{eff}} \) and \( q_f/q_m \).

5.6.7 Pattern anisotropy

The resulting crack patterns yield a strong anisotropy coherent to the applied boundary conditions. In this section, we quantify the anisotropy of the generated patterns as a function of fracture orientation and permeability. Orientation refers to
Figure 5.26: Curved versus straight: effective permeability. Decadic logarithm of effective permeability of the flaw area. The four graphs compare the conductivity between the curved pattern and its straight counterpart. Matrix permeability is high $k_m = 10^{-12}$ m$^2$ for (a) and (b), and low $k_m = 10^{-15}$ m$^2$ for (c) and (d). For (a) and (c) apertures are fixed, while for (b) and (d) they are stress-dependent. Notice that the $k_{eff}$ maxima of fixed and mechanical apertures are within the same order of magnitude, around one order of magnitude higher than the matrix permeability. Error bars denote the average accumulated numerical error for the fluid pressure field (see details in Section 2.4.2). The relative error ranges from $\sim 1 \times 10^{-3}$ to $\sim 0.06$. 
Figure 5.27: Curved versus straight: fracture-matrix flux ratio. Compares the fracture-matrix flux ratios between the flaw area and the “equivalent” straight pattern. Matrix permeability is high $k_m = 10^{-12} \text{m}^2$ for (a) and (b), and low $k_m = 10^{-15} \text{m}^2$ for (c) and (d). For (a) and (c) apertures are fixed, while for (b) and (d) they are stress-dependent. Error bars denote the average accumulated numerical error for the fluid pressure field (see details in Section 2.4.2). The relative error ranges from $\sim 1 \times 10^{-3}$ to $\sim 0.06$. 
the inclination of the segments that compose a fracture, while anisotropic permeability refers to the difference in hydraulic conductivity of the system in two different directions. Fractures are composed by a set of segments with local arbitrary orientations which are not necessarily representative of the overall orientation of the crack. We formulate a macroscopic crack orientation by weighting segment orientation with their length, as follows

\[ X_f = \frac{1}{w_f} \sum_{i=0}^{i<s} w_f^i \cos \chi_i \]  

(5.9)

where \( X_f \) is the macroscopic orientation of the crack, \( s \) are the segments of the crack centreline, \( w_f^i \) is the total centreline length, \( w_f^i \) is the length of segment \( i \), and \( \chi_i \) is the angle between segment \( i \) and the x axis. For a crack that only has straight segments \( X_f = 1 \), while for a completely vertical crack \( X_f = 0 \). For straight or planar cracks, the previous expression is reduced to the orientation of the centreline \( X_f = X_c \). Similarly, we can define an overall orientation of the fracture dataset

\[ X_{fs} = \frac{1}{w_{fs}} \sum_{f=0}^{f<fs} w_f^f X_f \]  

(5.10)

where \( f \) is a fracture in the fracture dataset \( fs \), \( w_f^f \) is its length, and \( w_{fs} \) is the combined length of all fractures. It follows that \( X_{fs} \) is a quantification of the fracture dataset orientation. In our datasets, \( X_{fs} \) decreases with growth ranging from 0.99 to 0.96 as density soars. Figure 5.28 plots the anisotropy measurements of all fractures as a function of increased density. It shows how the pattern remains highly anisotropic throughout growth. This is due to the pure tensile boundary conditions that induce fracture propagation. Figure 5.29 and 5.30 show the increase in \( k_{eff} \) measure in the vertical and horizontal direction as a function of growth. Superposed, the slight decrease of the fracture set anisotropy is caused by the curvature due to fracture interaction. We observe that effective permeability of the system increases rapidly in the direction of the anisotropy (horizontal) while remaining almost stagnant in the perpendicular direction.

### 5.6.8 Effective permeability estimation

The effective permeability of the system depends on an array of factors such as: the total size of the model, fracture volume, initial flaw distribution, alignment between the flaws and extension forces causing propagation, fracture set growth stage, permeability of the matrix, and aperture distribution. In particular, the density and
distribution of initial flaws combined with the matrix permeability represents, to some extent, the initial porosity of the system. In this section, we formulate an expression that estimates, based on the results and input data available in this study, the effective permeability of a given fractured system. This formula is intended to coarsely predict the hydraulic conductivity of the system by relating the different factors that influence flow in our simulations. It follows

\[
\log(k_{\text{eff}}) \approx c \log(k_f) + (1 - c) \log(k_m)
\]

where \(c\) is a constant that captures connectivity, \(k_m\) is the matrix permeability, and \(k_f = f(X_{fs}, \tau, \phi_i)\), where \(\tau\) is proportional to the fracture tortuosity, \(\phi_i\) is the initial flaw volume in the direction of the flow, and \(X_{fs}\) captures anisotropy (see Equation 5.10). Fracture permeability is defined in terms of a functional form \(k_f\), instead of fixing a certain value. For a fixed \(k_f\), \(k_{\text{eff}}\) is a combination of \(k_m\) and \(k_f\), weighted by the connectivity of the fractures, \(c\), a real number that varies between 0 and 1. Thus, it follows that \(\log(k_{\text{eff}})\) is bound by \(\log(k_f)\) and \(\log(k_m)\). It follows that if \(k_f \approx k_m\) then

\[
\log(k_{\text{eff}}) \approx \log(k_m) \approx \log(k_f)
\]
Figure 5.29: Pattern anisotropy: high permeable matrix. Horizontal permeability increases by 20%, whereas vertical permeability increase is 0.01%. As density increases the fracture dataset remains highly anisotropic, decreasing only by 3.5%.

If the dataset is fully interconnected and there is a percolating path, then $c \approx 1$. In this case
\[
\log(k_{eff}) \approx \log(k_f)
\]  
(5.13)

Else, if the dataset is poorly interconnected $c \approx 0$ and
\[
\log(k_{eff}) \approx \log(k_m)
\]  
(5.14)

This formula estimates the order of magnitude of the predicted hydraulic conductivity. It relates the conductivity, aperture, and anisotropy concepts to the overall effective permeability of the system. Equation 5.11 is empirically derived and serves the purpose of highlighting the dependence of the effective permeability on a variety of factors that combine the permeability of the matrix with the ability of the fractures to enhance flow.

## 5.7 Discussion

Our results quantify the conductivity of geomechanically generated discrete fracture patterns. As opposed to other studies, fractures that populate the models are generated as a function of a failure criterion, a propagation criterion, and a
Figure 5.30: Pattern anisotropy: low permeable matrix. Horizontal permeability increases by \(\sim 450\%\), whereas vertical permeability increase is around 1\%. As density increases the fracture dataset remains highly anisotropic, decreasing only by 3.5\%.

propagation angle. Using the FEM, flexible unstructured grids avoid lossy downsampling of topology and geometry. Arbitrary lines represent fractures, and their shapes are a part of the domain as an alternative to superposing flow contributions of the domain as in Baca et al. [1983]. Our findings show that fracture curving has a significant impact on the increase of conductivity. In our experiments, \(k_{\text{eff}}\) increases as a function of growth, and is strongly dependent on the aperture distribution of the fracture network. Fixed fracture apertures overestimate the model permeability by several orders of magnitude. As compared to the total conductivity of the system, local observation areas exhibit higher effective permeabilities and fracture-matrix ratios. However, for stress dependent apertures, observation area measurements are close to the total system conductivity.

Averaging fracture-only velocity fields, as in Nakashima et al. [2000], neglects flow through the matrix and considerably underestimates the \(k_{\text{eff}}\) of a fractured porous media [Bogdanov et al., 2003]. In contrast, we model flow through matrix and fractures simultaneously. Specifically, we compute a fluid pressure field, which results of applying a macroscopic pressure gradient to the sample. This captures flow properties more realistically as compared to approaches, such as the one presented by Kamath et al. [1998], which assume a constant fluid pressure field throughout the
model.

Other recent efforts to measure flow properties of geomechanically generated fracture datasets include work by Rijken [2005]. This two-step approach starts with the simulation of fracture growth using the BEM. It follows by computing flow properties using the finite difference method also assuming a constant fluid pressure distribution. The main disadvantage of using the finite difference method is that it relies on structured grids. Therefore, much of the detail of the geomechanically-generated fracture set must be sacrificed in order to produce a mesh that captures the essence of the connectivity of the fracture system.

The discrete fracture models presented here is not directly comparable to a binary permeability system because our aperture dependent fracture permeability definition. Due to the non-monotonous fracture conductivity, it is not straightforward to compare equivalent permeability computations with other available effective medium theory approximations.

5.7.1 Conductivity as a function of density

For the Bogdanov et al. [2007] experiment, the size of the smallest fracture controls the power-law length distribution. Therefore, density plays an important role in the increase or decrease of the overall conductivity of the system. They study the overall effect of density and percolation on the $k_{eff}$ of a set of randomly generated fractures with fixed apertures. The results presented in this work are a particular case of their set of experiments: the geometry is less dense, highly anisotropic, and geomechanically generated. In randomly generated datasets, density increases by adding more fractures to the dataset. This causes paths to link up and increases connectivity and density simultaneously. In geomechanically-generated datasets, density increases with the growth a finite set of fractures, instead of introducing new ones. Conductivity is a by-product of coalescing fractures and not a function of their amount. This enhances results by introducing behaviour resulting from the self-organisation properties of the network. Our results compare well: permeability increases almost linearly up to the percolation threshold. Then, it increases significantly to reflect the $k_{eff}$ of the dominant fracture path. In our case, permeability increase before percolation is slightly higher than as reported by Bogdanov et al. [2007]. However, their results assume that density must significantly increase in order to attain such a change. In our experiments, a minimal change in density causes a surge of the conductivity in the system.
5.7.2 Impact of matrix permeability and relationship to percolation theory

Matrix permeability plays an important role by indirectly connecting fracture clusters. The more permeable the matrix, the less flow is focused through the network, and thus, the less important variables such as fracture connectivity and aperture distribution become. The opposite is also true: the lower the matrix permeability the clearer the effect fractures exert over flow in the system. In the case where flow through the matrix is negligible, flow is a function of the connectivity of the system’s permeable structure: the fracture network, and therefore, a direct consequence of its connectivity. Percolation theory relates the increase of measurable parameters such as porosity and fracture density to the connectivity and conductivity of a system [King et al., 2001]. Once matrix permeability, $k_m$, reaches zero the permeability of the system is controlled by the conductivity of the fracture network. This is directly dependent on the aperture distribution along the fractures. For fixed apertures, and as $k_m$ decreases, fractures contribute more to the flow.

For naturally variable mechanical apertures, there is a characteristic transition $k_{mc}$ at which fractures cease to make a significant contribution to the flow. At this stage, the tendency of the system shifts from being dominated by matrix flow to being dominated by fracture flow. Thus, fractures significantly influence fluid flow above the $k_{mc}$ threshold. Additionally, the apertures of the fractures also play an important role in determining the influence of the cracks on the flow. For the same underlying topology, there are diverse connectivity structures that arise due to changes in the fracture apertures. In a sense, apertures affect the analogous percolation theory cluster structure. Thus, one can intuitively define a parameter aperture $a_c$ above which the system is well interconnected, and below which the system is poorly connected. Importantly, apertures are not constant. Instead, they are defined as a function of the mechanical state of the system. Thus, $a_c$ takes a functional form which we refer to as an aperture distribution. For $a_i$ and $a_j$ aperture distributions, $a_i > a_j$ if and only if for $a_i$ the fracture dataset is more conductive than for $a_j$. It follows that for an aperture distribution $a$: for datasets where $a > a_c$ a spanning fracture cluster will also be a percolating cluster that connects the left and right flow boundaries. Similarly, for $a < a_c$ the cluster does not percolate. Thus, the formulation of the problem, in our case, is quite different to the standard percolation theory scenario. Instead of having a set of bonds on a regular lattice or a regular set of fractures, we have a set of mechanically interacting polydisperse fractures with varying shapes. Additionally, instead of assigning probabilities to the
bonds of the system, connectivity is controlled by a variable aperture distribution along the fractures. Furthermore, the increase of matrix permeability is analogous to the increase in connected bonds in the percolation theory regular lattice. Thus, a higher $k_m$ implies higher connectivity and conductivity capacity of the porous matrix [Yanuka, 1992]. Figure 5.31 shows the increase in effective permeability as a function of decrease of matrix permeability. We observe a sigmoidal trend in the increase of the contribution of fractures to the total conductivity of the system as matrix permeability decreases. Fractures start to make a significant contribution when $k_m \sim 10^{-15} \text{m}^2$, peaking at $k_m \sim 10^{-19} \text{m}^2$. Thereafter, the effect of fractures on the total system remains almost constant. The four curves represent four increasing densities of the fracture dataset with mechanical apertures. We observe that this behaviour scales with density.

![Figure 5.31: Impact of matrix permeability. Sigmoidal trend of the increase in effective permeability of the system due to the fractures. Values are plotted as a function of decreasing matrix permeability. The four curves correspond to four different fracture datasets with mechanical apertures and increasing density.](image)

In our system, effective permeability not only depends on the existence of a connecting path as it is common in the percolation theory [Ronayne and Gorelick, 2006]. Instead, once a connecting path exists, there must also be a favourable aperture distribution that enables flow. Thus, the transition point at which the system starts to behave differently is a combination between the existence of a connecting path, and the tortuosity associated to the path. Figure 5.32 shows the contribution of the fractures to the total system flow as a function of increase in density. We observe,
as expected, that the maximum contribution of a fracture set on the flow scales with density. The average contribution scales to the approximate power of 0.42 as a function of density.

![Graph showing the relationship between density and permeability increase.](image)

Figure 5.32: Increase of effective permeability as a function of density. Square glyphs correspond to effective permeability increase due to increase in fracture density. White circles correspond to the average values per density. A power-law trendline is fitted through the average values. We measure an exponent of approximately 0.42. Points correspond to matrix permeabilities that range between $1 \times 10^{-12} \text{m}^2$ and $1 \times 10^{-22} \text{m}^2$.

### 5.7.3 Topology and overestimation of conductivity

Flow in fractures is hierarchical. The largest fractures channel the most flow. Nevertheless, flow through small connecting fractures and through the porous rock matrix plays an important role. Fractures are mechanical heterogeneities of the medium, which propagate and form intricate geometric networks. At initial development stages where fractures do not yet mechanically interact, they grow independently and the remote stress field solely controls their growth. Once fractures start to interact and re-accommodate, bigger ones start to grow at the expense of smaller ones. At this stage, local stress fields around each tip drive growth, and connectivity of the cracks forms a network that generates an intricate topological network of flow channels.
As geomechanically generated datasets develop, larger fractures experience acceleration in growth, while smaller ones decelerate or arrest. Furthermore, fractures with greater length have larger apertures, which enable them to channel more flow, faster. In the ideal case, all fractures are open and interconnected, all paths are unblocked, and fracture set geometry defines topology. In this case, it is the larger fractures that control system’s $k_{eff}$. However, permeability does not only depend on the topology of the fracture network. Studies show that the permeability contrast between fracture and matrix has less effect on flow focusing than fracture aspect ratio [Phillips, 1991]. Thus, it is not only important that fractures are connected, and that they form coherent paths, but these must also form channels of preferred flow with viable apertures. If one analyses the topology of the system by itself, and only observes the connectivity structure, disregarding geomechanically determined aperture distribution, the $k_{eff}$ of the system is likely to be overestimated. Results of our study indicate that networks with fixed apertures overestimate $k_{eff}$ by up to six orders of magnitude.

Fracture patterns, as they evolve, construct an underlying conductivity structure that gradually channels flow toward each other. In order to create a connective path only a small gap of matrix, that has not yet been fragmented, bridges pre-existing flow paths. The inherent permeability of the system is the one determined by the fracture network only, without taking into account aperture effects [Zhang et al., 2002]. In this case, underlying fracture topology defines fracture connectivity. We observe that the inherent permeability is consistently higher than the stress-state dependent $k_{eff}$. Zhang and Sanderson [2002] studied the effect of stress on the geomechanical aperture distribution of a fragmented medium and its consequences on its flow properties. Their two-stage approach relies on the generation of apertures using the discrete element approach, and computing the $k_{eff}$ using a dual permeability approach. In their datasets, an initial set of rotating blocks is the base to generate a fracture network of fluid pressure dependent aperture. Zhang and Sanderson [1996, 2002] do not report a jump in the $k_{eff}$ as fracture networks surpass percolation. Instead, in agreement with our findings, they observe a gradual increase in the conductivity of the system beyond this stage.

Due to the stark difference between fracture and matrix conductivity, for a low permeable matrix, once the preferred path is set, the rest of the fractures contribute little to the flow and their growth does not significantly influence the system. These results agree with experiments reported by Bogdanov et al. [2007]. Velocity plots confirm what is happening: flow channels through a preferred connected path. Figures
5.22 (a) and (b) illustrate how velocities around the connecting fractures increase as the fracture set develops. It starts out with velocities within the fractures that are slightly higher than in the matrix. Once fractures grow, they build networks supporting the incipient channelled flow. For constant apertures, and constant permeability along the fracture, the topology of the system clearly controls its $k_{eff}$. For a variety of fracture to matrix permeability ratios ranging from 5 up to 9 orders of magnitude (matrix permeability of $10^{-15} - 10^{-19}$ m$^2$) we obtain the similar behaviour. Interestingly, after the percolation threshold, all cases increase permeability up to values which reflect the permeability of the underlying fracture network. Thus, the amount by which the $k_{eff}$ increases at percolation is greater for the less permeable matrix. The more connections, the faster the flow, and the more permeable the specimen becomes.

5.8 Conclusions

We have compared $k_{eff}$ and $q_f/q_m$ measurements for fixed and stress dependent apertures for two observation areas of a geomechanically generated dataset. Results show that fixed apertures overpredict the $k_{eff}$ of the system by up to six orders of magnitude because they assume that topological connectivity implies flow connectivity. Thus, mono-permeability fractures significantly overestimate the conductivity of a porous medium. For stress dependent apertures, with variant fracture permeability, the preferred flow path is no longer determined by topology only. The straight path analogue of the dataset exhibits overall lower conductivity properties due to the lack of fracture connectivity.

During growth, connective structures build up gradually, and although fractures are not interconnected, they channel flow through preferred paths long before reaching percolation. Thus, $k_{eff}$ increases by an order of magnitude before there is any connecting path between flow boundaries. Percolation causes a jump in the $k_{eff}$ by several orders of magnitude when an open path connects the flow boundaries. After percolation, we observe a steady linear increase in the effective permeability of the system. For a high permeable matrix, significant flow occurs through the matrix at all stages. Channelling of flow is more pronounced for fixed aperture datasets as compared to geomechanical apertures.

Extensions of this work include the systematic study of geomechanically generated datasets to evaluate the influence of random seed placement on the fracture set characterisation after growth, $k_{eff}$ in the presence of multiple fracture sets, and
evaluation of the effects of damage and dissolution/precipitation processes on the permeability of the matrix during growth.
6 Discussion

A geometric-based kernel for fracture housekeeping during growth proves to be an accurate and efficient approach for the geomechanical simulation of fracture propagation. A flexible and independent discretisation of fracture and matrix domains is important because it yields the flexibility required to capture emergent behaviour. In mechanics, fractures may grow due to stress variations inducing changes in their shapes. This change in the geometry is handled by dynamic redefinition of the discretisation of the space. In flow, these emerging patterns include flow localisation and concentration fronts. Mesh adaptivity is instrumental to capture the local variability of the field.

Our results indicate that fracture geomechanical interaction and spatial self-organisation during growth significantly influence the flow properties of a fractured porous medium. They confirm that local stresses override remote boundary conditions. Our observations indicate that two-dimensional fracture growth algorithms are appropriate to model fracturing in layered media where interaction with neighbouring layers does not dominate the deformation. Curvature is an emergent phenomenon resulting from the stress halo overlap of fractures that affects flow properties of the system by enhancing hydraulic connectivity. Our findings suggest that fracture aperture distribution has a stronger effect on the conductivity of the system than its topology.

Crack curvature is a strong indicator of the stresses that formed it [Olson and Pollard, 1989]. For example, fractures which exhibit straight paths are more likely to have formed under large differential remote stress conditions [Olson, 1990]. These findings are supported by physical experiments involving the propagation of two cracks [Thomas and Pollard, 1993]. Comparison of multi-fracture patterns that arise under zero, intermediate, and large differential stresses show that local interactions override the effects of remote stresses inducing the formation of curved cracks in all cases. Other experiments show that acrylic coated plates subjected to large differential stress bending develop fracture patterns dominated by straight paths [Renshaw and Pollard, 1994b]. Our findings indicate that interaction with neighbouring layers dominates deformation and suppresses crack interaction effects. In nature, these phenomena mix and produce patterns that weigh the effects of fracture interaction, layer friction, and remote stress differential on the final pattern.
Additionally, our fluid flow experiments show that fluid flow properties exhibited by straight patterns strongly differ from their curved counterparts. Finally, our observations of increased hydraulic conductivity measurements for fixed aperture percolating fracture networks replicate the trends of stochastic based 3D results of Bogdanov et al. [2003] and Bogdanov et al. [2007].

Friction between rough fracture walls may induce the propagation of straighter paths, as observed by [Renshaw and Pollard, 1994a]. For larger fractures, the contact area between the walls becomes larger and the energy required to change the direction of propagation is probably higher than when they are are shorter. However, in order to weigh these effects against remote stress and friction with the over/underlying layers, we must incorporate these processes into the numerical deformation model and study them independently. The density and size of the initial flaw distribution of the dataset may also control flow properties of the system at a latter development stage. Ideally, the propagation algorithm should also incorporate crack nucleation modelling in order to enable fracture formation due to emerging strain accumulations. Fluid flow through the fractures also reduces the strength of the rock around the tip, causing hydraulic fracturing [e.g. Boone and Ingraffea, 1990]. This phenomenon has a significant effect on the increase of the permeability of a fractured system. Apertures measured by the mechanics-only simulations assume a constant fluid pressure in the fracture. These may not be accurate enough to represent the effects of high fluxes on the final aperture of the fractures. In fact, apertures in nature are emerging properties of the system that continuously change flow localisation by dynamically opening and closing paths.

The contribution of our work is threefold. We showed how complex emergent systems can be efficiently and dynamically discretised in order to capture solution field variability. We demonstrated a fracture propagation algorithm that is capable of creating discrete fracture systems based on an initial material description and a set of boundary conditions. Finally, we showed that these can be used to analyse fluid flow properties of fractured rock analogues in order to quantify the effects of aperture, topology, and curvature on the conductivity of the system. All the methods have been developed as practical tools and can be used to conduct similar studies involving mechanical and fluid flow effects.

This model has three main limitations. First, our methodology does not support the formation of disconnected matrix blocks. Their rotation and translation during deformation have a strong effect on the distribution of the apertures of the system, as shown by Zhang and Sanderson [2002]. The formation of these blocks can either be
prevented by disallowing self-intersection of fractures, or handled by disregarding the block in further computations. Ideally, the methodology should support the formation of independent domains, controlled by a hybrid discrete element formulation. Second, the geometric handling of fracture propagation, described in 2D, is significantly more complex in 3D. Fracture polylines become intricate polyhedra that must be propagated, intersected, closed, and merged. This implementation is significantly more resource consuming than its 2D counterpart. It requires more complex geometric algorithms and is considerably more memory and time consuming than in 2D due to the sharp increase in the amount of nodes. However, an alternative 2.5D implementation for the propagation of cracks in layered media would create well-formed extruded meshes in 3D, measure the stress distribution in the 3D model, and map it to a 2D representation of the crack. Third, this formulation is not appropriate to model deformation due to shear. The lack of cohesion between fracture walls constrains the applicability of this model to the formation of opening mode fractures only. A discrete element model is better suited in this case, as it can reproduce the damage localisation processes such as the formation of shear bands while also capturing fragmentation phenomena that accompany this deformation [Tillemans and Herrmann, 1995].

The extension of this model might incorporate: formation of discontinuous blocks by means of a hybrid finite element-discrete element formulation; deformation and propagation of fractures in 2.5D and 3D to capture the more realistic boundary conditions; handling of friction between fracture walls by incorporating intra-fracture stiff elements that regulate stress transduction; and, the incorporation of dissolution and precipitation effects due to stress concentration around the tips arising during deformation that influence the final permeability structure of the fractured porous medium.
7 Conclusions

Discrete crack models allow to study flow through matrix and fractures simultaneously. They capture growth, mechanical interaction, self-organisation, and emerging patterns. We perform FEM-based calculations on discretely represented cracks to measure the role that fractures play on fluid flow. By means of a hybrid FE-FV model we demonstrate that advection and diffusion equations can be numerically solved to compute pressure and saturation of geometric models within discrete fracture domains.

We presented a geomechanically consistent model that expresses the effects of fracture interaction on pattern formation and coalescence during growth. It captures fracture aperture variation taking into account the mutual influence of fractures on one another. Thereby, it generates new data for discrete fracture and matrix flow modelling. The crack propagation method is unique in that it employs dynamic meshing and keeps track of the fracture geometry in an independent manner. Fracture shapes are represented polygonally in a non-discretised form. Computational geometry techniques are applied to deal with fracture intersection and automatic tip tracking. This process is completely automatic and does not require user intervention during the simulation.

Aside from the geometric kernel, the method makes use of a failure criterion and a quasi-static propagation criterion. Failure and propagation criteria as well as material constitutive relations can be assigned flexibly and calibrated with laboratory experiments. Thus, it is not only applicable to sub-critical crack growth, but it also exhibits flexibility in terms of the crack tip stress functions that can be applied.

The generated crack patterns match fracture length and aperture statistics measured in field outcrops. Importantly, in contrast with stochastical methods, the spatial arrangement of the fractures is the result of their mechanical interaction during growth. This implies a physically meaningful fracture commutativity which is important for fluid flow simulation. Compared to analytical models of fracture growth, the ability of the numerical model to generate fracture patterns allows us to study their variability due to geologically realistic boundary conditions. Thus, relevant field data can be used to define the boundary conditions that control growth.

Geomechanically determined apertures influence the effect of fracture patterns on the overall effective permeability of a fractured porous medium. Fixed apertures
overpredict effective permeability by up to six orders of magnitude because they assume that topological connectivity implies flow connectivity. For mechanical apertures, with variant fracture permeability, the preferred flow path is not determined by topology only. Due to fracture linkage, effective permeability increases by an order of magnitude before there is any connecting path between flow boundaries. For geomechanical patterns, percolation is attained due to growth without significant increase in density or connectivity. Percolation only causes a jump in the conductivity when a path of constant aperture connects the flow boundaries. If this path has a variable, mechanically determined aperture, conductivity exhibits a linear increase. After percolation, we observe a steady linear increase in the effective permeability for all systems. For the high permeable matrix, significant flow occurs through the matrix at all stages. The effective permeability measured within an observation area is a good estimate of the overall permeability of the system for mechanically determined apertures.

7.1 Outlook

Extensions of this work include the development of a full 3D crack growth algorithm needed to correctly capture the effects of the thickness and mechanical layering. Other physical mechanisms to be integrated include: tractions between fracture walls, handling of disconnected blocks due to rock fragmentation during pattern development, and capturing of rock weakening at off-tip locations that induce independent failure. Discrete blocks will translate, rotate, and transduce stress. These must be captured by a kinetics-based physical model.

The 2D deformation simulation makes the assumption of plane stress or strain and does not incorporate notions of varying forces in the third dimension. An extension to 3D requires the geometric representation of fractures as three dimensional bodies. Thus, their centreline becomes a surface and their delimiting volume becomes a polyhedron. Boolean operations for intersection and merging as well as the geometric extension of the fracture body during growth must be formulated in 3D. Additionally, the imposed failure and propagation criteria shall be formulated and tested in 3D. As must be the orientation and extension of the new fracture surface plane. Finally, 3D remeshing requires handling of complex solids and automatic generation of volumetric meshes.

From the computational point of view, future work includes the following. The parallelisation of the fracture growth algorithm is imperative in order to handle larger
datasets and take advantage of the available computational resources. This implies subdividing the matrix and fractures into multiple domains. Including keeping track of dismembered fractures, handling of communication between the nodes, distributed remeshing, and mapping of the fields. Future work also includes the extension of the fracture object and geometric kernel into 3D. The next generation of the geometric kernel must handle 3D disconnected matrix regions that form during the fracturing process. In order to handle these moving independent fragments of rock, the geometric kernel must be extended to handle real-time three-dimensional collisions.
References


displacement-based stress intensity factor computation techniques. *International

adaptation algorithms with control of adaptivity. In *13th International Meshing

34:268287.

Structural Engineering and Materials*, 4:381399.


parameters in fractured reservoirs using percolation theory. In *14th Europec
Biennial Conference*, number SPE 94186.

equidistributed uniform pseudo-random number generator. *ACM Transactions on

Sealing and Fluid Flow in Hydrocarbon Reservoirs*, chapter Numerical simulation
of departures from radial drawdown in a faulted sandstone reservoir with joints and


csp3d3.0: Users guide. Technical report, ETH Zürich Research Reports.

Matthai, S. K., Geiger, S., Roberts, S. G., Paluszny, A., Belayneh, M., Burri, A.,
*Structurally Complex Reservoirs*, chapter Numerical simulation of multi-phase fluid


188


Appendix
A Hybrid discretisation details

Figure A.1: Face numbering. Face and node numbering for the isoparametric linear elements used in the hybrid element discretisation. Face numbers are underlined.

<table>
<thead>
<tr>
<th>Element</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isoparametric Linear Bar</td>
<td>1(-1); 2(1)</td>
</tr>
<tr>
<td>Isoparametric Linear Triangle</td>
<td>1(0,0); 2(1,0); 3(0,1)</td>
</tr>
<tr>
<td>Isoparametric Linear Quadrilateral</td>
<td>1(-1,-1); 2(1,-1); 3(1,1); 4(1,1)</td>
</tr>
<tr>
<td>Isoparametric Linear Tetrahedron</td>
<td>1(0,0,0); 2(1,0,0); 3(0,1,0); 4(0,0,1)</td>
</tr>
<tr>
<td>Isoparametric Linear Prism</td>
<td>1(0,0,0); 2(1,0,0); 3(0,1,0); 4(0,0,1); 5(1,0,1); 6(0,1,1)</td>
</tr>
<tr>
<td>Isoparametric Linear Pyramid</td>
<td>1(-1,-1,-1); 2(-1,-1,-1); 3(1,1,-1); 4(-1,1,-1); 5(-1,1,1); 6(1,1,1); 7(1,1,1); 8(-1,1,1)</td>
</tr>
</tbody>
</table>

Figure A.2: Node Coordinates in Parametric Space. Formatting follows the scheme number of node, followed by its coordinates in parametric space.
<table>
<thead>
<tr>
<th>Element</th>
<th>Node</th>
<th>Shape Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isoparametric Linear Bar</td>
<td>$N_0$</td>
<td>$\frac{1-r}{1+r}$</td>
</tr>
<tr>
<td></td>
<td>$N_1$</td>
<td>$\frac{1}{1+r}$</td>
</tr>
<tr>
<td>Isoparametric Linear Triangle</td>
<td>$N_0$</td>
<td>$1-r-s$</td>
</tr>
<tr>
<td></td>
<td>$N_1$</td>
<td>$r$</td>
</tr>
<tr>
<td></td>
<td>$N_2$</td>
<td>$s$</td>
</tr>
<tr>
<td>Isoparametric Linear Quadrilateral</td>
<td>$N_0$</td>
<td>$\frac{1}{2}(1-r)(1-s)$</td>
</tr>
<tr>
<td></td>
<td>$N_1$</td>
<td>$\frac{1}{2}(1 + r)(1-s)$</td>
</tr>
<tr>
<td></td>
<td>$N_2$</td>
<td>$\frac{1}{2}(1 + r)(1+s)$</td>
</tr>
<tr>
<td></td>
<td>$N_3$</td>
<td>$\frac{1}{2}(1-r)(1+s)$</td>
</tr>
<tr>
<td>Isoparametric Linear Tetrahedron</td>
<td>$N_0$</td>
<td>$1-r-s-t$</td>
</tr>
<tr>
<td></td>
<td>$N_1$</td>
<td>$r$</td>
</tr>
<tr>
<td></td>
<td>$N_2$</td>
<td>$s$</td>
</tr>
<tr>
<td></td>
<td>$N_3$</td>
<td>$t$</td>
</tr>
<tr>
<td>Isoparametric Linear Prism</td>
<td>$N_0$</td>
<td>$\frac{(1-r-s)(1-t)}{2(1-t)}$</td>
</tr>
<tr>
<td></td>
<td>$N_1$</td>
<td>$\frac{1}{2}(1 + r)(1-s)(1-t)$</td>
</tr>
<tr>
<td></td>
<td>$N_2$</td>
<td>$\frac{1}{2}(1 + r)(1+s)(1-t)$</td>
</tr>
<tr>
<td></td>
<td>$N_3$</td>
<td>$\frac{(1-r)(1+s)(1-t)}{2(1-t)}$</td>
</tr>
<tr>
<td></td>
<td>$N_4$</td>
<td>$\frac{1}{2}(1 + r)(1+s)(1+t)$</td>
</tr>
<tr>
<td></td>
<td>$N_5$</td>
<td>$\frac{(1-r)(1+s)(1+t)}{2(1+t)}$</td>
</tr>
<tr>
<td>Isoparametric Linear Pyramid</td>
<td>$N_0$</td>
<td>$\frac{1}{2}((1 - r)(1 - s) - t + \frac{rst}{(1-t)})$</td>
</tr>
<tr>
<td></td>
<td>$N_1$</td>
<td>$\frac{1}{2}((1 + r)(1 - s) - t + \frac{rst}{(1-t)})$</td>
</tr>
<tr>
<td></td>
<td>$N_2$</td>
<td>$\frac{1}{2}((1 + r)(1 + s) - t + \frac{rst}{(1-t)})$</td>
</tr>
<tr>
<td></td>
<td>$N_3$</td>
<td>$\frac{1}{2}((1 - r)(1 + s) - t + \frac{rst}{(1-t)})$</td>
</tr>
<tr>
<td></td>
<td>$N_4$</td>
<td>$t$</td>
</tr>
<tr>
<td>Isoparametric Linear Hexahedron</td>
<td>$N_0$</td>
<td>$\frac{1}{2}((1 - r)(1 - s)(1 - t))$</td>
</tr>
<tr>
<td></td>
<td>$N_1$</td>
<td>$\frac{1}{2}((1 + r)(1 - s)(1 - t))$</td>
</tr>
<tr>
<td></td>
<td>$N_2$</td>
<td>$\frac{1}{2}((1 + r)(1 + s)(1 - t))$</td>
</tr>
<tr>
<td></td>
<td>$N_3$</td>
<td>$\frac{1}{2}((1 - r)(1 + s)(1 - t))$</td>
</tr>
<tr>
<td></td>
<td>$N_4$</td>
<td>$\frac{1}{2}((1 - r)(1 + s)(1 + t))$</td>
</tr>
<tr>
<td></td>
<td>$N_5$</td>
<td>$\frac{1}{2}((1 + r)(1 - s)(1 + t))$</td>
</tr>
<tr>
<td></td>
<td>$N_6$</td>
<td>$\frac{1}{2}((1 + r)(1 + s)(1 + t))$</td>
</tr>
<tr>
<td></td>
<td>$N_7$</td>
<td>$\frac{1}{2}((1 - r)(1 + s)(1 + t))$</td>
</tr>
</tbody>
</table>

Figure A.3: Linear shape functions.
<table>
<thead>
<tr>
<th>Element</th>
<th>[rst]</th>
<th>Shape Function Derivatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isoparametric Linear Bar</td>
<td></td>
<td>( -\frac{1}{2}, \frac{1}{2} )</td>
</tr>
<tr>
<td>Isoparametric Linear Triangle</td>
<td></td>
<td>( (-1,0,1) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( (-1,1,0) )</td>
</tr>
<tr>
<td>Isoparametric Linear Quadrilateral</td>
<td></td>
<td>( (-\frac{1}{2}, \frac{1}{2}, r, -\frac{1}{2}, \frac{1}{2}, s, \frac{1}{2}, \frac{1}{2}, s, -\frac{1}{2} - \frac{1}{2} s) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( (-1,0,0,1) )</td>
</tr>
<tr>
<td>Isoparametric Linear Tetrahedron</td>
<td></td>
<td>( (-1,1,0,0) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( (-1,0,1,0) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( (-1,0,0,1) )</td>
</tr>
<tr>
<td>Isoparametric Linear Prism</td>
<td></td>
<td>( \left( \frac{1}{2} (-1 + t), \frac{1}{2} (1 - t), 0, \frac{1}{2} (-1 - t), 0 \right) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \frac{1}{2} (-1 + t), 0, \frac{1}{2} (1 - t), 0, \frac{1}{2} + \frac{1}{2} t )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \frac{1}{2} (-1 + r + s), - \frac{1}{2}, - \frac{1}{2}, (1 - r - s), \frac{1}{2}, \frac{1}{2} )</td>
</tr>
<tr>
<td>Isoparametric Linear Pyramid</td>
<td></td>
<td>( \left( \frac{1}{2} (-1 + s + \frac{rt}{1-s}), \frac{1}{2} (1 - s - \frac{rt}{1-s}), \frac{1}{2} (1 + s + \frac{rt}{1-s}), \frac{1}{2} (1 - r - \frac{st}{1-t}), 0 \right) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \left( \frac{1}{2} (-1 + r + \frac{st}{1-t}), \frac{1}{2} (1 - r - \frac{st}{1-t}), \frac{1}{2} (1 + r + \frac{st}{1-t}), \frac{1}{2} (1 - s - \frac{rt}{1-t}), 0 \right) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \left( \frac{1}{2} (-1 + \frac{rs}{1-s} + \frac{rs}{1-t}), \frac{1}{2} (-1 - \frac{rs}{1-s} - \frac{rs}{1-t}), \frac{1}{2} (-1 + \frac{rs}{1-s} + \frac{rs}{1-t}), \frac{1}{2} (1 - \frac{rs}{1-s} - \frac{rs}{1-t}) \right) )</td>
</tr>
<tr>
<td>Isoparametric Linear Hexahedron</td>
<td></td>
<td>( \frac{dN_r}{dr} \left( \begin{array}{c} -\frac{1}{2} (1 - s)(1 - t) \ \frac{1}{2} (1 - s)(1 - t) \ -\frac{1}{2} (1 - s)(1 - t) \ -\frac{1}{2} (1 - s)(1 - t) \end{array} \right) = \frac{dN_s}{ds} \left( \begin{array}{c} -\frac{1}{2} (1 - r)(1 - t) \ -\frac{1}{2} (1 - r)(1 - t) \ -\frac{1}{2} (1 - r)(1 - t) \ -\frac{1}{2} (1 - r)(1 - t) \end{array} \right) = \frac{dN_t}{dt} \left( \begin{array}{c} -\frac{1}{2} (1 - r)(1 - s) \ -\frac{1}{2} (1 - r)(1 - s) \ -\frac{1}{2} (1 - r)(1 - s) \ -\frac{1}{2} (1 - r)(1 - s) \end{array} \right) )</td>
</tr>
</tbody>
</table>

**Figure A.4:** Linear shape function derivatives.
B CSMP++ mechanics module: user’s guide

All algorithms explained in this work have been integrated into the CSMP++ simulation Application Interface Tool (API). The mechanics module has three main use cases:

Propagate fractures  Grows a set of fractures and outputs the geometry at each iteration step. This geometry can later be analysed with the fracture pattern characterisation module or used as an input for a single-phase flow experiment. It takes as an input a variable file (see Table B.1) and a generic configuration file (see Table B.3). This routine collects information about timing. The input file names are “mechanics-2D-configuration.txt” and “CSP deformation variables.txt”. The output are

- *simulation_info.log* Contains information about the simulation progress. Amount of nodes, time, time-stamp of last iteration. It is ideal to check the progress of long or remote runs.
- *time_compute_stress_accumulate.log* Stores the time taken to accumulate the integrals for the simulation.
- *time_compute_stress_move.log* Stores the time to move the coordinates.
- *time_compute_stress_solve.log* Stores the time to solve the integrals using the SAMG.
- *time_compute_stress_ss.log* Stores the time to compute stresses and strains.
- *time_for_output.log* Stores the time for file output.
- *time_for_propagation.log* Stores the total propagation time.
- *time_to_compute_stress.log* Stores the total time to compute stress.
- *time_to_create_flaws.log* Stores the time required to generate the initial random flaw distribution.
• *time_to_create_sg.log* Stores the time to create the SuperGroup and other data structures.

• *time_to_extract_shapes.log* Stores the time to extract shapes.

• *nodes.log* Stores the amount of nodes that were generated.

• *fracture_vsetN.000000.vset* Current stress state at iteration N.

• *fracture_set_N.txt* Fracture set at iteration N.

• *after_update_lengthN.txt* Rhino command fracture set at iteration N. Copy and paste the contents of this file to the command line to debug the fracture geometry.

• *matrix_displacementN.vtk* VTK visualisation of the displacement field (vectors) at iteration N.

• *matrix_stressN.vtk* VTK visualisation of the stress field (tensors) at iteration N.

• *matrix_nmean_stressN.vtk* VTK visualisation of the nodal mean stress field (scalars) at iteration N. Suitable to create contours.

• *matrix_emean_stressN.vtk* VTK visualisation of the element mean stress field (scalars) at iteration N.

• *Fracture_Lines_N.vtk* VTK visualisation of the fracture dataset (scalars) at iteration N.

The usage is via the command line. It currently supports the following unary flags:

• *-straight (default:off)* Propagates straight fractures only.

• *-shrink (default:off)* Applies isotropic shrinkage boundary conditions.

• *-LargeDifferential (default:on)* Applies large differential stress boundary conditions.

• *-IntermDifferential (default:off)* Applies intermediate differential stress boundary conditions.

• *-ZeroDifferential (default:off)* Applies zero differential stress boundary conditions.
- **-LargeDifferentialHorizontal (default: off)** Applies large differential stress boundary conditions in the horizontal direction (to generate vertical cracks).

- **-printoff (default: on)** Turns all printing off.

Warning: If more than one boundary condition is specified, only the last one will be taken into account. The executable also provides binary input. This refers to the combination of a flag with another token, such as a string or a number. The following is a list of the available binary flags:

- **-s (default: 10)** Specifies a scale for the meshing.

- **-i (default: 0)** Specifies first iteration step. If zero it generates a new FlawSet as the first step. If 1 or larger it attempts to load the dataset and the corresponding stress file (e.g. for -i 2 it expects to find in the local directory: “fracture_set2.txt” and “fracture_vset2.000000.vset”).

- **-n or -imax (default: 100)** Specifies a last iteration. It is when the simulation stops.

- **-d (default: 2.0 \times 10^{-5})** Assigns a displacement for the boundaries.

- **-w (default: 2)** Specifies a width of the model.

- **-h (default: 2)** Specifies a height of the model.

- **-f (default: 50)** Specifies a number of initial fractures.

- **-b (default: off)** Specifies the initial flaw angle. If not called, fractures are generated with random orientations, else, it assigns an fixed initial angle to all of them.

- **-a (default: 1.0 \times 10^{-5})** Initial flaw aperture.

- **-z (default: 1.0 \times 10^{-2})** Initial flaw mean size. Standard deviation is set to \((mean/2)^2\).

- **-sp (default: 3)** Initial spacing between flaws. Defined relative to the fracture mean size.

- **-g (default: 0.35)** Growth index.

- **-p (default: 0)** Perturb material properties. Followed by a percentage.

- **-adv (default:)** Value of the largest tip advance per iteration.
Conductivity analysis of the generated pattern Measures effective permeability and fracture-matrix flux ratio of the flaw area, and observation areas. Data for observation areas is hard-coded. Works best for datasets of dimensions 1m × 4m. Can be adjusted in the code for other geometries by changing the main file. It takes as an input a hybrid fluid and mechanics variable file (see Table B.1 and B.2) and a generic configuration file (see Table B.4). This programme operates in two modes: it either loads a pre-existing Ansys ICEM mesh or generates one. Fractures are represented as segments or volumetric entities. Computes the fluid pressure error, aperture distribution statistics, effective permeability, fracture-matrix flux ratio, and stream-lines (for volumetric cracks only). All statistics are output in the form of *.log files. When the simulation reaches the maximum iteration a folder is created and all results are moved there. This allows to run multiple consecutive simulations without having to manually move files. The input file names are “mechanics-2D-configuration.txt”, “CSP_hybrid_variables.txt”, “CSP_deformation_variables.txt”, and ”fluid_flow-configuration.txt”. The output are the same as in the previous executable plus:
'mechanics-configuration.txt' standard mechanics file

# assigning general material properties to the model
displacement 0. 0.
force 0. 0.
dilatation 0.
Neumann stress 0. 0.
mean stress 0.
stress 0. 0. 0. 0.
strain 0. 0. 0. 0.
sigma_1 0.
# density * g (-9.80665) acting opposite Y-axis
gravity force 0. -26477.955

# property assignment to model subregions
matrix complete Young’s modulus 20e+9
matrix complete Poisson’s ratio 0.2

Figure B.3: Mechanics configuration file.

- **effective_k_info.log** Stores the effective permeabilities.
- **effective_k_short.log** Stores the effective permeabilities in short format ready to import to MS Excel.
- **fm_ratio.log** Stores the fracture-matrix flux ratios and equivalent matrix permeabilities.
- **center_observation_area_N.vtk** VTK visualisation of the entire dataset at iteration N.
- **fluid-pressureN.vtk** VTK visualisation of fluid pressure (scalar) at iteration N.
- **frac-apertureN.vtk** VTK visualisation of apertures (scalar) at iteration N.
- **FRACTURES_node_frac-apertureN.vtk** VTK visualisation of nodal fracture apertures (scalar) at iteration N.
- **FRACTURES/MATRIX-frac-apertureN.log** histogramme data of aperture distribution.
- **INITY_permeabilityN.vtk** VTK visualisation of initial permeability (scalar) at iteration N.
'hybrid-configuration.txt' generic hybrid configuration

#assigning material properties to the overall model
#porosity 0.25
fluid volume source 0.0
fluid pressure 0.
#MECHANICS VARIABLE INIT
displacement 0.0 0.0
force 0.0 0.0
dilatation 0.0
Neumann stress 0.0 0.0
mean stress 0.
stress 0. 0. 0. 0.
strain 0. 0. 0. 0.
stress-y 0.
stress-xy 0.
principal stress 0. 0.
principal strain 0. 0.
# density * g (-9.80665) acting opposite Y-axis
gavity force 0.0 -26477.955
# fluid flow-related properties
permeability 1.e-15
dummy_face 0
porosity 0.15
storativity 1.0e-9
fluid density 1.e3

# property assignment to model subregions (complete, interior, boundary)
FRACTURES complete porosity 1.0
FRACTURES complete permeability 1.e-10
MATRIX complete permeability 1.e-15
MATRIX complete Young's modulus 20e+9
MATRIX complete Poisson's ratio 0.2
MATRIX complete permeability 1.e-15

# essential conditions on model boundaries
LEFT Dirichlet fluid pressure 1.0e7 1.0e7
RIGHT Dirichlet fluid pressure 1.0e6 1.0e6

Figure B.4: Hybrid configuration file. Defines mechanics and flow initial values.
• meshN.vtk VTK visualisation of the mesh (scalar) at iteration N.
• permeabilityN.vtk VTK visualisation of permeability (scalar) at iteration N.
• pore velocityN.vtk VTK visualisation of the pore velocity (vector) at iteration N.
• velocityN.vtk VTK visualisation of the velocity (vector) at iteration N.
• volume-fluxN.vtk VTK visualisation of the volume flux (scalar) at iteration N.

The following is a list of the available unary flags:

• -fixed Specifies fixed apertures.
• -mech Specifies mechanical apertures.
• -keff_off Turns off effective permeability computation.
• -error_on Turns on discretisation error computation.

The following is a list of the available binary flags:

• -i (default:0) Specifies first iteration step. It attempts to load the dataset and the corresponding stress file (e.g. for -i 2 it expects to find in the local directory: “fracture_set2.txt”, and “fracture_vset2.000000.vset” if i is larger than zero).
• -a Apertures for fixed, and displacement for mechanics computations.
• -km Matrix permeability.
• -f File root. Required if Ansys ICEM meshes are being loaded. For example: “flaw_area”, “straight”, “deform14_obsA”).
• -n or -imax (default:100) Specifies a last iteration. It is when the simulation stops. Moves all *.vtk and *.log files to specific directories such as: “hg_mech_aperture_1e-12_0.01_p0”, “hg_fixed_aperture_1e-15_0.001_p0”.

205
Fracture pattern characterisation Analyses fracture patterns by measuring connectivity, density, spacing, among others. It only requires the fracture set file, “fracture_setN.txt”, as an input. It does not take any command line parameters. It is tailored to work for observation areas A and B, but can be adjusted to other geometries. The output files are the following:

- *densities_total.log* Total density.
- *densities_flaw_area.log* Density of the flaw area.
- *densitiesA.log* Density of observation area A.
- *densitiesB.log* Density of observation area B.
- *number_of_fractures.log* Total number of fractures.
- *connectivity.log* Connectivity.
- *spacing.log* Spacing.
- *length_distribution_mean.log* Length means at each growth step.
- *length_distribution_stddev.log* Length standard deviation at each growth step.
- *lengths.log* Complete length data of all fractures.
- *max_subfractures.log* Maximum number of sub-fractures. Equivalent to maximum cluster size.
- *avg_subfractures.log* Average number of sub-fractures. Equivalent to average cluster size.
- *max_extension.log* Maximum fracture extension.
- *centerline_lengths.log* Centreline lengths of all fractures.
C  CSMP++ mechanics module: implementation details

This chapter is only available in electronic format.