Imperial College London

Coupled deformation, fluid flow and fracture propagation in porous media

Anthony Raphael Lamb
Department of Earth Science and Engineering
Imperial College London

A thesis submitted in partial fulfilment of the requirements
for the degree of
Doctor of Philosophy
May 2011

To my wife, Deborah, an engineer and mathematician whom I greatly respect and admire, and to our little sunshine, Isabelle.

"For the Lord giveth wisdom; out of His mouth cometh knowledge and understanding"

Proverbs 2:6

Declaration

I declare that the work within this thesis is my own and all other work that is not my own has been appropriately acknowledged.

Anthony R. Lamb

Abstract

Polygonal faults are non-tectonic fault systems which are layer-bound (at some vertical scale) and are widely developed in fine-grained sedimentary basins. Although several qualitative mechanisms have been hypothesised to explain the formation of these faults, there is a weak general consensus that they are formed by the coupled deformation and fluid expulsion of the host sediments (consolidation). This thesis presents a numerical framework that can be extended to investigate the role consolidation plays in the development and evolution of these faults.

The method is also applicable to reservoir engineering and CO_2 storage. An understanding of the coupled mechanical response and fluid flow is critical in determining compaction and subsidence in oil reservoirs and fault-seal integrity during CO_2 disposal and storage.

The technique uses a fracture mapping approach (FM) and the extended finite element method (XFEM) to modify the single phase FEM consolidation formulation. A key feature of FM-XFEM is its ability to include discontinuities into a model independently of the computational mesh.

The fracture mapping approach is used to simulate the flow interaction between the matrix and existing fractures via a transfer function. Since fractures are represented using level set data, the need for complex meshing to describe fractures is not required.

The XFEM component of the method simulates the influence of the pore fluid on the mechanical behaviour of the fractured medium. In XFEM,

enrichment functions are added to the standard finite element approximation to ensure an accurate approximation of discontinuous fields within the simulation domain.

FM-XFEM produces results comparative to the discrete fracture method on relatively coarse meshes. FM-XFEM has also been extended to model the propagation of existing fractures using a mixed-mode criterion applicable to geological media. Stress concentrations at the tips of existing fractures show good agreement with an analytical solution found in literature.

Publications

Lamb, A. R., Gorman, G. J. & Elsworth, D. Coupled deformation and fluid flow in fractured porous media using a fracture mapping approach and the extended finite element method. Under Review

Conference Proceedings

Lamb, A. R., Gorman, G. J., Gosselin, O. R. & Onaisi, A. 2010. Coupled deformation and fluid flow in fractured porous media using dual permeability and explicitly defined fracture geometry. SPE EUROPEC/EAGE *Annual Conference and Exhibition.* Barcelona, Spain, SPE 131725-MS: Society of Petroleum Engineers.

Acknowledgements

I would firstly like to thank Stephan Matthäi, now at Montan University of Leoben, Austria, and Laure Moen-Maurel of Total for inviting me to Imperial College to complete this PhD. The financial support provided by Total Geoscience Research Centre (GRC) is also acknowledged.

Gerard Gorman and Martin Blunt are thanked for taking over the supervision of this project upon Stephan's departure from Imperial. Special thanks to my PhD steering committee; Olivier Gosselin, Atef Onaisi and Bertrand Gauthier. I'd also like to thank the members of the CSMP++ group Evren Unsal, Murtaza Gulamali, Lateef Akanji, Adriana Paluszny, Martin Iding and Hamid M-Nick for helpful discussions on fluid flow in porous media and fracture propagation. Mandefro Belayneh is also thanked for helpful chats on structural geology.

I am deeply indebted to Stéphane Bordas and many other members of his computational mechanics discussion group on Google. Stéphane along with Marc Duflot, Elisa Budyn, Natarajan Sukumar were no further than an email away for any queries that came up regarding the implementation of the extended finite element method. I would also like to thank Gui-Rong Liu for making time in his busy schedule to discuss the use of the radial point interpolation method and other meshfree methods he and has group have been developing over the last couple of years. A special thanks to Rafid Al-Khoury for helpful discussions during the initial development of the numerical scheme presented in this thesis. I also need to acknowledge Derek Elsworth for his help

and advice. Although Derek has a very hectic schedule he was always available for friendly chats about my work via Skype. I also need to extend thanks to Matthew Piggott and Garth Wells for agreeing to be my examiners and making invaluable comments which have greatly improved the quality of this document.

I would like to thank my wife Deborah for her unwavering support over the last three years. She was always there to bounce my ideas off and to give me a quick kick in the pants whenever I appeared to be slacking off. Our ten month old daughter Isabelle provided a pleasant distraction whenever the write-up seemed to be a bit too much. That infectious giggle always brightens up my world. Thanks Izzy!!

CONTENTS

Declaration **4**

Abstract **5**

Acknowledgements **7**

1 Introduction **19**

1.1 Scope of Thesis 23
1.2 The Numerical Model 24
1.3 Implementation of the Numerical Model 29
1.4 Thesis Layout 31
1.5 Notation 32
1.6 Thesis Contributions 32

2 Consolidation Theory and its Finite Element Formulation **34**

2.1 Consolidation Theory 34
2.2 Governing Equations of Consolidation Theory 35
2.2.1 Initial and Boundary conditions 38
2.3 Numerical Simulation 39
2.4 Spatial Discretization 39
2.5 Temporal Discretization 43
2.5.1 Absolute Load 44
2.5.2 Incremental Load 44
2.6 Consolidation Code Verification 45
2.6.1 One-dimensional Consolidation 46
2.6.2 Two-dimensional Consolidation 50
2.7 Conclusion 56

3 Fracture Mapping 57

3.1 Introduction 57
3.2 Modern approaches to fracture flow modelling 60
3.3 Transfer Function Mapping 64
3.4 Fracture Representation 67
3.4.1 Assignment of fractured elements 69
3.5 Governing Equations of Single Phase Fluid Flow in Porous Media 70
3.5.1 Boundary conditions 72
3.6 Spatial Discretization 72
3.7 FEM Implementation 73
3.8 Meshfree Implementation 74
3.9 Numerical Examples 79
3.9.1 One well intersecting a vertical fracture 79
3.9.2 Discrete-fracture model comparison 85
3.9.3 Intersecting Fractures 93
3.10 Conclusions 97

4 FM-XFEM formulation 98

4.1 Modelling Discontinuities 98
4.1.1 Intersecting Fractures 102
4.2 Numerical Integration 105
4.3 Fracture Closure 106
4.3.1 Penalty Method Implementation 110
4.4 Spatial Discretization 112
4.5 Temporal Discretization 114
4.6 FM/XFEM Implementation 114
4.7 Numerical Examples 116
4.7.1 Contact Algorithm Example 116
4.7.2 Coupled Numerical Example 119
4.7.3 Intersecting Fractures Example 126

4.8 Conclusions 128

5 Fracture Propagation 129

5.1 Stress Intensity Factor Calculation 129
5.2 Fracture Propagation 133
5.2.1 Fracture Extension Length 134
5.3 Fracture Propagation Algorithm 135
5.4 Numerical Examples 137
5.4.1 SIF Example 137
5.4.2 Fracture Interaction 142
5.4.3 Fracture Extension Example 144
5.5 Conclusion 145

6 Conclusions and Future Work 146

6.1 Conclusions 146
6.2 Future work 148
6.2.1 Extension to Multiphase Flow 148
6.2.2 Extension to 3D 149
6.2.3 Meshfree Implementation 150

Appendix A 151

Appendix B 158

Appendix C 164

Appendix D 169

Appendix E 172

References 180

LIST OF FIGURES

Fig. 1.1. Schematic diagram of a polygonal fault systems; (a) 2D and (b) 3D....20

Fig. 1.2. Horizontal cross-section of Upper Oligocene in Block 16/26 of the UK sector of the North Sea showing polygonal faults. (taken from Cartwright and Lonergan, (1996).....21

Fig. 2.1. Element combination used for interpolation of the coupled solid-fluid interaction (a) Quadratic elements are used for displacement, (b) Linear elements are used for pressure. Displacements are calculated at element vertices and mid-side nodes of the quadratic elements. 41

Fig. 2.2. Illustration of Q9/Q4 finite element pair. 42

Fig. 2.3. Domain used for one-dimensional consolidation example.....48

Fig. 2.4. Surface settlement variation with time.....49

Fig. 2.5. Excess pore pressure variation with time at various locations in the domain (see Fig. 2.3).....49

Fig. 2.6. 3D-axisymmetric problem reduced to a 2D problem. 52

Fig. 2.7. Varying finite element meshes used for numerical simulations. 52

Fig. 2.8. Excess pore pressure at the centre of Cryer's consolidation ball vs dimensionless time. 53

Fig. 2.9. Surface displacement of Cryer's consolidation ball vs dimensionless time. 54

Fig. 2.10. Rate of convergence of implemented finite element code for Cryer's consolidation ball. 56

Fig. 3.1. Schematic representation of the matrix and fracture elements/nodes extracted from discretized fractured domain. 65
Fig. 3.2. Schematic representation of fracture mapping for intersecting fractures. 66
Fig. 3.3. Fracture permeability is aligned with the Cartesian axes using the rotation tensor **R.** 67
Fig. 3.4. Level set functions for 2D fracture representation. 69
Fig. 3.5. Assignment of fractured elements using level set values. 70
Fig. 3.6. Domain discretization (a) FEM (b) MFree 77
Fig. 3.7. Voronoi cells used for nodal integration. 78
Fig. 3.8. Selection of supporting nodes required for shape function calculations. 79
Fig. 3.9. Closed rectangular reservoir containing a well intersected by a vertical fracture. Due to domain symmetry only one-quarter of the domain is discretized for the numerical simulation. This region is represented by the hatched area. 80
Fig. 3.10. Domain discretizations used for the closed rectangular reservoir. Given the domain symmetry, only one-quarter of the domain has been discretized. 82
Fig. 3.11. Dimensionless well drawdown vs dimensionless time, showing comparison of numerical and analytical solutions. (a) FEM comparison, (b) meshfree comparison. 83
Fig. 3.12. 2D fractured domain with applied pressures at top and bottom boundary. 85

Fig. 3.13. Elements used for discrete fracture model. (a) Triangular element (T3) used to discretize the matrix and (b) line element (L2) used to discretize the fracture.....86
Fig. 3.14. Mesh refinements used to determine rate of convergence of DFM. ..86
Fig. 3.15. Domain discretizations used to determine the rates of convergence of the FEM and Meshfree fracture mapping implementation. 88
Fig. 3.16. Convergence rates for the discrete fracture model and the fracture mapping approach. The fracture mapping approach has been implemented using FEM and the Meshfree method for both regular and irregular nodal distributions.....89
Fig. 3.17. Domain discretizations used to create comparative pressure fields...90
Fig. 3.18. Pressure fields obtained for DFM and fracture mapping implementations.....91
Fig. 3.19. Pressure profile for a vertical section taken through the centre of the fractured domain.....92
Fig. 3.20. 2D fractured domain containing intersecting fractures with applied pressures at top and bottom boundary.....93
Fig. 3.21. Finite element meshes used for simulation. 94
Fig. 3.22. Pressure fields for discrete fracture and fracture mapping models for intersecting fractures.....95
Fig. 3.23. Influence of interacting fractures on the flow field direction for DFM and FM. 96
Fig. 4.1. Discretized fractured domain. (a) Fractures cutting across elements, (b) Elements aligned with fracture. 99

Fig. 4.2. Coordinate configuration for fracture front enrichment functions in 2D. $\mathbf{x}*$ represents the fracture tip and $\mathbf{x}$ represents either an integration point or finite element node. 100

Fig. 4.3. Assignment of enriched nodes. ○ Heaviside enriched □ Tip enriched 102

Fig. 4.4. Intersecting fractures modelled by: (a) finite element method and (b) the extended finite element method. 103

Fig. 4.5. Discontinuous enrichment functions, $H_1(\mathbf{x})$, $H_2(\mathbf{x})$ and $Br(\mathbf{x})$. 103

Fig. 4.6. Nodal enrichment for intersecting fracture..... 104

Fig. 4.7. Element decomposition into sub-elements: (a) elements containing fracture tip; (b) elements completely cut by fracture. Sub-elements are obtained by applying Delaunay triangulation to the vertex nodes of the element and end nodes of fracture segments contained within the element. 106

Fig. 4.8. Penetration condition of fracture faces. 108

Fig. 4.9. Segments created to determine if contact has occurred between fracture faces. 110

Fig. 4.10. 2D fractured domain with assigned boundary conditions. 117

Fig. 4.11. Initial FEM mesh along with deformed meshes: (a) contact between fracture faces is considered; (b) contact between fracture faces is ignored..... 118

Fig. 4.12. Displacement field: (a) contact between fracture faces is considered; (b) contact between fracture faces is ignored. 118

Fig. 4.13. 2D fractured domain with assigned boundary conditions. 120

Fig. 4.14. Computational meshes used to conduct the comparison between DFM and FM-XFEM. 121

Fig. 4.15. Displacement field for fractured domain after 100days for DFM and FM-XFEM. 122
Fig. 4.16. Surface displacement at point A obtained for DFM and FM-XFEM. 123
Fig. 4.17. Pressure field for fractured domain after 100days for DFM and FM-XFEM. 125
Fig. 4.18. Excess pore pressure at point B obtained for DFM and FM-XFEM.125
Fig. 4.19. Convergence rate of FM-XFEM. 126
Fig. 4.20. Computational mesh along with resulting displacement field and pressure contours for domain containing intersecting fractures. 127
Fig. 5.1. Integration domain for evaluation of *J*-integral..... 131
Fig. 5.2. Fundamental fracture tip displacement modes. 132
Fig. 5.3. Flow chart of fracture propagation algorithm. 136
Fig. 5.4. Model for stress intensity simulation. 138
Fig. 5.5. Error in numerically determined stress intensity factor. 140
Fig. 5.6. Final fracture configuration and domain stress fields..... 141
Fig. 5.7. Initial and final fracture configurations along with mesh..... 143
Fig. 5.8. Final fracture configuration and domain stress fields..... 144
Fig. E.1. Infinite cracked plate loaded by remote tensile stress. 176
Fig. E.2. Finite element mesh used to model infinite plate with crack..... 177
Fig. E.3. Strain fields, ε_x, obtained for infinite plate with crack. 178
Fig. E.4. Comparison of exact strain to recovered strain, ε_x, at vertical sections taken at (a) x=0.5, y, (b) x=1.0, y and (c) x=1.5, y, through the domain..... 179

LIST OF TABLES

Table 2.1. Material properties for consolidation example.....48

Table 3.1 Computed well drawdown vs dimensionless time.....84

Table 4.1. Material properties for contact algorithm example. 117

Table 4.2. Material properties for coupled example. 120

Table 5.1. Comparison between analytical and numerical stress intensity factor

..... 139

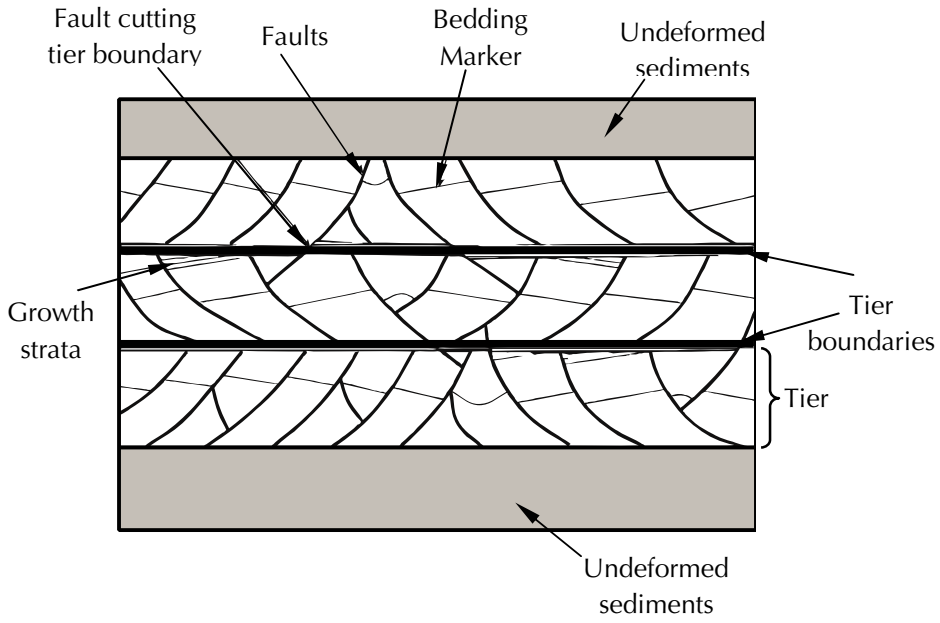
LIST OF BOXES

Box 4.1. The Newton-Raphson Iterative Procedure. 111

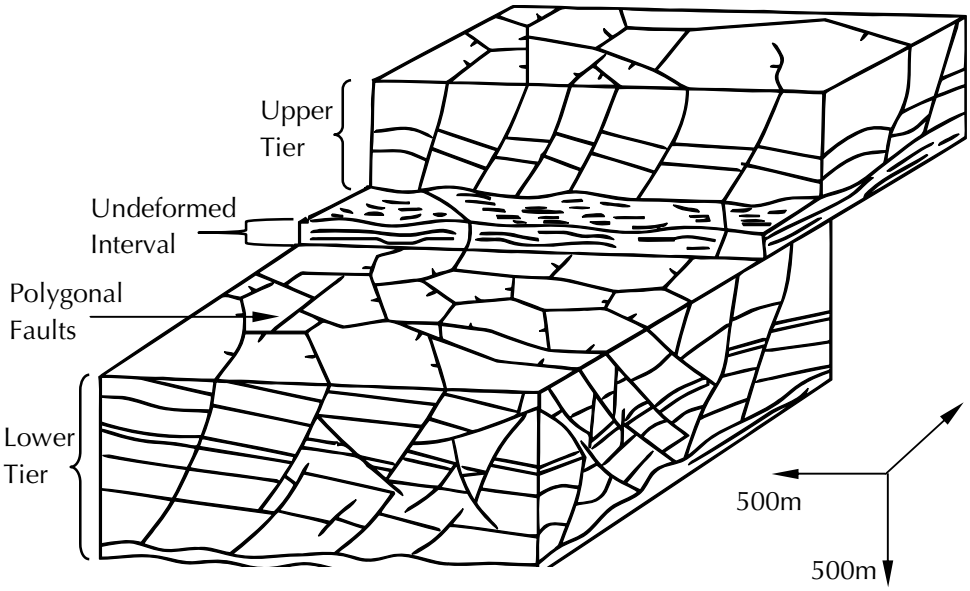
CHAPTER 1

Introduction

Polygonal structures have been recently identified in many fields of the Arabian Gulf, both onshore and offshore platform carbonates, due to recent refinements in 3D seismic processing and interpretation (Bockel-Rebelle *et al.*, 2006; Moen-Maurel *et al.*, 2007). Connectivity of these faults radically affects the behaviour of hydrocarbon reservoirs and can both facilitate and impede flow (hydrocarbon and/or water), therefore their distribution and effect on fluid flow has direct impacts on the development of reservoir recovery schemes and their economic evaluation. Faults in one tier may partially interconnect with those in adjacent tiers by cross-propagation of a sub-set of the total fault population, but the majority of the faults in the separate tiers are contained wholly within individual tiers (Fig. 1.1). A horizontal cross-section of the polygonal faults observed in the North Sea is also shown in Fig. 1.2.



(a) Schematic diagram of a seismic section of a 2D layer-bound fault system taken from Dewhurst *et al. (*1999). Individual faults do not generally cut through the entire deformed succession, but are restricted to particular layers, termed tiers to convey the idea of a set of partially connected faulted layers. Some larger faults do connect from tier to tier, but these are subordinate in number to those confined within individual tiers.



(b) Three-dimensional schematic diagram of a polygonal fault system. (taken from Dewhurst *et al. (*1999))

Fig. 1.1. Schematic diagram of a polygonal fault systems; (a) 2D and (b) 3D.

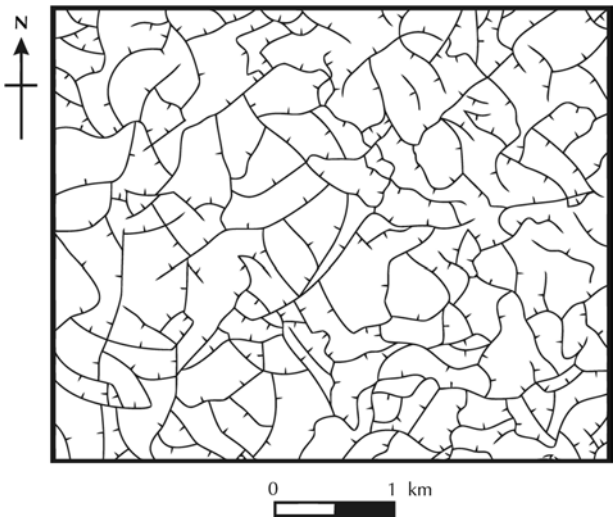


Fig. 1.2. Horizontal cross-section of Upper Oligocene in Block 16/26 of the UK sector of the North Sea showing polygonal faults. (taken from Cartwright and Lonergan, (1996).

The formation mechanism for polygonal structures and the necessary physical conditions are also of great interest to exploration and production schemes as new faults may develop or existing faults propagate during hydrocarbon recovery. The four competing, hypothesised genetic formation mechanisms found in literature are Density Inversion – (Henriet *et al.*, 1991; Cartwright, 1994), Gravity Collapse – (Clausen *et al.*, 1999), Syneresis – (Cartwright and Dewhurst, 1998) and Compactional loading at low coefficients of residual friction – (Goulty, 2001). A consensus exists among these mechanisms that polygonal faults are formed during coupled compaction and compaction driven flow of the host sediments (consolidation). However, to the author's knowledge, no numerical models have been developed to test the hypothesised formation mechanisms. This research work presents a numerical framework for simulating fracture propagation in porous media that can be extended to test the validity of these formation mechanisms.

Previously, numerical models for simulating fracturing in saturated porous media have been presented by De Borst *et al.* (2006) and Al-Khoury and Sluys (2006). The methods, developed independently of each other, use the partition of unity method (PUM), in their formulation to describe discontinuous displacement and/or pressure fields across existing fractures. PUM the is predecessor of the extended finite element method (Melenk and Babuska, 1996; Belytschko *et al.*, 2009). At this juncture a definition for a porous medium is provided. A porous medium is composed of a matrix and interconnected pore spaces. These interconnected pores may contain and conduct fluid (Coussy, 2004).

The model proposed by De Borst assumes that both the displacement and pressure fields within the medium are discontinuous. The method also assumes that a diaphragm exists at the discontinuity. This diaphragm allows fluid flow into the fracture and assumes that once the fluid has entered the fracture it can only flow along the fracture length and never re-enter the porous matrix. This assumption is a limitation of their model since a reversal in the pressure gradient would undoubtedly cause fluid migration from the fracture to the porous matrix. Their model therefore does not provide a realistic representation of the fluid interaction between the porous matrix and fractures.

Al-Khoury and Sluys also assumes a discontinuous displacement field; however, the dual porosity concept is used to simulate the fluid interaction between the porous matrix and fractures within the domain. Although the introduction of the dual porosity concept into their formulation adequately simulates the fluid flow interaction between the matrix and fracture, the model

does not address the scaling of the fracture properties (porosity and permeability) within the domain. Also their formulation does not include the tip-enrichment terms or address fracture propagation.

1.1 Scope of Thesis

This work presents a numerical framework developed for the coupled deformation, fluid flow and fracture propagation in porous media that can be extended to study the formation and evolution of polygonal faults. This technique uses a new fracture mapping approach to quantify the fluid flow interaction between the matrix and fractures and uses the extended finite element method to account for the discontinuity in the displacement field across existing fractures. Fracture propagation is achieved using a mixed mode fracture criterion dependent on stress intensity factors determined using the *J*-integral approach. The methodology developed is generic and also applicable to the fields of reservoir engineering and CO_2 storage.

Coupled deformation and fluid flow reservoir simulations are conducted in stress sensitive, naturally fractured hydrocarbon reservoirs to accurately capture changes in stress dependent permeabilities that may occur during production as the fluid pressure decreases. Although not all hydrocarbon reservoirs are located in geologically weak fractured units, coupled simulation still plays an important role since over-pressurized reservoirs in stable environments have undergone subsidence at the start of production (Minkoff *et al.*, 2003). Modelling coupled fluid flow and deformation within a reservoir is necessary for predicting pressure response, recovery, and the impact of

subsidence or heave during production. CO_2 injection into deep subsurface aquifers or disused hydrocarbon reservoirs causes changes in pore pressure and may lead to geomechanical deformation of both the reservoir and surrounding rock. The deformation may lead to the generation of new fractures or reactivation of existing fractures, providing pathways for migration from targeted storage zones.

1.2 The Numerical Model

The finite element method (FEM) has been successfully used to develop coupled deformation and flow models (Potts and Zdravkovic, 1999; Zienkiewicz *et al.*, 2005); however, these models currently do not address the inclusion of strong (displacement) or weak (strain) discontinuities. Geological faults are defined as fractures that have undergone appreciable shear deformation and fall under the heading of strong discontinuities. To obtain accurate results within the finite element framework the computational mesh must be aligned with existing discontinuities and refined in areas where the solution is expected to possess singularities or large gradients. As discontinuities evolve with time the computational domain must be continually remeshed; remeshing is computationally expensive and very time consuming for complex geometries containing several hundred discontinuities. The standard FEM is therefore not ideally suited for fault modelling on the field scale; on the kilometre scale.

Over the last few years several numerical techniques have been developed to model discontinuities in solids independently of the FE mesh. These include

and are not limited to, the incorporation of a discontinuous mode on an element level (Oliver, 1995), a moving mesh technique (Rashid, 1998), and an enrichment technique for finite elements based on a partition-of-unity which does not require remeshing (Moës *et al.*, 1999). The method presented by Moes *et al.* (1999), referred to as the extended finite element method (XFEM) has gained a great deal of attention due to its accuracy and versatility in simulating discontinuities (Dolbow, 1999; Daux *et al.*, 2000; Stolarska *et al.*, 2001; Moës *et al.*, 2002; Huang *et al.*, 2003; Budyn *et al.*, 2004; Bordas *et al.*, 2008). The method allows the integration of discontinuities into the computational domain independently of the finite element mesh and reduces the level of complexity and refinement of the mesh in the vicinity of discontinuities. XFEM has been used to modify the single phase FEM coupled deformation and fluid flow model to allow the inclusion of discontinuities into the computational domain as part of this work.

Although the discontinuous displacement field across fractures can be facilitated through the integration of XFEM into the consolidation model, the flow between the porous matrix and fractures also needs to be addressed. A fracture mapping (FM) approach has been introduced into the formulation to account for this flow interaction. The mapping approach allows the incorporation of fractures into the computational domain without having the mesh aligned with existing discontinuities. Finite elements which contain a fracture are viewed as two overlapping elements; a matrix element and a fracture element. The fracture properties are mapped onto the fracture element and the flow interaction between the fracture and matrix elements is done

using the quasi steady-state transfer function, adopted from Barenblatt *et al.* (1960). Mapping is used to produce a realistic representation of fracture fluid flow properties and geometry within individual finite elements.

In FM-XFEM the discretization of the computational domain and the fractures are performed independently, i.e. there is no requirement to explicitly mesh the fractures into the computational mesh. Existing fractures are represented using the level set method (LSM) (Osher and Sethian, 1988) which provides a mathematical description of fractures within the domain. The method represents an interface, in this case a fracture, as the zero of a function called the level set function (Duflot, 2007). In the FEM implementation elements intersected by the level set are decomposed into smaller elements to prevent numerical difficulties that may arise during numerical integration (Sukumar *et al.*, 2000; Sukumar and Prévost, 2003).

Propagation of existing fractures is achieved using classical fracture mechanics theory which is based on the theory of linear elasticity and hence referred to as Linear Elastic Fracture Mechanics (LEFM). This theory assumes that material nonlinearity associated with the fracturing process only occurs at the crack tip and remains valid if the fractured domain as a whole behaves in an approximately elastic manner (Janssen *et al.*, 2004). If the domain undergoes irrecoverable plastic deformation elastic-plastic fracture mechanics (EPFM) then becomes applicable. Since the work conducted in this research is limited to elastic deformation LEFM theory has been used for fracture calculations. The use of standard finite element approaches in LEFM requires the mesh conform to existing discontinuities and costly mesh generation/regeneration as

discontinuities propagate. The mesh also has to be highly refined to capture the singularity in the stress fields at the fracture tip. The extended finite element method provides a viable alternative to solving LEFM without the shortcomings experienced using standard FEM.

This work also presents a meshfree implementation of the fracture mapping scheme for fluid flow in fractured porous media. Meshfree methods have evolved from the need to reduce the dependence on a finite element mesh in the formulation of a numerical solution. The meshfree implementation is ideal for modelling complicated geological formations as well as areas in the vicinity of existing fractures that would normally require complex meshing using FEM. Meshfree methods use a set of nodes scattered within the problem domain as well as along the boundaries of the domain to represent the problem domain and its boundaries. These methods include, among others, the discrete element method (DEM, Cundall and Strack, 1979), element free galerkin (EFG, Belytschko *et al.*, 1994), and the radial point integration method (RPIM, Wang and Liu, 2002). A nodal integration variant of RPIM, NI-RPIM, was used for the meshfree implementation of the fracture mapping approach presented.

In the DEM, computational points are associated with a finite size and shape. Interaction between points (particles) occurs only where they are in contact - according to the specified interaction rules. This method is very well suited to modelling fractures since particle interactions can take the form of breakable bonds, however, DEMs are processor intensive and this limits either the length of a simulation or the number of particles.

EFG is continuum method in which the shape functions are calculated using the moving least-square method (MLS). Shape functions constructed using MLS do not satisfy the Kronecker delta function properties, i.e., the shape function at the node is equal to one and zero for all other nodes used to construct the shape function. These shape functions therefore have the following disadvantages; (1) difficulties in the implementation of essential boundary conditions and; (2) complexity of algorithms to construct shape functions. Several methods have been proposed to overcome these disadvantages for example the Lagrangian method (Lu *et al.*, 1994), the penalty method (Onate *et al.*, 1996), the collocation method (Onate *et al.*, 1996; Wagner and Liu, 2000) for essential boundary conditions. Analytical integration (Lancaster and Salkauskas, 1981), the recursive method (Breitkopf *et al.*, 2000) and parallel computing (Danielson *et al.*, 2000) have been proposed for the construction of shape functions.

The RPIM formulation combines radial and polynomial basis functions to address the problems of implementing essential boundary conditions and construction of shape functions (2002). Involvement of radial basis functions overcome possible singularity associated with the meshfree methods based on only the polynomial basis (Liu and Gu, 2001). The interpolation (shape) function obtained passes through all scattered points in an influence domain and thus satisfy the Kronecker delta function property. This makes the implementation of essential boundary conditions much easier than meshfree methods based on the moving least-squares approximation, e.g., the element

free Galerkin method. A nodal integration variant of RPIM was used for the meshfree implementation (NI-RPIM) presented.

Although the meshfree method was able to adequately describe the fluid flow in fractured porous media, enriching RPIM with the partition-of-unity as part of this work did not correctly describe the discontinuous strain field across the fracture and therefore a fully coupled meshfree scheme has not been presented in this thesis. However, the development of a fully coupled meshfree scheme has been cited as an area of future work and is discussed in chapter 6 of this thesis.

1.3 Implementation of the Numerical Model

The numerical methods discussed within this thesis have been implemented using MATLAB®. MATLAB® is a high-level computing language and interactive environment for algorithm development. The software's immense predefined mathematical library allows users to quickly develop and code numerical methods. MATLAB® also has matrix (sparse and dense), vector and many linear algebra tools already defined allowing users to focus entirely on algorithm implementation without having to define these data structures. The extensive mathematics and graphics functions further free users from the arduous task of developing these functions themselves or finding equivalent pre-existing libraries. It should be noted that the built-in functions are already compiled and are extremely efficient and should be used as much as possible.

There is a disadvantage to this ease of development. Since MATLAB® is an interpretive language, each line of code is interpreted by the command line

interpreter and executed sequentially at run time; the run times can be much greater than that of compiled programming languages like FORTRAN or C++. The slow down due to the interpretive nature of the program can be circumvented through the use of MATLAB® Executable (MEX) files. MEX-files are dynamically linked subroutines generated from FORTRAN or C++ source code that, when compiled, can be used in the same manner as built-in functions or MATLAB® script files. The main advantages provided by the use of MEX-files are: the ability to speed up bottleneck computations by rewriting routines in a lower level programming language; and the ability to call existing FORTRAN or C++ routines/libraries without having to rewrite code as native MATLAB® script. An example of a FORTRAN 90 and C++ MEX-file are included in Appendix A. The FORTRAN 90 MEX-file, Ma57Mex.f90, is an interface to the hsl_m57a direct matrix solver (Duff, 2004) which is 50% faster than the MATLAB® built-in solver. The C++ Mex-file, VoroMex.cpp, is an interface to Voro++ (Rycroft *et al.*, 2006); a C++ library for the computation of 2D and 3D Voronoi cells. Since MATLAB® does not calculate bounded Voronoi cells or return cell information such as cell areas, cell centroids or connecting vertices, Voro++ was used to prepare the Voronoi cells used for the meshfree nodal integration implementation described in chapter 3. MEX-files were also used for the global matrix assembly procedure.

Regular quadrilateral meshes used for the numerical simulations presented were constructed using a mesh generator developed in MATLAB® as part of this research while unstructured meshes were generated using DISTMESH (Persson and Strang, 2004);

1.4 Thesis Layout

The organization of this thesis is as follows. Chapter 2 discusses the theory of consolidation (coupled deformation and fluid flow) and presents the finite element discretzation of the governing equations. Numerical examples are also provided to verify that the MATLAB® consolidation code implemented within this study correctly solves the governing equations which describe coupled deformation and fluid flow in porous media.

Chapter 3 presents the fracture mapping scheme developed to describe fluid flow interaction between fractures and the porous matrix. The FEM and NI-RPIM implementation procedures are then discussed. A numerical example of a closed reservoir containing a vertical fracture is also included and the numerical results are compared to the analytical solution existing in literature. The method is also compared to the discrete fracture model in which existing fractures are explicitly meshed along with the porous matrix.

XFEM implementation is discussed in Chapter 4. The discretized equations which incorporate the fracture mapping are then presented along with a numerical example that tests the viability of the fracture mapping - extended finite element method (FM-XFEM) consolidation formulation.

Chapter 5 presents the fracture propagation aspect of this work. The chapter begins with a brief description of the stress intensity factor determination using the *J*-integral approach. Stress intensity factors provide a measure of the stress concentration at fracture tips. The criterion for mixed mode fracture propagation is presented along with the propagation algorithm. A numerical

example is presented to compare the stress intensity factors obtained from the developed FM-XFEM to an analytical solution. The final chapter presents conclusions drawn from the proposed methodology along with possible directions for future work.

1.5 Notation

Stress and strain tensors are written using Voigt notation throughout this document. Voigt notation exploits the symmetry of these tensors and allows them to be represented in vector form. The two dimensional symmetric stress and strain tensors take the following forms using this notation:

$$\begin{aligned} \boldsymbol{\sigma} &= \begin{pmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{pmatrix} = \left[\sigma_x, \sigma_y, \tau_{xy}\right]^{\mathrm{T}}, \\ \boldsymbol{\varepsilon} &= \begin{pmatrix} \varepsilon_x & \gamma_{xy} \\ \gamma_{xy} & \varepsilon_y \end{pmatrix} = \left[\varepsilon_x, \varepsilon_y, \gamma_{xy}\right]^{\mathrm{T}}. \end{aligned} \tag{1.1}$$

1.6 Thesis Contributions

The work presented in this thesis can be categorized into the following contributions to the existing body of scientific knowledge:

1) A new fracture mapping scheme which allows the inclusion of fractures into a domain without explicit discretization, i.e., fractures are not meshed and need not conform to the computational mesh. A finite element and meshfree implementation of this scheme are also presented.

2) A scheme for coupled deformation and fluid flow in fractured porous media. This numerical technique combines the extended finite element method with the fracture mapping approach (FM-XFEM).
3) The FM-XFEM method has been extended to handle fracture propagation in porous elastic (poroelastic) media.

CHAPTER 2

Consolidation Theory and its Finite Element Formulation

This chapter presents a brief overview of linear elastic consolidation theory along with the finite element discretization of the governing equations which describe the coupled deformation and fluid flow in porous media. Numerical experiments are included at the end of this chapter to verify the single phase FEM consolidation implementation in MATLAB®. This verification is used to ensure that the implemented code correctly solves the governing equations before fracture mapping and the extended finite element method are used to modify the formulation in chapter 4.

2.1 Consolidation Theory

When a layer of saturated soil is subjected to stress, the volume of the layer is reduced and excess fluid is expelled from the layer; this constitutes the process of consolidation. The first mathematical description of the consolidation process was developed by Terzaghi (1925). Terzaghi's theory is based on the physical assumption of mass conservation within the soil-pore fluid system. Despite the widespread use of this theory in the field of geotechnical analysis it

does not adequately address the coupling between the soil skeleton deformation and pore fluid flow. Terzaghi assumed that the pore pressure field and the applied stress field are uncoupled; i.e., the pore pressure history curve is expected to behave monotonically after a stress is applied to a drained system.

Biot (1941; 1955) generalized Terzaghi's theory and provided a three dimensional formulation for the coupling between the soil skeleton and pore fluid, based on stress-strain constitutive relations. Biot's theory combines the mechanics of elastic rock deformation described by Hooke's Law and the fluid flow in a porous media which is sufficiently approximated by Darcy's law. The soil skeleton is treated as a porous elastic solid and the laminar flow of the pore-fluid is coupled to the solid by the conditions of compressibility and continuity. This theory produced a realistic model for the mechanical behaviour of saturated soils or rocks and accounts for the time dependent relationship that exists between the soil skeleton and the pore fluids.

2.2 Governing Equations of Consolidation Theory

This section provides an overview of the equations that describe the physical process of coupled elastic deformation and fluid flow in porous media. The stress and strain matrices given within these equations take the following forms in 2D using Voigt notation:

$$\begin{aligned} \boldsymbol{\sigma} &= \left[\sigma_x, \sigma_y, \tau_{xy}\right]^{\mathrm{T}}, \\ \boldsymbol{\varepsilon} &= \left[\varepsilon_x, \varepsilon_y, \gamma_{xy}\right]^{\mathrm{T}}. \end{aligned} \tag{2.1}$$

A saturated soil/rock mass can be viewed as a two-phase deformable porous medium in which the pores are filled with a liquid. Although the porous matrix and pore fluids interact at the micro-level when subject to external loading, Biot's theory provides a comprehensive description of this interaction at the macro level and is developed from the following concepts:

1) Force equilibrium of the soil-water mixture:

$$\nabla_s^T \boldsymbol{\sigma} + \rho \mathbf{g} = 0, \tag{2.2}$$

where $\boldsymbol{\sigma}$ is total stress, $\mathbf{g}$ is gravity and ρ is the averaged density of a multiphase system and ∇_s is the symmetric gradient operator matrix given by:

$$\nabla_s = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix}, \tag{2.3}$$

2) The total stress of the system, assuming that tension is positive, is given by:

$$\boldsymbol{\sigma} = \boldsymbol{\sigma}'' - \alpha \mathbf{I} P_w, \tag{2.4}$$

where $\boldsymbol{\sigma}''$ is a general form of the effective stress, P_w is the water pore pressure in the porous matrix, $\mathbf{I}$ is the unit tensor written as $[1\ 1\ 0]^T$ in two dimensions using Voigt notation and α is Biot's compressibility coefficient. Equation (2.4) represents Terzaghi's effective stress principle which has been modified to take into account the compressibility of the system through the inclusion of Biot's compressibility coefficient.

3) The constitutive stress-strain relationship of the solid phase is given by:

$$d\boldsymbol{\sigma}'' = \mathbf{D} d\boldsymbol{\varepsilon}, \tag{2.5}$$

where $\mathbf{D}$ is the linear elastic material matrix and $\boldsymbol{\varepsilon}$ is the strain of the system. For two dimensional plane stress problems $\mathbf{D}$ is defined as:

$$\mathbf{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix}, \tag{2.6}$$

and for plane strain it is defined as:

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \dfrac{1}{2}-\nu \end{bmatrix}, \tag{2.7}$$

where E is the modulus of elasticity, ν is Poisson's ratio. Plane stress is defined as a state of stress in which the normal stress σ_z, and the shear stresses, τ_{yz}, τ_{zx} directed perpendicular to the x-y plane are assumed to be zero while plane strain is defined to be a state of strain in which the strain normal to the x-y plane, ε_z, and the shear strain γ_{yz}, γ_{zx} are assumed to be zero. The numerical examples presented in this thesis are all plane strain problems.

4) The relationship between the change in strain, $d\boldsymbol{\varepsilon}$, and the change in displacement, $d\mathbf{u}$, given by:

$$d\boldsymbol{\varepsilon} = \nabla_s d\mathbf{u}, \tag{2.8}$$

This linear relationship is only valid under the assumption of small strain theory where the components of $\nabla_s \mathbf{u}$ are much smaller in magnitude than 1 (Davis and Selvadurai, 2002).

5) Darcy's seepage velocity given by:

$$v_d = \frac{\mathbf{K}}{\mu_w}(-\nabla P_w + \rho \mathbf{g}), \tag{2.9}$$

where μ_w is the fluid viscosity and $\mathbf{K}$ is the material permeability.

6) The flow through the porous material which is equal to the reduction in volume and is given by:

$$\nabla \cdot v_d = -\alpha \mathbf{I}^T \frac{\partial \boldsymbol{\varepsilon}}{\partial t}, \tag{2.10}$$

7) The rate of change of grain volume due to pressure changes within the domain which is given by:

$$\left(\frac{\alpha - n}{K_s} + \frac{n}{K_w}\right)\frac{\partial P_w}{\partial t} \tag{2.11}$$

where K_s and K_w represent the bulk modulus of the material skeleton and water respectively. Equations (2.9) - (2.11) together define the continuity/mass balance equation which is given by:

$$\left(\frac{\alpha - n}{K_s} + \frac{n}{K_w}\right)\frac{\partial P_w}{\partial t} + \nabla \cdot \left(\frac{\mathbf{K}}{\mu_w}(-\nabla P_w + \rho \mathbf{g})\right) + \alpha \mathbf{I}^{\mathrm{T}} \frac{\partial \boldsymbol{\varepsilon}}{\partial t} = 0, \tag{2.12}$$

2.2.1 Initial and Boundary conditions

Initial conditions which specify the displacements and water pressures at $t = 0$ are given by:

$$\mathbf{u} = \mathbf{u}_0 \quad \text{in } \boldsymbol{\Omega} \cup \boldsymbol{\Gamma}, \tag{2.13}$$

$$P_w = P_w^0 \quad \text{in } \boldsymbol{\Omega} \cup \boldsymbol{\Gamma}, \tag{2.14}$$

where $\boldsymbol{\Omega}$ is the domain of interest and $\boldsymbol{\Gamma}$ its boundary. Boundary conditions can be imposed values on $\boldsymbol{\Gamma}_{field}$ (Dirichlet boundary conditions), or fluxes on $\boldsymbol{\Gamma}_{field}^q$ (Nemann boundary conditions) where the boundary $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_{field} \cup \boldsymbol{\Gamma}_{field}^q$. Dirichlet boundary conditions for displacements and pore-water pressures are given by:

$$\mathbf{u} = \hat{\mathbf{u}} \quad \text{on } \boldsymbol{\Gamma}_u, \tag{2.15}$$

$$P_w = \hat{P}_w \quad \text{on } \boldsymbol{\Gamma}_w, \tag{2.16}$$

while Neumann boundary conditions for imposed stresses and flux conditions are as follows:

$$\bar{\mathbf{t}} = \mathbf{n} \cdot \boldsymbol{\sigma} \quad \text{on } \boldsymbol{\Gamma}_u^q, \tag{2.17}$$

$$q_w = \rho_w \frac{\mathbf{K}}{\mu_w}(-\nabla P_w + \rho_w \mathbf{g})^{\mathrm{T}} \cdot \mathbf{n} \quad \text{on } \Gamma_w^q, \tag{2.18}$$

where $\mathbf{n}$ is the unit normal vector to the surface, $\bar{\mathbf{t}}$ is the prescribed external traction force and q_w is the applied mass flux normal to the boundary.

2.3 Numerical Simulation

The goal of numerical simulation is to transform a complex practical problem into a simple discrete mathematical description, recreate and solve the problem numerically. It is often possible to develop a numerical/approximate solution for complex natural events, as long as an appropriate numerical method is employed. When using numerical methods a discrete approximation for the solution is sought, i.e., computed values of the field variable at a set of specified points within the domain.

FEM has been successfully applied to solve consolidation problems (Smith and Griffiths, 2004; Zienkiewicz and Taylor, 2005). In FEM the computational domain is subdivided into discrete elements (domain discretization) that are interconnected at their vertices called nodes. Field variables (e.g. displacement and pore pressure) are determined at the nodes and are interpolated over the elements using polynomial approximating functions (shape functions).

2.4 Spatial Discretization

The discrete equations are obtained by applying the Galerkin-based finite element approach to the governing and continuity equations presented in

section 2.2. The application of the finite element method to the consolidation equations is given in Appendix B. The dependent displacement and pressure variables are approximated by:

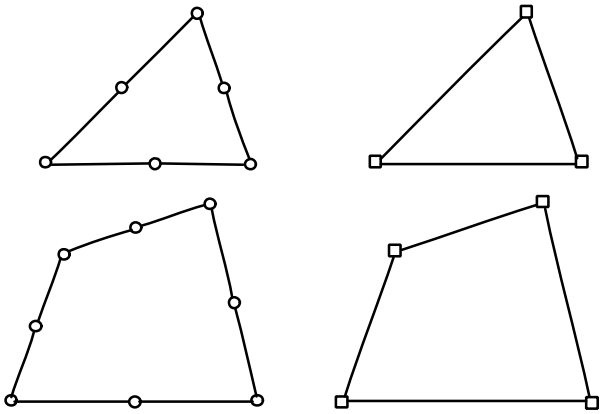
$$\mathbf{u} \approx \mathbf{u}^h = \mathbf{N}_u \bar{\mathbf{u}}, \tag{2.19}$$

and

$$P_w \approx P_w^h = \mathbf{N}_p \overline{\mathbf{P}}_w, \tag{2.20}$$

where $\mathbf{u}^h$ and P_w^h are the discrete approximations of the displacement and pressures respectively; $\bar{\mathbf{u}}$ and $\overline{\mathbf{P}}_w$ are the vectors of nodal unknowns of displacement and excess pressures at the finite element nodes. A second order polynomial is used for the displacement shape function $\mathbf{N}_u$ while a first order polynomial is used for the pressure $\mathbf{N}_p$. The use of equal-order interpolation violates the Babuska-Brezzi convergence condition, particularly when approaching the undrained limit state (Babuška, 1971, 1973; Brezzi, 1974). Although equal-order interpolants are possible, stabilization techniques are required for example the Galerkin least square method, enhanced strain stabilization and characteristic base split procedure. Mixed order shape functions have been used in this study owing to its relative simplicity and prior successful implementation in existing consolidation codes (Potts and Zdravkovic, 1999; Smith and Griffiths, 2004; Zienkiewicz and Taylor, 2005). The finite element combinations used for the coupled numerical simulations presented in this thesis are the Taylor-Hood element (triangular six/three node element pair - T6/T3) and the quadrilateral eight/four node element pair (Q8/Q4) (Fig. 2.1). Displacements are calculated at the element vertices and mid-side nodes of the quadratic elements (triangular - T6 and quadrilateral -

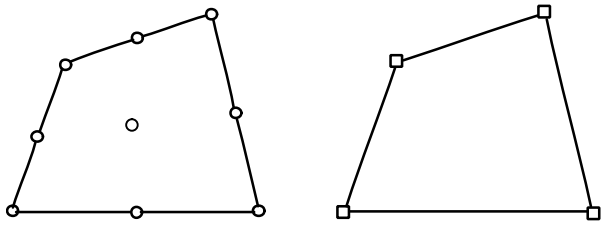
Q8) and pressures are calculated at the vertices of the linear elements (triangular - T3 and quadrilateral - Q4).



(a) Quadratic displacement elements (b) Linear pressure elements

Fig. 2.1. Element combination used for interpolation of the coupled solid-fluid interaction (a) Quadratic elements are used for displacement, (b) Linear elements are used for pressure. Displacements are calculated at element vertices and mid-side nodes of the quadratic elements.

Since the numerical examples presented in this thesis are allowed to drain along a boundary, a volume change can occur and the domain cannot be considered as incompressible (Davis and Selvadurai, 2002). If the domain was not allowed to drain, the Q9/Q4 (Fig. 2.2) element pair may also be considered since it satisfies the Babuska-Brezzi stability condition, also known as the Inf-Sup condition. It should be noted that despite satisfying the Inf-Sup stability condition, very poor approximations of the incompressibility condition are frequently noted when using the Q9/Q4 finite element pair (Hughes, 2000). The Q8/Q4 is also widely used in incompressible analysis (Hughes, 2000).



(a) Quadratic displacement elements (b) Linear pressure elements

Fig. 2.2. Illustration of Q9/Q4 finite element pair.

Applying the Galerkin-based finite method to the governing and continuity equations presented in section 2.2, the fully coupled discretized equations describing deformation and fluid flow in porous media take the form (Appendix B provides details on the derivation of these terms):

$$-\mathbf{Ke}\,\frac{\partial \overline{\mathbf{u}}}{\partial t} + \mathbf{L}\frac{\partial \overline{\mathbf{u}}}{\partial t} = -\mathbf{F}^{u} \tag{2.21}$$

$$\mathbf{H}\overline{\mathbf{P}}_{w} + \mathbf{L}^{\mathrm{T}}\frac{\partial \overline{\mathbf{u}}}{\partial t} + \mathbf{S}\frac{\partial \overline{\mathbf{P}}_{w}}{\partial t} = \mathbf{F}^{p} \tag{2.22}$$

Equations (2.21) and (2.22) can be conveniently cast into matrix form and re-written as:

$$\begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{H}\end{bmatrix}\begin{bmatrix}\overline{\mathbf{u}}\\ \overline{\mathbf{P}}_{w}\end{bmatrix} + \begin{bmatrix}-\mathbf{Ke} & \mathbf{L}\\ \mathbf{L}^{\mathrm{T}} & \mathbf{S}\end{bmatrix}\frac{\partial}{\partial t}\begin{bmatrix}\overline{\mathbf{u}}\\ \overline{\mathbf{P}}_{w}\end{bmatrix} = \begin{bmatrix}-\mathbf{F}^{u}\\ \mathbf{F}^{p}\end{bmatrix}, \tag{2.23}$$

where

Ke is the elastic stiffness matrix,

L is the coupling matrix,

H is the permeability matrix,

S is the compressibility matrix,

$\mathbf{F}^{u}$ is the compressibility matrix,

and $\mathbf{F}^{p}$ is the force vector due to the flow field.

The elements of these matrices are given below.

$$\mathbf{Ke} = \int_{\Omega} \mathbf{B}^{\mathrm{T}}\, \mathbf{D}\mathbf{B}\, d\Omega, \tag{2.24}$$

$$\mathbf{L} = \int_{\Omega} \mathbf{B}^{\mathrm{T}}\, \alpha \mathbf{I} \mathbf{N}_p\, d\Omega, \tag{2.25}$$

$$\mathbf{H} = \int_{\boldsymbol{\Omega}} \left(\nabla \mathbf{N}_p\right)^{\mathrm{T}} \frac{\mathbf{K}}{\mu_w}\, \nabla \mathbf{N}_p\, d\Omega, \tag{2.26}$$

$$\mathbf{S} = \int_{\Omega} \mathbf{N}_p^{\mathrm{T}} \left(\frac{\alpha - n}{K_s} + \frac{n}{K_w}\right) \mathbf{N}_p\, d\Omega, \tag{2.27}$$

$$\mathbf{F}^u = \int_{\Omega} \mathbf{N}_u^{\mathrm{T}}\, \rho \mathbf{g}\, d\Omega + \int_{\Gamma_u^q} \mathbf{N}_u^{\mathrm{T}}\, \bar{\mathbf{t}}\, d\Gamma, \tag{2.28}$$

$$\mathbf{F}^p = \int_{\Omega} \left(\nabla \mathbf{N}_p\right)^{\mathrm{T}} \frac{\mathbf{K}}{\mu_w} \rho^w\, \mathbf{g}\, d\Omega - \int_{\Gamma_w^q} \mathbf{N}_p^{\mathrm{T}}\, \frac{q^w}{\rho^w}\, d\Gamma, \tag{2.29}$$

$$\mathbf{B} = \nabla_{\mathrm{s}} \mathbf{N}_u = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \mathbf{N}_u, \tag{2.30}$$

$$\nabla \mathbf{N}_p = \begin{bmatrix} \dfrac{\partial}{\partial x} \mathbf{N}_p \\ \dfrac{\partial}{\partial y} \mathbf{N}_p \end{bmatrix}. \tag{2.31}$$

2.5 Temporal Discretization

The temporal discretization can be divided into two distinct methods; the absolute load version and an incremental load version. The absolute load version applies the entire external loading at each time step and is suitable for linear elastic problems. The incremental version is best suited for non-linear systems in which the load must be redistributed in each time step to ensure the system return to equilibrium, e.g. the use of penalty method for contact algorithm (see Chapter 4).

2.5.1 Absolute Load

The discretized form of the consolidation equations can be written in a more concise form, i.e.

$$\mathbf{AX} + \mathbf{B}\frac{\mathbf{dX}}{\mathbf{dt}} = \mathbf{C}. \tag{2.32}$$

Time discretization is achieved using a variable-θ finite difference scheme given by the following approximations (Smith and Griffiths, 2004):

$$\left(\frac{d\mathbf{X}}{dt}\right) = \frac{\mathbf{X}_{n+1} - \mathbf{X}_n}{\Delta t}, \tag{2.33}$$

$$\mathbf{X} = (1-\theta)\mathbf{X}_n + \theta\mathbf{X}_{n+1}, \tag{2.34}$$

where Δt is the time step length, $\mathbf{X}_n$ and $\mathbf{X}_{n+1}$ are the state vectors at times t_n and t_{n+1} and θ is a parameter which has limits $0 \leq \theta \leq 1$. Substituting (2.33) and (2.34) into (2.32) produces:

$$(\mathbf{B} + \mathbf{A}\Delta t\theta)\mathbf{X}_{n+1} = (\mathbf{B} - \mathbf{A}\Delta t(1-\theta))\mathbf{X}_n + \mathbf{C}\Delta t. \tag{2.35}$$

The final discrete equations take the form:

$$\begin{bmatrix} -\mathbf{Ke} & \mathbf{L} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{S} + \Delta t\theta\mathbf{H} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{P}}_w \end{bmatrix}_{n+1} = \begin{bmatrix} -\mathbf{Ke} & \mathbf{L} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{S} - (1-\theta)\Delta t\mathbf{H} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{u}} \\ \overline{\mathbf{P}}_w \end{bmatrix}_n + \Delta t \begin{bmatrix} -\mathbf{F}^u \\ \mathbf{F}^p \end{bmatrix}. \tag{2.36}$$

These equations are used to determine displacement and pressure at any time relative to their initial values.

2.5.2 Incremental Load

If $\Delta\mathbf{F}^u$ is the change in load between successive times, the incremental form of equation (2.23) is given by:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{u} \\ \Delta\overline{\mathbf{P}}_w \end{bmatrix} + \begin{bmatrix} -\mathbf{Ke} & \mathbf{L} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{S} \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \Delta\overline{\mathbf{u}} \\ \Delta\overline{\mathbf{P}}_w \end{bmatrix} = \begin{bmatrix} -\Delta\mathbf{F}^u \\ \mathbf{F}^p \end{bmatrix}, \tag{2.37}$$

where $\Delta\bar{\mathbf{u}}$ and $\Delta\bar{\mathbf{P}}_w$ are the incremental changes in displacement and excess pore pressure respectively. Linear interpolation in time using the θ-method to the displacement vector yields:

$$\Delta\bar{\mathbf{u}} = \Delta t\left((1-\theta)\left[\frac{d\bar{\mathbf{u}}}{dt}\right]_n + \theta\left[\frac{d\bar{\mathbf{u}}}{dt}\right]_{n+1}\right). \tag{2.38}$$

Applying this interpolation Smith and Griffiths (2004) presented the following incremental relation:

$$\begin{bmatrix}\mathbf{Ke} & -\mathbf{L}\\ -\mathbf{L}^{\mathrm{T}} & \mathbf{S}-\Delta t\theta\mathbf{H}\end{bmatrix}\begin{bmatrix}\Delta\bar{\mathbf{u}}\\ \Delta\bar{\mathbf{P}}_w\end{bmatrix} = \begin{bmatrix}\Delta\mathbf{F}^u\\ \Delta t\mathbf{H}\bar{\mathbf{P}}_{w_n}+\mathbf{F}^p\end{bmatrix}. \tag{2.39}$$

At each time step, both displacement and excess pore pressure are updated as follows:

$$\bar{\mathbf{u}}_{n+1} = \bar{\mathbf{u}}_n + \Delta\bar{\mathbf{u}}, \tag{2.40}$$

$$\bar{\mathbf{P}}_{w_{n+1}} = \bar{\mathbf{P}}_{w_n} + \Delta\bar{\mathbf{P}}_w. \tag{2.41}$$

For both time discretization methods the value of θ determines the accuracy and stability of the time integration scheme. When $\theta = 1$ the scheme is fully implicit (backward Euler) and imposes no requirements on the time step size, Δt, chosen for both stability and the elimination oscillatory effects in the solution. The fully implicit scheme has been implemented in all numerical tests conducted in this thesis.

2.6 Consolidation Code Verification

Code verification is the process by which one demonstrates that the code correctly solves the fundamental governing equations (Knupp and Salari, 2003). Consolidation provides the foundation upon which the current work has been built, it is therefore essential to demonstrate that the implemented code solves

the governing equations correctly before further complexities are added to the model, specifically, fracture mapping and XFEM.

This section verifies the finite element consolidation code implemented within MATLAB®. The examples used for the code verification are one and two dimensional consolidation problems which have analytical solutions that can be found in literature. These analytical solutions have also been included in this section for ease of reference. The absolute load time discretization is used for examples presented in this chapter.

2.6.1 One-dimensional Consolidation

The finite element mesh used for the one dimensional plane strain consolidation example along with support conditions are given in Fig. 2.3. The domain is fully saturated and allowed to freely drain at the top (excess pore water pressure is equal to zero) and is assumed to be sealed on the right, left and bottom boundaries. A static load of 10kPa is applied at the top of the block at time $t = 0$ and maintained throughout the duration of the simulation. The block is homogenous and assumed to behave in a linearly elastic manner. The time step used, Δt, is 1day. Material properties are shown in Table 2.1. Comparisons between the numerical and analytical results for the surface settlement and excess pore pressure are shown in Fig. 2.4 and Fig. 2.5 respectively. These plots indicate that the code correctly solves the governing equations on both the coarse and refined finite element meshes.

The analytical solution for one dimensional consolidation of a porous domain of depth H, subjected to a total stress σ and supported on an impermeable base was presented by Terzaghi (1925).

The closed solution for the excess pore fluid pressure presented by Terzaghi is given by:

$$p = \Delta\sigma \sum_{m=0}^{\infty} \frac{2}{M} \sin\left(\frac{Mz}{H}\right) \exp(-M^2 T_v). \tag{2.42}$$

and the degree of consolidation U_t is

$$U_t = 1 - \sum_{m=0}^{\infty} \frac{2}{M^2} \exp(-M^2 T_v), \tag{2.43}$$

where the parameters are defined by

$$M = \frac{1}{2}\pi(2m + 1), \tag{2.44}$$

$$T_v = \frac{c_v t}{H^2}, \tag{2.45}$$

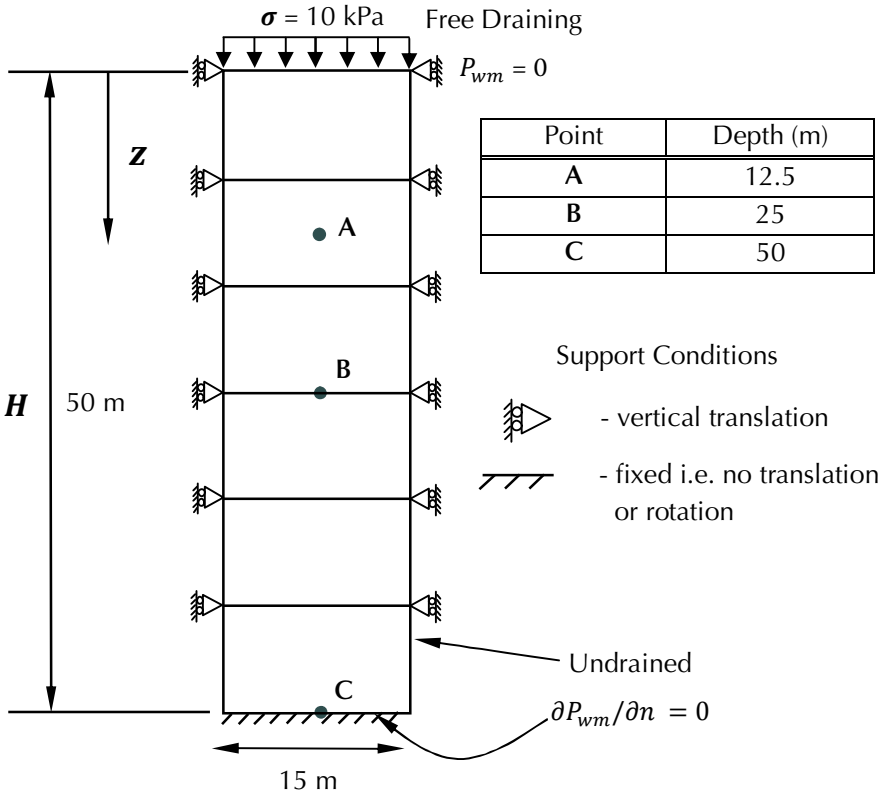
$$c_v = \frac{k}{\gamma_w m_v}, \tag{2.46}$$

$$m_v = \frac{(1 + \nu)(1 - 2\nu)}{E(1 - \nu)}. \tag{2.47}$$

The parameters T_v, c_v and m_v represent the time factor, coefficient of consolidation and coefficient of compressibility respectively.

The deformation at the surface S_t at any time t is given by:

$$S_t = U_t m_v \Delta\sigma H. \tag{2.48}$$



Point	Depth (m)
A	12.5
B	25
C	50

Fig. 2.3. Domain used for one-dimensional consolidation example.

Table 2.1. Material properties for consolidation example.

Parameter	Definition	Magnitude	Units
E	Modulus of Elasticity	40	MPa
ν	Poisson's ratio	0.3	-
n	Matrix porosity	0.2	-
$\mathbf{K}_1/\mu$	Matrix mobility	1×10^{-6}	$m^2/(MPa \cdot s)$

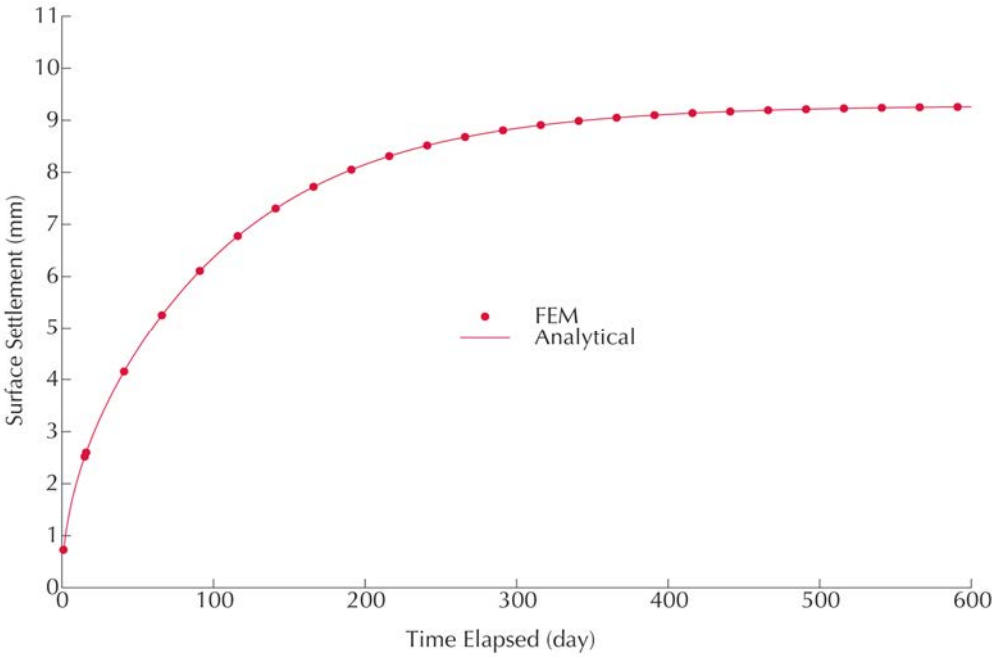


Fig. 2.4. Surface settlement variation with time.

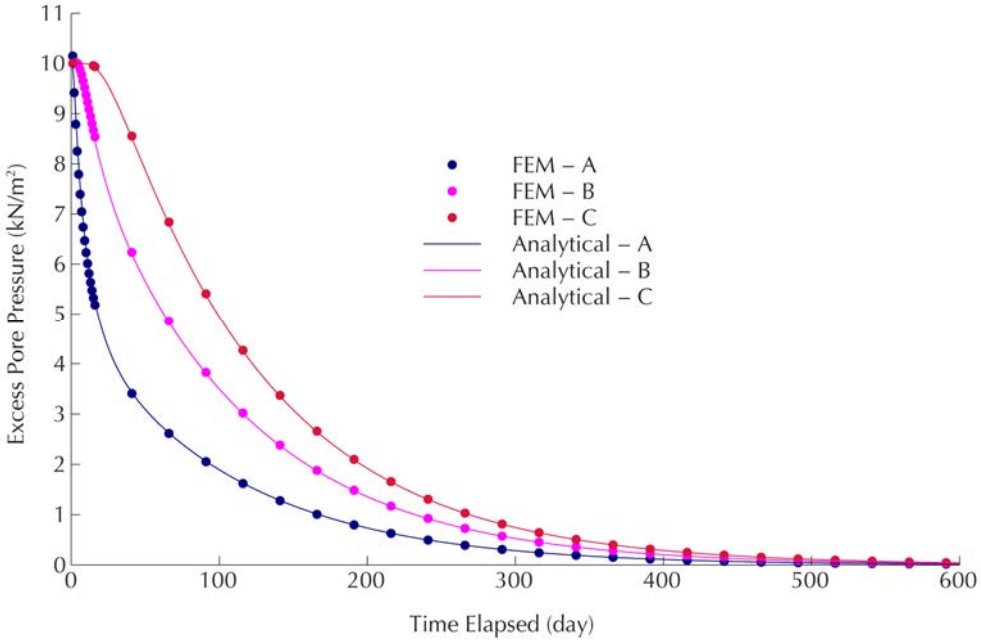


Fig. 2.5. Excess pore pressure variation with time at various locations in the domain (see Fig. 2.3).

2.6.2 Two-dimensional Consolidation

Cryer's ball (Cryer, 1963) was used for the two dimensional FE consolidation example. Cryer developed an analytical solution for a drained sphere of elastic porous material having an applied constant surface pressure. He found that the pore pressure at the centre of the sphere exceeded the externally applied pressure, and then reverses and dissipates. This non-monotonic behaviour, due to poroelastic coupling, is termed the Mandel-Cryer effect.

The Mandel-Cryer or stress transfer effect is characterised by an increase in excess pore water pressure within a poroelastic body above the initial stress condition at early times in the consolidation process. This behaviour can be explained by the contraction at the drained boundaries inducing a pore pressure build-up within the body. This phenomena was first identified by Mandel (1953), who presented an analytical solution for uniaxially loaded body under plane strain conditions. Gibson *et al.* (1963) also observed the Mandel-Cryer effect on laboratory experiments conducted on several clay spheres.

The closed solution for the excess pore water pressure at the centre of the sphere is given by (Cryer, 1963):

$$P(0,s) = \frac{\sinh\sqrt{s} - \sqrt{s}}{(s + 4\mu_c)\sinh\sqrt{s} - 4\mu_c\sqrt{s}\cosh\sqrt{s}}, \tag{2.49}$$

where s is the Laplace parameter and μ_c is a dimensionless constant and given by :

$$\mu_c = \frac{\mu}{\lambda + 2\mu}, \tag{2.50}$$

where λ and μ are the first and second Lamé parameters of the soil skeleton respectively. The closed form solution for surface displacement of the sphere is given by:

$$U(1,s) = \frac{3-4\mu}{s} \frac{\sqrt{s}\cosh\sqrt{s} - \sinh\sqrt{s}}{(s+4\mu_c)\sinh\sqrt{s} - 4\mu_c\sqrt{s}\cosh\sqrt{s}}, \tag{2.51}$$

The Laplace transform closed form solutions for the excess pore pressure and displacement were determined numerically using the MATLAB® function invlap.m (Hollenbeck, 1998).

The axisymmetric nature of the sphere allows the ball to be reduced to a two dimensional problem (Fig. 2.6). The material properties used in the 1D consolidation example are also used for the sphere. The sphere is fully saturated, homogenous and assumed to behave in a linearly elastic manner. The finite element meshes used for the numerical simulations are shown in Fig. 2.7.

The numerical and analytical results for the excess pore pressure at the centre of the ball and the surface settlement plotted against the dimensionless time factor T_v are shown in Fig. 2.8 and Fig. 2.9 respectively. The results display good agreement with the analytical solution. The rate of convergence for the mixed formulation is shown in Fig. 2.10.

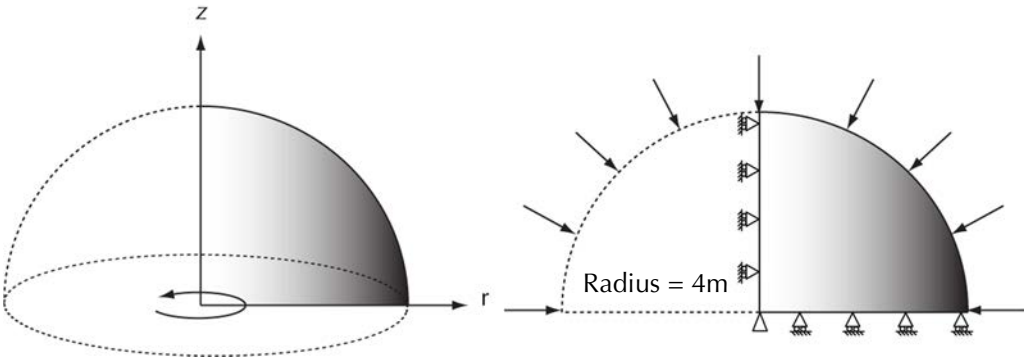


Fig. 2.6. 3D-axisymmetric problem reduced to a 2D problem.

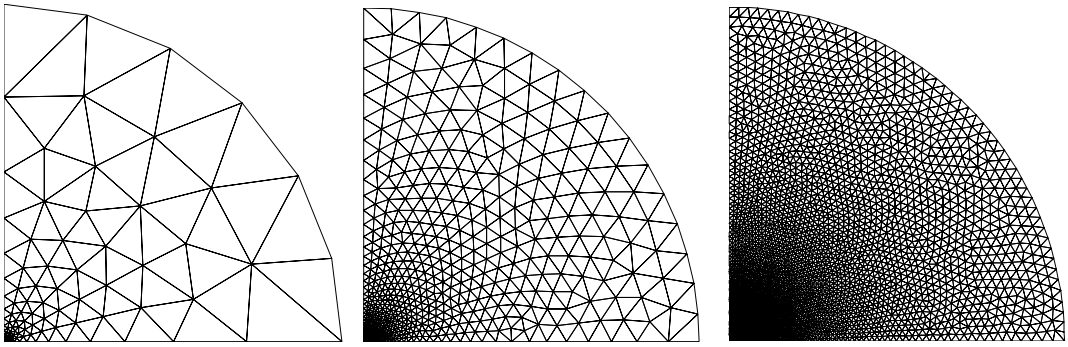


Fig. 2.7. Varying finite element meshes used for numerical simulations.

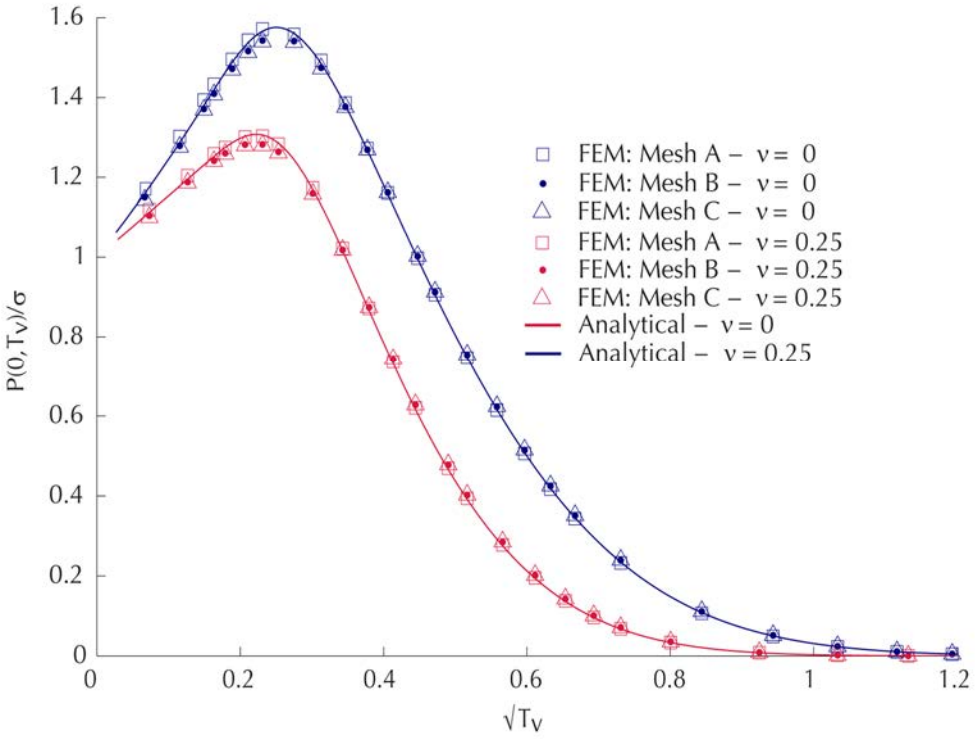


Fig. 2.8. Excess pore pressure at the centre of Cryer's consolidation ball vs dimensionless time.

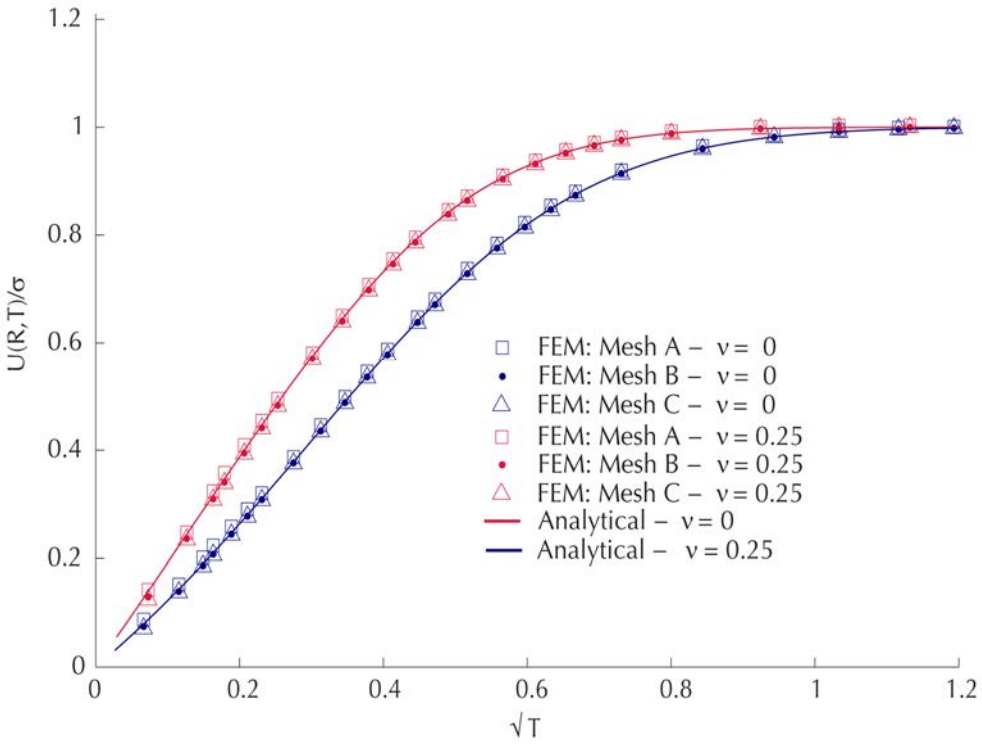


Fig. 2.9. Surface displacement of Cryer's consolidation ball vs dimensionless time.

The error measure of the coupled formulation used to determine both displacement and pressure within the domain was calculated as the sum of the L_2-norm of the displacement gradients and the L_2-norm of the pressure and is given by (Hughes, 2000):

$$\|\mathbf{u}_{num} - \mathbf{u}_{ex}\|_1 + \|\mathbf{p}_{num} - \mathbf{p}_{ex}\|_0 = O(h^{\min(k,l+1)}), \tag{2.52}$$

where $\mathbf{u}_{num}$ and $\mathbf{p}_{num}$ are the numerically determined displacement and pressure values respectively; $\mathbf{u}_{ex}$ and $\mathbf{p}_{ex}$ are the exact values of the displacement and pressure respectively; k and l are the orders of the displacement and pressure interpolations and have the values of 2 and 1 respectively. A rate of convergence equal to 2 is therefore "optimal" for this system. The norms in (2.52) are given by:

$$\|\mathbf{u}_{ex} - \mathbf{u}_{num}\|_1 = \left(\int_\Omega (\mathbf{u}_{ex} - \mathbf{u}_{num})^2 + (\nabla \mathbf{u}_{ex} - \nabla \mathbf{u}_{num})^2 d\Omega \right)^{1/2}, \quad (2.53)$$

$$\|\mathbf{p}_{ex} - \mathbf{p}_{num}\|_0 = \left(\int_\Omega (\mathbf{p}_{ex} - \mathbf{p}_{num})^2 d\Omega \right)^{1/2}, \quad (2.54)$$

These norms are normalised by the norm of the exact solution before determining the rate of convergence and are given by (Fish and Belytschko, 2007):

$$\frac{\|\mathbf{u}_{ex} - \mathbf{u}_{num}\|_1}{\|\mathbf{u}_{ex}\|_1} = \frac{\left(\int_\Omega (\mathbf{u}_{ex} - \mathbf{u}_{num})^2 + (\nabla \mathbf{u}_{ex} - \nabla \mathbf{u}_{num})^2 d\Omega \right)^{1/2}}{\left(\int_\Omega (\mathbf{u}_{ex})^2 + (\nabla \mathbf{u}_{ex})^2 d\Omega \right)^{1/2}}, \quad (2.55)$$

$$\frac{\|\mathbf{p}_{ex} - \mathbf{p}_{num}\|_0}{\|\mathbf{p}_{ex}\|_0} = \frac{\left(\int_\Omega (\mathbf{p}_{ex} - \mathbf{p}_{num})^2 d\Omega \right)^{1/2}}{\left(\int_\Omega (\mathbf{p}_{ex})^2 d\Omega \right)^{1/2}}, \quad (2.56)$$

The rate of convergence is given by the gradient of the log-log plot of the L_2 error norm vs. the nodal spacing h. The rate of convergence obtained for the coupled scheme is 2.085.

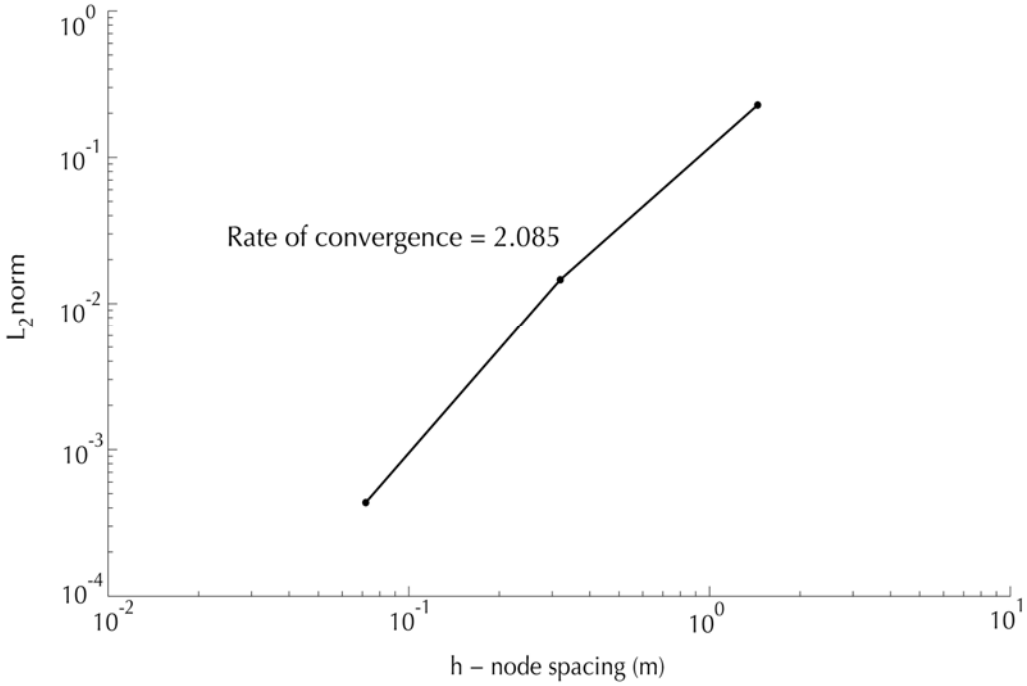


Fig. 2.10. Rate of convergence of implemented finite element code for Cryer's consolidation ball.

2.7 Conclusion

This chapter has provided an overview of the consolidation process followed by a description of the governing equations that describe coupled deformation and fluid flow in porous media. Examples presented within the chapter demonstrate that the finite element code implemented correctly solves the governing equations which describe the physical system of coupled deformation and fluid flow. This code provides the foundation for the fracture mapping – extended finite element code (FM-XFEM) which is presented in chapter 4.

CHAPTER 3

Fracture Mapping

This chapter presents a new fracture mapping approach developed as part of this work to handle the fluid flow interaction between the porous matrix and the fractures. The mapping approach uses a transfer function to account for the fluid flow interaction between the matrix and fractures contained within the domain. The fracture geometry is defined using level sets. In contrast to the discrete fracture model (DFM) the fracture representation is not meshed along with the computational domain. The method can therefore effectively determine the influence of fracture geometry and orientation on fluid flow within a fractured domain without the computational resource requirements of DFM. The formulation is presented along with finite element and meshfree numerical experiments to demonstrate the validity of the approach.

3.1 Introduction

A porous medium can be described as a matrix of solid material with interconnected pores, with at least several continuous paths (Bear, 1988). These pores are void-spaces within the medium that may contain and conduct fluid. When fractures are also present, the medium then consists of two distinct pore spaces which are represented by the void-spaces between fracture walls and the

pore spaces within the intact matrix blocks. Several conceptual methods have been developed to quantitatively estimate the flow behaviour in fractured porous media. The methods that have dominated this area of research are the equivalent continuum (Berkowitz *et al.*, 1988), dual continuum (Barenblatt *et al.*, 1960), discrete fracture model (Karimi-Fard *et al.*, 2004) and discrete fracture network (Lee *et al.*, 1999; Lee *et al.*, 2001). These methods fall under two main categories; continuum methods and discrete fracture methods.

The equivalent continuum approach treats the fractured medium as a single continuum with its hydraulic properties (porosity and permeability) averaged over the sub domains. The averages are determined for a representative elementary volume (REV) which must contain a large number of fractures. This approach is well suited for simulating large-scale flow since the heterogeneities resulting from the existing fractures are smoothed out over a very large area (Huyakorn *et al.*, 1983; Samardzioska and Popov, 2005). Although this approach is simple to implement and requires low computational resources, it cannot reliably estimate fluid flow in localised areas around fractures.

Barenblatt *et al.* (1960) introduced the dual continuum (DC) concept by modelling flow through a non-deformable, fractured porous medium. The dual continuum concept provides a mathematical framework for the fluid flow interaction between the matrix domain and fractures and was originally applied to the field of reservoir engineering by Warren and Root (1963) to characterize and simulate fluid flow in naturally fractured reservoirs. The dual continuum concept can be subdivided into the dual porosity and dual permeability approach (Diodato, 1994). The dual porosity approach assumes that the porous

matrix is discontinuous and flow within the domain is achieved via the fractured network. The simulated fluid flow is therefore directly affected by fracture network connectivity and distribution. In contrast, the dual permeability model assumes that the matrix is continuous and fluid flow occurs within both the matrix and fractures (Fung, 1991; Diodato, 1994; Zhang and Sanderson, 2002). Fracture network connectivity is not a prerequisite when using the dual permeability method. DC approaches are very simple to implement; however since these methods rely on averaged properties they cannot adequately simulate fluid flow in highly heterogeneous domains. Since the approach does not consider fracture geometry and orientation and are unable to accurately determine the influence large scale fractures have on fluid flow within a domain.

The discrete fracture model (DFM) has received considerable interest over the last few years in the field of reservoir simulation and hydrology (Lee *et al.*, 1999; Kim and Deo, 2000; Karimi-Fard *et al.*, 2004; Matthäi *et al.*, 2007; Tran and Ravoof, 2007; Geiger-Boschung *et al.*, 2009). In this approach fractures are represented within the domain in a spatially explicit manner and are meshed along with the matrix domain. Fractures are therefore represented by 1D line elements or 2D triangular/quadrilateral elements for a 2D simulation model. For 3D simulation models fractures are modelled using 2D or 3D elements. Although this method considers real fracture geometry, size and orientation, its demands on computer storage for models containing hundreds or thousands of fractures limits its applicability to domains with low fracture density.

Generating computational meshes incorporating fractures are also challenging and time consuming to generate.

The discrete fracture network method (DFN) explicitly models fractures without including the porous matrix. Since fracture permeability is usually several orders of magnitude larger than the matrix permeability it is assumed that the matrix is impervious and therefore eliminated from the model (Lee *et al.*, 2001). Flow within the model is restricted to the fracture network therefore fracture connectivity is necessary for successful implementation. The method, as with the discrete fracture method, is also computationally intensive.

3.2 Modern approaches to fracture flow modelling

Efforts to combine both continuum and discrete methods have been previously undertaken by Svensson (2001a, b), Mckenna and Reeves (2006) and Botros *et al.* (2008). Svensson (2001a, b) represented fracture networks as grid cell permeabilities in a finite- difference framework. This resulted in a 16% error in the simulated flow when compared to the DFN flow model. The error has been attributed to the ratio between the fracture width, W, and grid cell size, Δ, which must be greater than 0.1.

Mckenna and Reeves (2006) developed a technique called the fracture continuum model (FCM) which assumed critical properties of the fracture network to be spatially random variables that are assigned to finite-difference grid cells. The critical properties considered are the number of fractures within the computational domain, a measure of the fracture connectivity

(coordination number) and domain permeability. Although this model is easily implemented, deriving connectivity distributions and the number of the fractures within a grid block is not a straightforward task. The transformation of distributions of fracture length, aperture, orientation and fracture spacing obtained from borehole observations and pressure testing is a subjective matter and can significantly affect the flow simulation results (Botros *et al.*, 2008).

The method proposed by Botros *et al.* (2008) maps fractures directly onto a finite-difference grid in a manner similar to that outlined by Svensson (2001a, b). However, there is no restriction on the ratio of fracture width to cell size. Correction factors are used to account for the orientation and length of fractures contained within a finite-difference cell. This approach uses the sum of the cell permeability and permeability of the fracture contained within the cell for flow calculations. Although the method retains original fracture geometry and benefits from the processing speed of continuum models the method overestimates the flow through the porous network when compared to DFN. Botros *et al.* (2008) has stated that this overestimation is due to the enhanced permeability of matrix grid cells caused by the applied correction factor. Another limitation of the method is the fact that it has been developed for regular Cartesian finite difference grids and its applicability is yet to be extended to irregular grids.

The new fracture mapping (FM) technique developed as part of this research combines the merits of both discrete and continuum approaches; fracture geometry is preserved without having to mesh existing fractures within the simulation domain. In contrast to the discrete fracture model, fractures do

not need to conform to the computational mesh. Elements intersected by a fracture are treated as two overlapping elements; a matrix element and a fracture element. The orientation and permeability fractures are then mapped to the fracture element. The approach adopts the transfer function presented by Barenblatt *et al.* (1960) to account for fluid flow interaction between the overlapping matrix and fracture elements. Individual fractures are incorporated into the simulation domain using the level set method (LSM) introduced by (Osher and Sethian, 1988). LSM provides a mathematical description of fractures within the domain rather than an explicit representation of the crack geometry. The method represents an interface, in this case a fracture, as the zero of a function called the level set function (Duflot, 2007). In this thesis the level set function used is the signed distance function, which has a value of zero at the fracture. Fracture representation using LSM is discussed further in section 3.4 of this chapter.

Within this thesis the fracture mapping has been implemented using the finite element method and a meshfree method known as the nodal integration radial point interpolation method (NI-RPIM). In the finite element implementation the fracture permeability is mapped unto a fracture element which interacts with a matrix element. The same approach is adopted with the meshfree method; however the interaction is done between nodes and not elements. The resulting fracture mapping ensures that realistic fracture geometry is retained and the method adequately estimates the flow within a fractured domain when compared to the DFM.

The developed mapping approach is presented in the next section and is followed by a brief description of the level set method used to describe the fractures. The discretized flow equations are then presented along with details of the meshfree implementation procedure. Numerical examples are then presented to test the validity of the FEM and meshfree implementations of the mapping approach.

3.3 Transfer Function Mapping

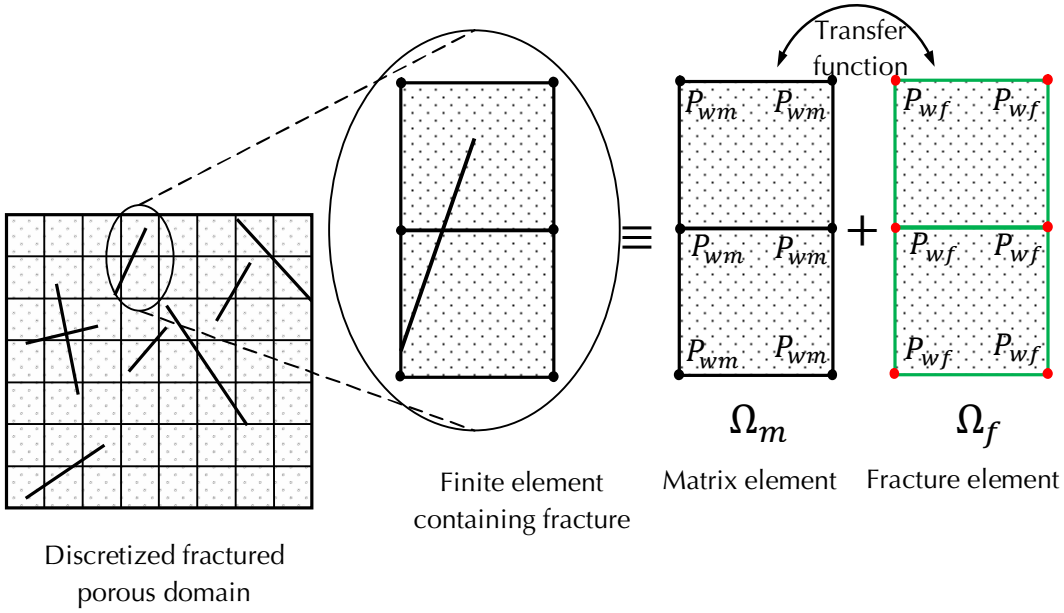
This section presents the fracture mapping approach developed as part of this work. In this approach elements intersected by a fracture are treated as the superposition of two sets of elements, matrix and fracture elements, which interact via a transfer function (Fig. 3.1). The transfer function provides a mathematical description of the fluid flow interaction between the matrix and the fracture. The transfer function adopted in this study is based on the quasi steady-state function proposed by Barenblatt *et al.* (1960), which assumes that the flow within an element is directly proportional to the difference between the matrix pore pressure and the fracture fluid pressure. The transfer function implemented is a 2x2 matrix given by:

$$\mathbf{T}_r = \frac{\bar{s}\mathbf{K}_f}{\mu_w}\left(P_{wm} - P_{wf}\right), \tag{3.1}$$

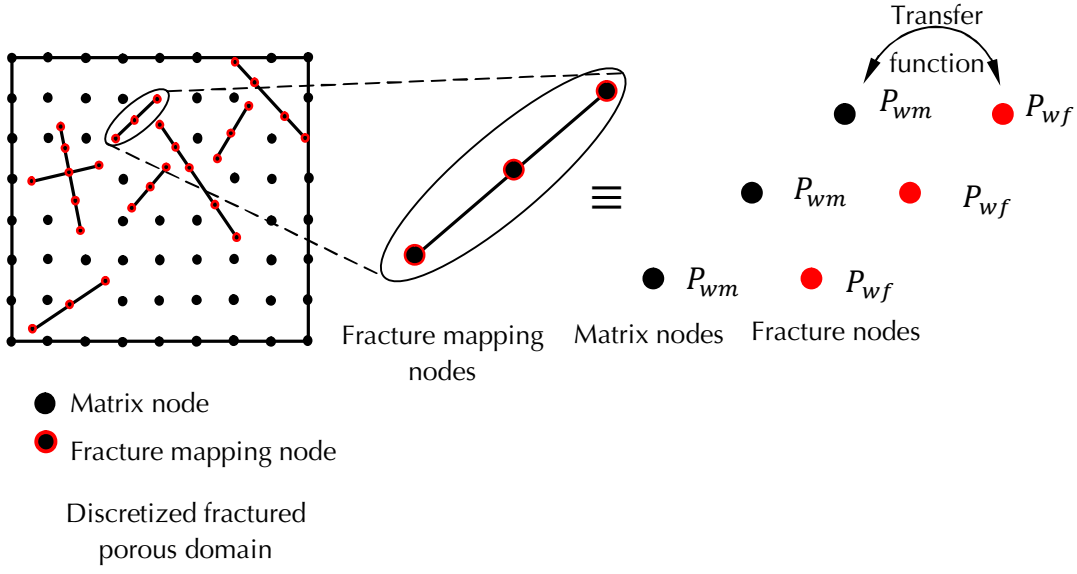
where $\mathbf{K}_f$ is the fracture permeability tensor, μ_w is the fluid viscosity, P_{wm} is the matrix fluid pore pressure, P_{wf} is the fracture fluid pressure. In the original transfer function presented by Barenblatt *et al.* (1960), $\bar{s}$ represents a shape factor which accounts for the fracture size and geometry. Since the fracture mapping presented in this work describes the size and geometry of a fracture within a fracture element, the shape factor has only been retained to ensure that the dimensional consistency of the transfer function is equivalent to the original formulation. The variable $\bar{s}$ has dimension L^{-2}, where L represents length. $\bar{s}$ was assigned the value of 1 in this work.

The amount of the fluid transferred between the matrix and fracture elements is characterized by the fracture permeability $\mathbf{K}_f$. Section 3.5 illustrates

the incorporation of the transfer function into the governing equations which describe fluid flow within the matrix and the fracture. It should be noted that since fracture elements share mesh nodes with neighbouring fracture elements the fluid pressure is continuous across these elements (see Fig. 3.1).



(a) Finite element discretization and fracture mapping.



(b) Meshfree discretization and fracture mapping.

Fig. 3.1. Schematic representation of the matrix and fracture elements/nodes extracted from discretized fractured domain.

When fractures intersect within an element the fracture mapping is accomplished by creating a fracture element for each fracture that exists within the element (Fig. 3.2).

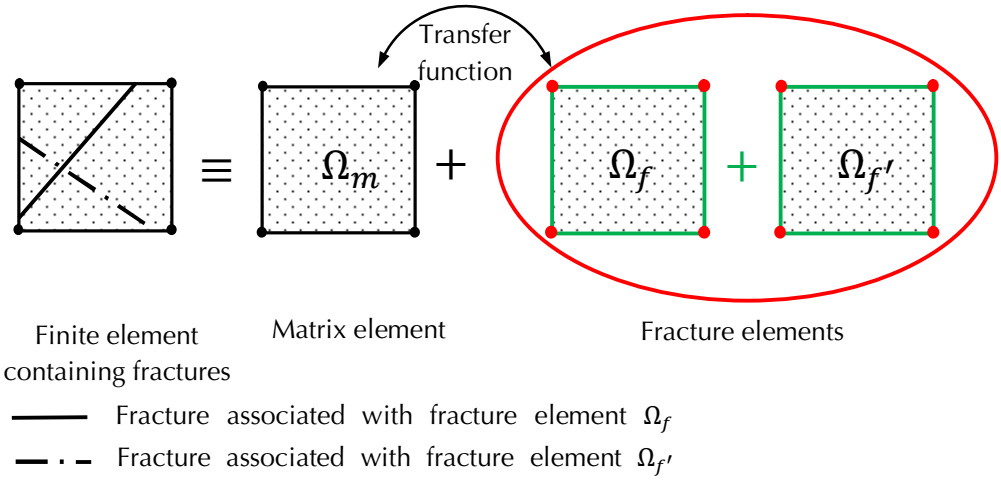


Fig. 3.2. Schematic representation of fracture mapping for intersecting fractures.

The fracture permeability is represented within a fracture element through the following mapping:

$$\boldsymbol{K}_{fe} = \frac{A_f}{A_e}\boldsymbol{K}'_f, \tag{3.2}$$

where $\mathbf{K}_{fe}$ is the fracture element permeability, A_f is the area of the fracture within the element, A_e is the area of the element and $\mathbf{K}'_f$ is the fracture permeability tensor aligned to the global Cartesian axes (Fig. 3.3). This mapping equates the total flow through the fracture to the total flow through the fracture element and ensures an effective representation of the fracture flow within the fracture element. When using the meshfree implementation the fracture permeability rotation is carried out on the nodes that describe the

fracture.

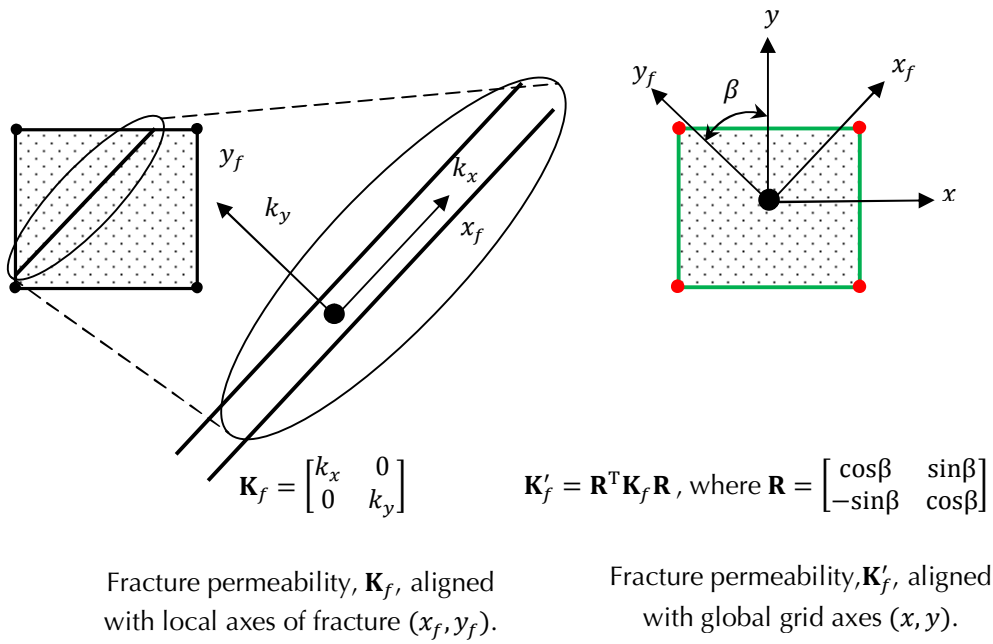


Fig. 3.3. Fracture permeability is aligned with the Cartesian axes using the rotation tensor R.

3.4 Fracture Representation

The level set method (LSM), commonly used in conjunction with XFEM to represent discontinuities instead of using an explicit representation, has been adopted to describe the geometry of fractures contained within the domain for the fracture mapping approach (Stolarska *et al.*, 2001; Sukumar *et al.*, 2001; Gravouil *et al.*, 2002; Moës *et al.*, 2002; Stazi *et al.*, 2003; Ventura *et al.*, 2003; Duflot, 2007; Prabel *et al.*, 2007). The fracture is represented as the zero level set of the signed distance function $\phi(\mathbf{x})$, i.e. by:

$$\phi(\mathbf{x}) = 0. \tag{3.3}$$

The signed distance function is defined by

$$\phi(\mathbf{x}) = \text{sign}[\mathbf{n} \cdot (\mathbf{x} - \bar{\mathbf{x}})] \min_{\bar{\mathbf{x}} \in \boldsymbol{\Gamma}_{\text{fr}}} \|\mathbf{x} - \bar{\mathbf{x}}\|, \tag{3.4}$$

where $\mathbf{x}$ is a finite element mesh node, $\bar{\mathbf{x}}$ is the normal projection of the point $\mathbf{x}$ onto the fracture defined by $\boldsymbol{\Gamma}_{\text{fr}}$ and $\mathbf{n}$ is the unit outward normal from the fracture at $\bar{\mathbf{x}}$ (see Fig. 3.4). The fracture tips are defined by tangential level set $\psi_i(\mathbf{x})$ given by:

$$\psi_i(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_i) \cdot \hat{\mathbf{t}}, \tag{3.5}$$

where $\hat{\mathbf{t}}$ is a unit vector tangent to the fracture at its tip and $\mathbf{x}_i$ is the location of the i^{th} fracture tip. When more than one fracture tip exists in the domain, the tangential level set can be represented by a single function and given as:

$$\psi(\mathbf{x}) = \max_i \ (\psi_i) \tag{3.6}$$

Using the signed distance function and tangential level set, a fracture in 2D is defined by:

$$\phi(\mathbf{x}) = 0 \text{ and } \psi(\mathbf{x}) \leq 0 \tag{3.7}$$

The level set functions, $\phi(\mathbf{x})$ and $\psi_i(\mathbf{x})$ and the representation of the fracture are also illustrated in Fig. 3.4.

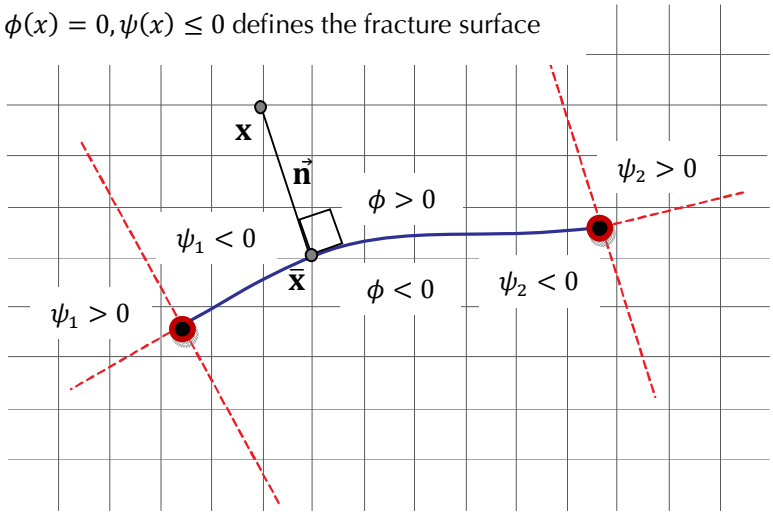


Fig. 3.4. Level set functions for 2D fracture representation.

3.4.1 Assignment of fractured elements

For a given fracture, whose input geometry is defined by a single line segment or a series of line segments, the values of $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ are calculated for all the finite element nodes within close proximity to the fracture (see Fig. 3.5). An element is defined as a fracture element if the values of $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ determined at its nodes satisfy the following conditions (Stolarska *et al.*, 2001):

$$\begin{cases} \phi_{max}\phi_{min} \leq 0, & \psi < 0, \text{ split element,} \\ \phi_{max}\phi_{min} \leq 0, & \psi_{max}\psi_{min} \leq 0, \text{ tip element.} \end{cases} \tag{3.8}$$

Split elements are defined as those elements that are completely cut a fracture and tip elements are those elements that contain a fracture tip. Fracture mapping is applied to both split and tip elements.

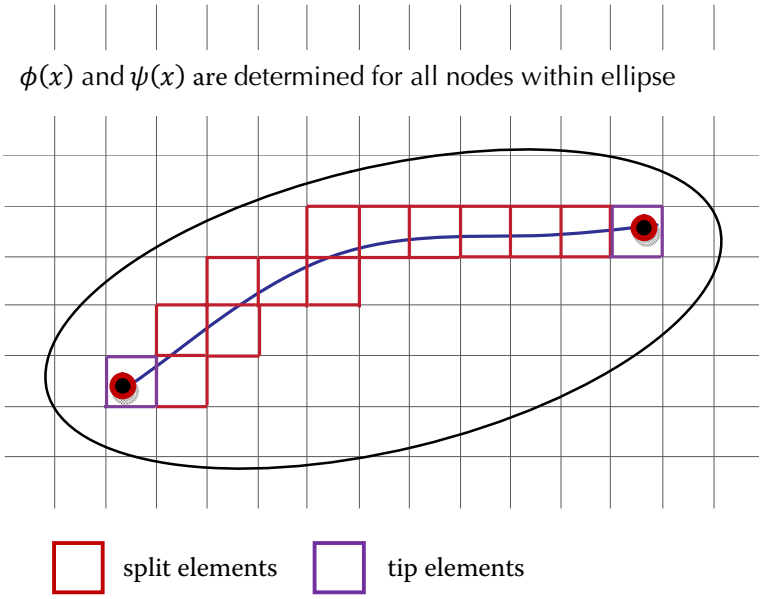


Fig. 3.5. Assignment of fractured elements using level set values.

3.5 Governing Equations of Single Phase Fluid Flow in Porous Media

Single phase fluid flow through porous media can be described by Darcy's law which is given by:

$$\boldsymbol{v}_d = \frac{\mathbf{K}}{\mu}(-\nabla P_w + \rho \mathbf{g}), \tag{3.9}$$

where $\boldsymbol{v}_d$ is Darcy's velocity, $\mathbf{K}$, is the spatially varying permeability tensor of the porous medium concerned, μ is the fluid viscosity, P_w is the pore fluid pressure, $\mathbf{g}$ is gravity and ρ is fluid density.

Continuity or mass conservation is a second important law in the development of the flow equations, i.e., for steady-state conditions, continuity

requires that the amount of fluid flowing into a control volume be equal to the amount flowing out. Therefore the continuity equation is given by:

$$\nabla \cdot \boldsymbol{v}_d = Q_w, \tag{3.10}$$

where Q_w is the specified source/sink term. Combining (3.9) and (3.10) results in the single phase, elliptical pressure equation:

$$\nabla \cdot \left(\frac{\mathbf{K}}{\mu} (-\nabla P_w + \rho \mathbf{g}) \right) = Q_w, \tag{3.11}$$

The transfer function $\mathbf{T}_r$ is then incorporated into equation (3.11) to account for the flow interaction between the matrix and fractures. The steady state flow equations then take the form:

$$\nabla \cdot \left(\frac{\mathbf{K}_m}{\mu} (-\nabla P_{wm} + \rho \mathbf{g}) \right) + \frac{\bar{\mathrm{s}} \mathbf{K}_f}{\mu} \left(P_{wm} - P_{wf} \right) = Q_w, \tag{3.12}$$

for the porous matrix, and,

$$\nabla \cdot \left(\frac{\mathbf{K}_f}{\mu} \left(-\nabla P_{wf} + \rho \mathbf{g} \right) \right) - \frac{\bar{\mathrm{s}} \mathbf{K}_f}{\mu} \left(P_{wm} - P_{wf} \right) = 0, \tag{3.13}$$

for the fractured domain. The transient flow equation is obtained by including the specific storage of the medium given by, n/K_w, assuming incompressible grains. The domain porosity is represented by n and K_w represents the bulk modulus of the fluid (see Istok (1989) for a detailed derivation of the specific storage coefficient). The transient flow equations take the form:

$$\nabla \cdot \left(\frac{\mathbf{K}_m}{\mu} (-\nabla P_{wm} + \rho \mathbf{g}) \right) + \frac{\bar{\mathrm{s}} \mathbf{K}_f}{\mu} \left(P_{wm} - P_{wf} \right) + \frac{n_m}{K_w} \frac{\partial P_{wm}}{\partial t} = Q_w, \tag{3.14}$$

for the porous matrix, and,

$$\nabla \cdot \left(\frac{\mathbf{K}_f}{\mu} \left(-\nabla P_{wf} + \rho \mathbf{g} \right) \right) - \frac{\bar{\mathrm{s}} \mathbf{K}_f}{\mu} \left(P_{wm} - P_{wf} \right) + \frac{n_f}{K_w} \frac{\partial P_{wf}}{\partial t} = 0, \tag{3.15}$$

for the fractured domain. The first numerical example in this chapter solves the transient pressure equation and the second example solves the steady state pressure equation.

3.5.1 Boundary conditions

When solving equations (3.14) and (3.15) boundary conditions are only applied to the matrix domain. These boundary conditions can be either Dirichlet or Neumann boundary conditions. Enforcing Dirichlet boundary conditions involves specifying constant pressures at domain boundaries or well locations. Neumann boundary conditions require specification of flow rates at the reservoir boundaries or wells.

3.6 Spatial Discretization

Applying the Galerkin-based finite method (see appendix B) to the steady state elliptic pressure equations for the porous matrix and the fractured domains given by equations (3.12) and (3.13), neglecting the effects of gravity, the resulting discretized equations take the form:

$$\mathbf{H}_m\overline{\mathbf{P}}_{wm} + \mathbf{H}_{T_r}\left(\overline{\mathbf{P}}_{wm} - \overline{\mathbf{P}}_{wf}\right) = \mathbf{Q}, \tag{3.16}$$

$$\mathbf{H}_f\overline{\mathbf{P}}_{wf} - \mathbf{H}_{T_r}\left(\overline{\mathbf{P}}_{wm} - \overline{\mathbf{P}}_{wf}\right) = \mathbf{0}. \tag{3.17}$$

Equations (3.16) and (3.17) can be conveniently cast into matrix form and re-written as:

$$\begin{bmatrix} \mathbf{H}_m + \mathbf{H}_{T_r} & -\mathbf{H}_{T_r} \\ -\mathbf{H}_{T_r} & \mathbf{H}_f + \mathbf{H}_{T_r} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{P}}_{wm} \\ \overline{\mathbf{P}}_{wf} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \mathbf{0} \end{bmatrix}, \tag{3.18}$$

where

$\mathbf{H}_m$ is the porous medium permeability matrix,

$\mathbf{H}_f$ is the fracture permeability matrix,

$\mathbf{H}_{T_r}$ is the transfer matrix,

$\mathbf{Q}$ is the flow field vector,

$\overline{\mathbf{P}}_{wm}$ and $\overline{\mathbf{P}}_{wf}$ are vectors representing the nodal pressures for the matrix and fracture respectively. The elements of these matrices are given below.

$$\mathbf{H}_m = \int_{\Omega} \left(\nabla \mathbf{N}_p\right)^{\mathrm{T}} \frac{\mathbf{K}_m}{\mu_w} \nabla \mathbf{N}_p \, d\Omega, \tag{3.19}$$

$$\mathbf{H}_f = \int_{\Omega} \left(\nabla \mathbf{N}_p\right)^{\mathrm{T}} \frac{\mathbf{K}_f}{\mu_w} \nabla \mathbf{N}_p \, d\Omega, \tag{3.20}$$

$$\mathbf{H}_{T_r} = \int_{\Omega} \mathbf{N}_{\mathrm{p}}^{\mathrm{T}} \frac{\bar{\mathrm{s}}\mathbf{K}_f}{\mu_w} \mathbf{N}_p \, d\Omega, \tag{3.21}$$

$$\mathbf{Q} = \int_{\Omega} \mathbf{N}_{\mathrm{p}}^{\mathrm{T}} Q_w \, d\Gamma, \tag{3.22}$$

$$\nabla \mathbf{N}_p = \begin{bmatrix} \dfrac{\partial}{\partial x} \mathbf{N}_p \\ \dfrac{\partial}{\partial y} \mathbf{N}_p \end{bmatrix}. \tag{3.23}$$

3.7 FEM Implementation

This section summarizes the implementation procedure for the FEM fracture mapping. The FEM fracture mapping involves the following steps:

1. Mesh generation: The domain is divided into a series of finite elements, quadrilateral or triangular.
2. Calculate the level sets of nodes in the vicinity of the line segments which describe the fracture geometry, see section 3.4;

3. Determine fracture elements using the equality conditions presented in section 3.4.1;
4. Loop over all elements (matrix assembly):
    a. Determine element contributions to discrete system equation (3.14): if an element is a fracture element, calculate element permeability using the procedure outlined in section 3.3. Element contributions are also made to the fracture permeability and transfer matrix ($\mathbf{H}_f$ and $\mathbf{H}_{T_r}$) for fracture elements;
    b. Add element contributions to the global system matrix
5. Apply boundary conditions;
6. Solve system equations to obtain field variable, in this instance nodal pore pressures.

Meshfree methods have been developed to eliminate mesh dependence by constructing the numerical approximation entirely in terms of nodes.

3.8 Meshfree Implementation

A meshfree implementation of the fracture mapping approach has also been developed as part of this work. The meshfree approach allows the inclusion of fractures within the domain without a dependence on the computational mesh. This reduces the need for complex meshes for complicated geological formations as well as localised mesh refinement required around existing fractures. Using the meshfree method the numerical approximation is constructed independently of a finite element mesh and is based entirely on the nodes used to discretize the domain. It should be noted that all meshfree

methods require cell generation for numerical integration of the approximation over the domain and the term meshfree highlights the fact that the approximation/shape functions are determined without the need of a finite element mesh. The issue of mesh quality which is associated with the task of finite element mesh generation does not factor into the generation of these background cells.

This section outlines the implementation of the nodal integration radial point interpolation method (NI-RPIM - Liu *et al.*, 2007) used in the meshfree fracture mapping portion of this study. The implementation of the NI-RPIM has the following steps:

1. Domain discretization: nodes are used to describe both the domain and existing fracture. Domain pore pressures are approximated at these node locations;
2. Generate Voronoi cells for nodal integration: Cells are generated using a MATLAB® Executable (MEX) interface to Voro++ (Rycroft *et al.*, 2006);
3. Loop over all nodes:
    a. Determine area of Voronoi Cell along with first and second moment area of cell corresponding to current node;
    b. Determine shape functions associated with current node;
    c. Evaluate nodal contributions to the discrete system equation (3.14): nodes that describe the fracture contribute to the fracture permeability and transfer matrix ($\mathbf{H}_f$ and $\mathbf{H}_{T_r}$);
    d. Assemble nodal contribution of current node to form global system matrices.

4. Apply boundary conditions;
5. Solve system equations to obtain field variable, in this instance nodal pore pressures.

Since steps 1-3 differ significantly from the standard procedures used in the finite element method, they are described in a bit more detail below.

STEP 1: Domain discretization

The problem domain and its boundary are represented using nodes scattered in the problem domain and on its boundary (Fig. 3.6); these nodes may have a regular or irregular distribution. Since these nodes carry the values of the field variables (displacement, pressure or velocity); in a meshfree formulation, they are often referred to as field nodes. The number of nodes used within the problem domain depends on the accuracy required and resources available. The nodal distribution can be irregular as meshfree methods work for arbitrary nodal distributions.

By comparison, the FEM requires meshing be performed to discretize the geometry and create elements of specific shapes. Information, such as element to node connectivity, also has to be created during meshing for later system assembly. Mesh generation is a very important part of the pre-processing stage of the finite element method. It is ideal to have an entirely automated mesh generator; unfortunately, it is not practically available for general situations with complex geometries.

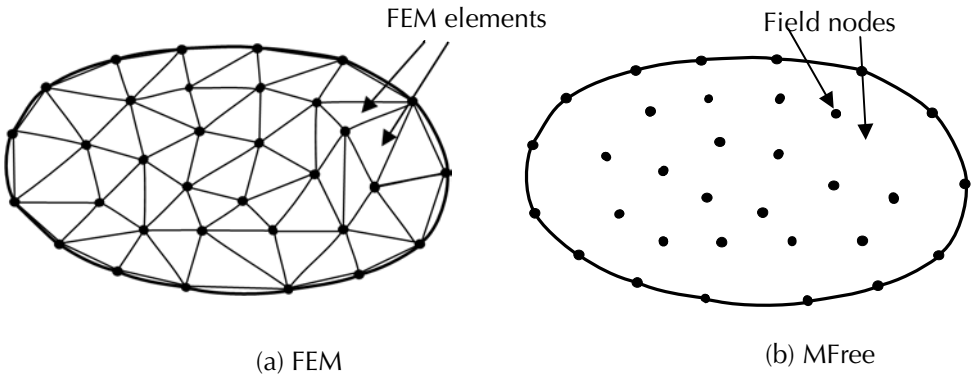


Fig. 3.6. Domain discretization (a) FEM (b) MFree.

STEP 2: Numerical integration cells

The finite element method relies on the use of Gaussian integration over finite elements cells to solve the discretized equations given by (3.14). Meshfree methods also employ Gaussian integration; however since these methods do not rely on a finite element framework, a background mesh independent of the nodal arrangement, is required for numerical integration. The aim of nodal integration schemes is to eliminate the need for a background mesh for numerical integration. These integration schemes use the field nodes that discretize the domain as the integration points for numerical integration. In their publication on the NI-RPIM, Liu *et al.* (2007) creates integration cells by connecting the centroids and mid-edge points of triangles formed from a tessellation of the field nodes. Since Voronoi cell generation is a straight forward process it was chosen over the tessellation method for the NI-RPIM implemented within this research work (Fig. 3.7). Mathematical details of the nodal integration procedure are given in Appendix C.

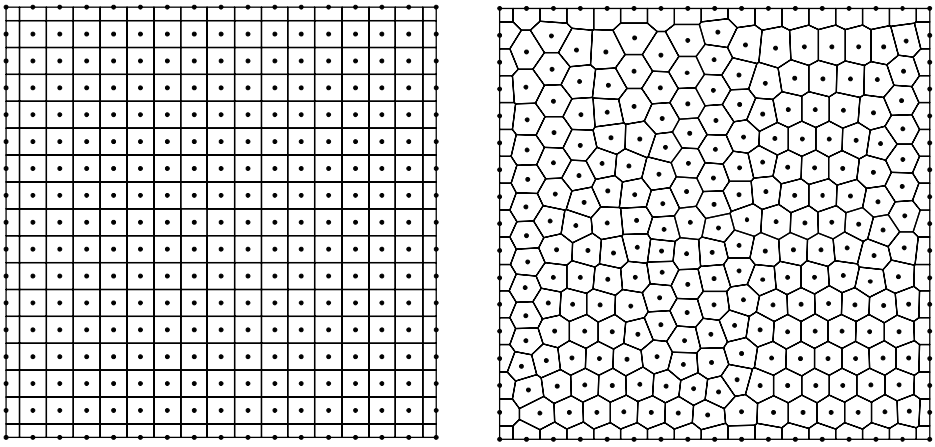


Fig. 3.7. Voronoi cells used for nodal integration.

STEP 3: Function interpolation/approximation

The meshfree shape functions are constructed during the simulation in contrast to the finite element for which shape functions are pre-defined for different elements. The shape function for a particular node is calculated based the nodes within its support domain. Fig. 3.8 illustrates the use of a circular support domain to determine the points to be used in the construction of the RPIM shape functions. The support domain can also take a rectangular form and is sized to ensure that a minimum of 10 points are captured within the domain; this guarantees that an adequate number of points are used for building the shape function. The derivation of the RPIM shape functions presented by Wang and Liu (2002) is given in Appendix C.

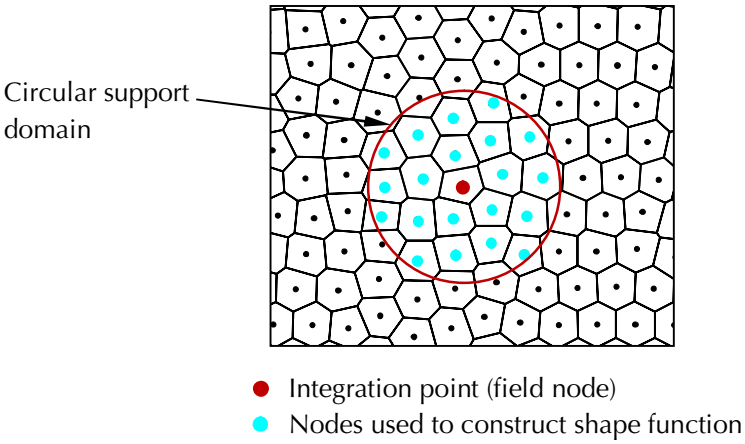


Fig. 3.8. Selection of supporting nodes required for shape function calculations.

3.9 Numerical Examples

Numerical experiments are presented to validate the fracture mapping approach. The first numerical example compares the FEM and meshfree solutions to the analytical solution for a vertical fracture intersecting a well in a closed reservoir. The second example is used to compare the results obtained using fracture mapping against those obtained from a discrete fracture model. The third example illustrates the resulting pressure field for a domain containing two intersecting fractures.

3.9.1 One well intersecting a vertical fracture

This example assesses the accuracy of the fracture mapping code by simulating a well fully penetrating a fractured confined reservoir. The model dimensions and properties are shown in Fig. 3.9. The fracture is aligned with the x-axis and is located at the centre of a square region of a confined reservoir. It is assumed that the fracture intersects a fully penetrating well at the centre of the reservoir.

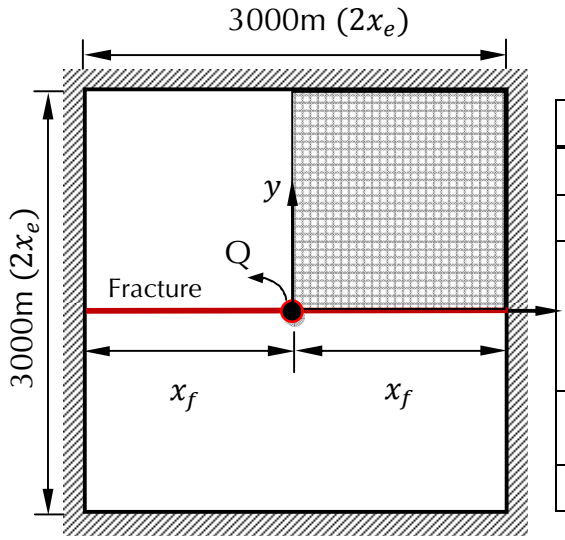
The analytical solution for this problem has been presented by Gringarten *et al.* (1974). The pressure drop on the fracture is given by:

$$P_w = 2\pi \int_0^{t_{DA}} \left[1 + 2\sum_{k=1}^{\infty} exp(-(2k\pi)^2 t_{DA})\right] \left[1 + 2\sum_{k=1}^{\infty} \exp(-(2k\pi)^2 t_{DA}) \frac{\sin\left(k\pi \frac{x_f}{x_e}\right)}{k\pi \frac{x_f}{x_e}} \cos\left(k\pi \frac{x_f}{x_e}\right)\right] dt_{DA}, \tag{3.19}$$

where t_{DA} represents the dimensionless time based on the drainage area and is given by:

$$t_{DA} = \frac{Tt}{4S_c(x_e y_e)^2}, \tag{3.20}$$

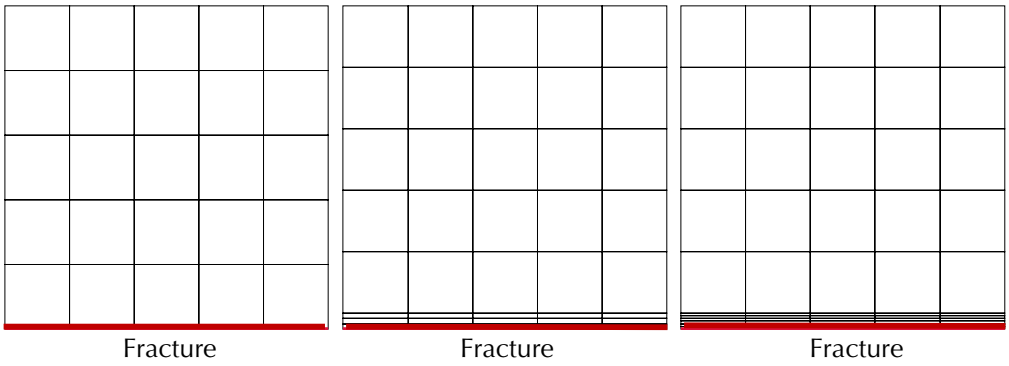
where T is the matrix transmissivity, t represents time and S_c is the matrix storage coefficient. x_f and x_e represent the fracture half length and rectangular reservoir half dimensions respectively (see Fig. 3.9). The pressure drop given in (3.19) was determined using the MATLAB® symbolic toolbox.



Property	Magnitude	Units
T	50	m^2/d
S	0.005	-
κ_f (fracture hydraulic conductivity)	108	m/d
Q (well discharge)	6280	m^3/d
x_f	1500	m

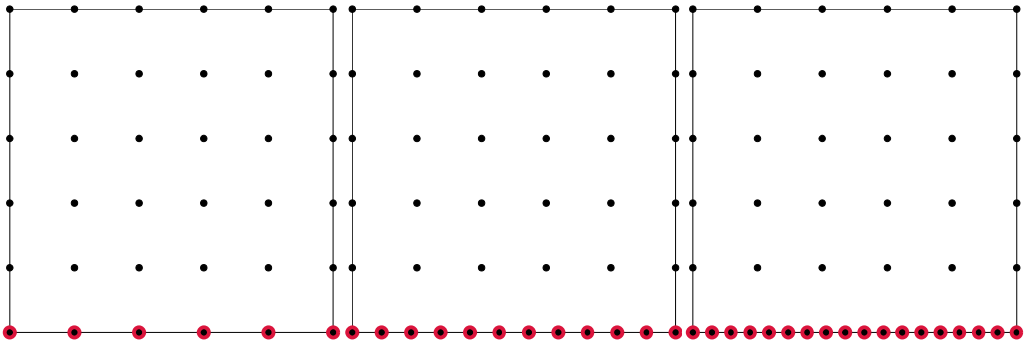
Fig. 3.9. Closed rectangular reservoir containing a well intersected by a vertical fracture. Due to domain symmetry only one-quarter of the domain is discretized for the numerical simulation. This region is represented by the hatched area.

Given the symmetry of the domain, only one-quarter of the domain has been discretized for the numerical simulations (see Fig. 3.9). The discretizations used for the numerical solutions are shown in Fig. 3.10. The finite element meshes ranges from a coarse mesh with 300m square Q4 elements to meshes containing a modest refinement along the fracture length. The refined elements extend to a length of 75m and the thicknesses of these elements are 25m and 12.5m respectively. For the meshfree implementation the refinement is obtained by increasing the number of nodes that describe the fracture (decreasing the distance between fracture nodes). A coarse discretization is used in this example to illustrate the effectiveness of the fracture mapping. A comparison of the analytical and numerical results for both methods is given in Fig. 3.11. The computed well drawdown vs dimensionless time for the discretization labelled C is given in Table 3.1 for both the FEM and meshfree implementation.



A) nodes:36 elements:25 B) nodes:54 elements:40 C) nodes:72 elements:55

(a) Finite element discretization



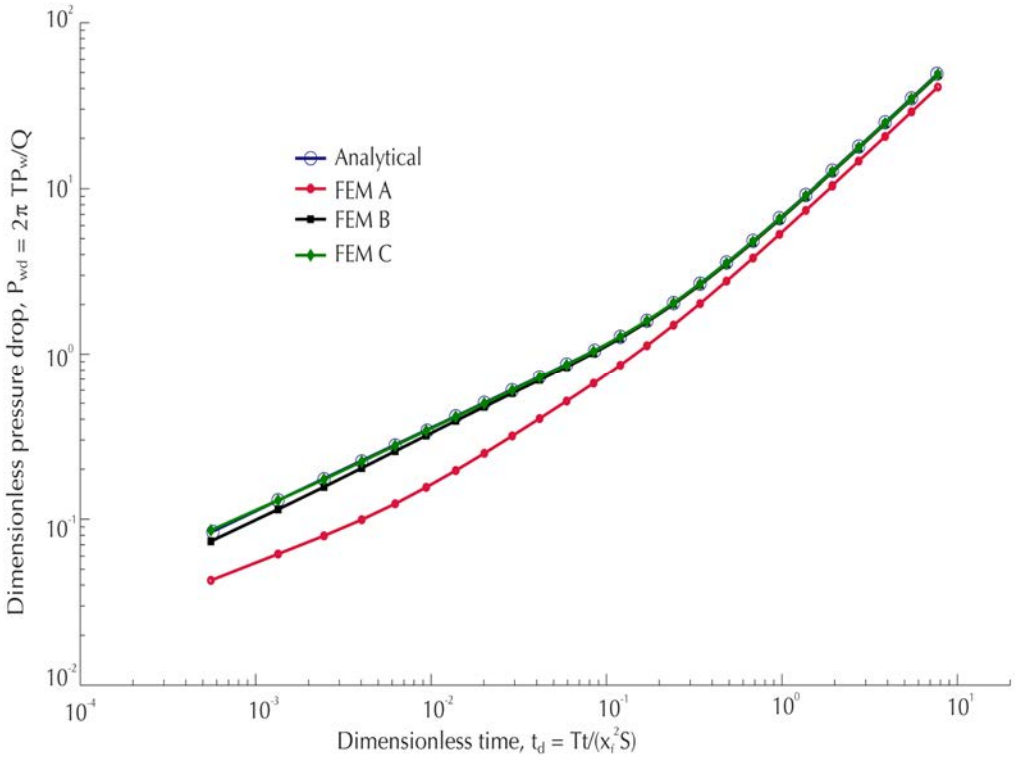
A) nodes:36 B) nodes:42 C) nodes:48

(b) Mfree discretization. The fracture is discretized using A) 6 nodes, B) 12 nodes and C) 18 nodes.

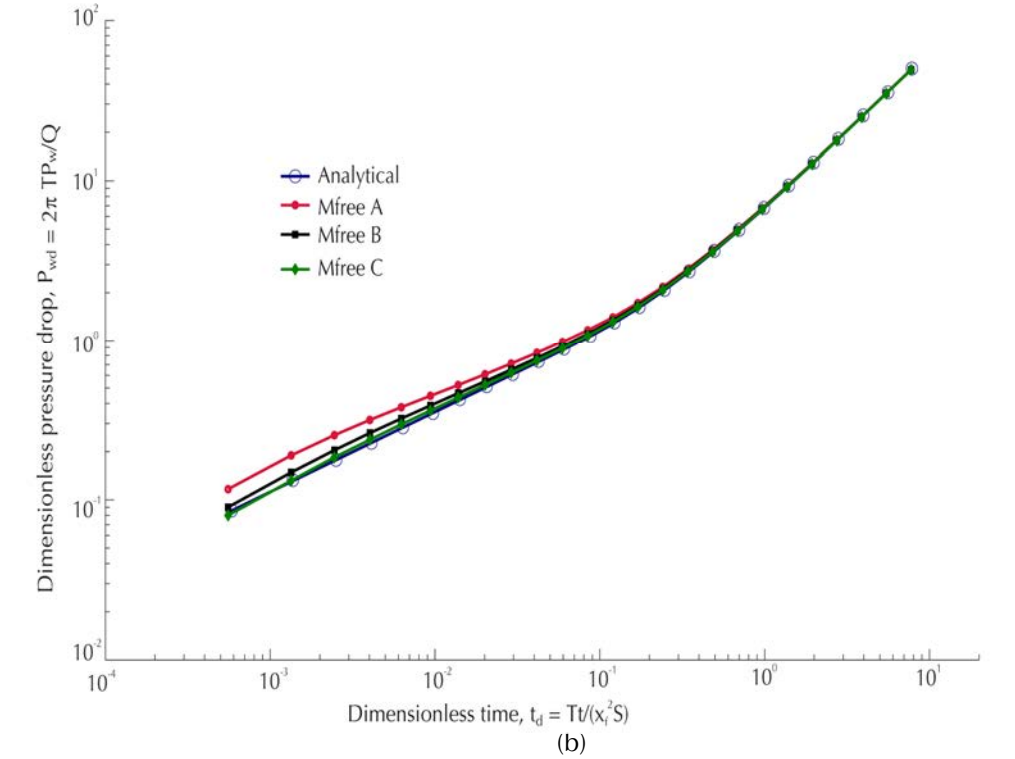
Fracture

Fracture node

Fig. 3.10. Domain discretizations used for the closed rectangular reservoir. Given the domain symmetry, only one-quarter of the domain has been discretized.



(a)



(b)

Fig. 3.11. Dimensionless well drawdown vs dimensionless time, showing comparison of numerical and analytical solutions. (a) FEM comparison, (b) meshfree comparison.

Table 3.1 Computed well drawdown vs dimensionless time.

Time t (days)	Dimensionless Time $t_d = \frac{Tt}{x_f^2 S_c}$	Dimensionless Pressure drop $P_{wd} = \frac{2\pi T P_w}{Q}$		
		Analytical	FEM	meshfree
0.50	5.56E-04	0.0835	0.0857	0.0811
1.21	1.34E-03	0.1298	0.1300	0.1342
2.21	2.45E-03	0.1755	0.1739	0.1848
3.62	4.02E-03	0.2248	0.2225	0.2372
5.62	6.24E-03	0.2801	0.2777	0.2944
8.45	9.38E-03	0.3434	0.3409	0.3586
12.44	1.38E-02	0.4168	0.4140	0.4325
18.09	2.01E-02	0.5026	0.4991	0.5185
26.08	2.90E-02	0.6035	0.5989	0.6195
37.38	4.15E-02	0.7227	0.7172	0.7392
53.36	5.93E-02	0.8654	0.8602	0.8840
75.95	8.44E-02	1.0424	1.0393	1.0653
107.89	0.12	1.2740	1.2736	1.3022
153.06	0.17	1.5917	1.5926	1.6242
216.92	0.24	2.0380	2.0370	2.0726
307.22	0.34	2.6684	2.6629	2.7038
434.92	0.48	3.5599	3.5471	3.5954
615.47	0.68	4.8204	4.7973	4.8558
870.78	0.97	6.6028	6.5649	6.6375
1231.78	1.37	9.1230	9.0643	9.1562
1742.23	1.94	12.6867	12.5985	12.7162
2464.02	2.74	17.7257	17.5959	17.7472
3484.62	3.87	24.8508	24.6622	24.8554
4927.75	5.48	34.9258	34.6539	34.8952
6968.34	7.74	49.1718	48.7822	49.0691

The numerical results obtained show good agreement with the analytical solution. NI-RPIM provides a good approximation of the problem using fewer nodes within the domain than the FEM implementation.

3.9.2 Discrete-fracture model comparison

In this example the steady state pressure equation (equations (3.12) and (3.13)) is solved on a fractured domain using the discrete fracture model (DFM) and the fracture mapping approach. The rates of convergence of each method are determined and the pressure field obtained within the domain is also compared. The model used in this numerical example is shown in Fig. 3.12.

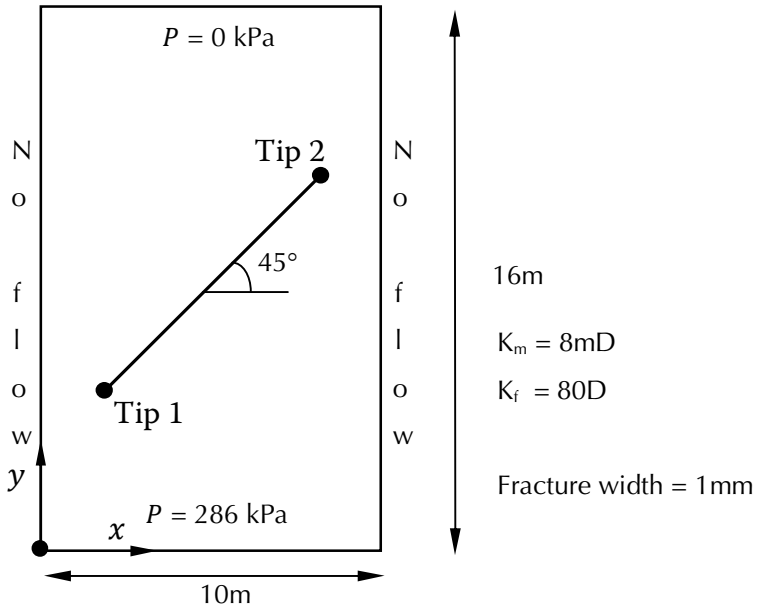


Fig. 3.12. 2D fractured domain with applied pressures at top and bottom boundary.

The finite element discrete fracture model in this example uses triangular, three noded elements (T3) to discretize the matrix and 2-noded linear line (L2) elements to discretize the fracture (Fig. 3.13).

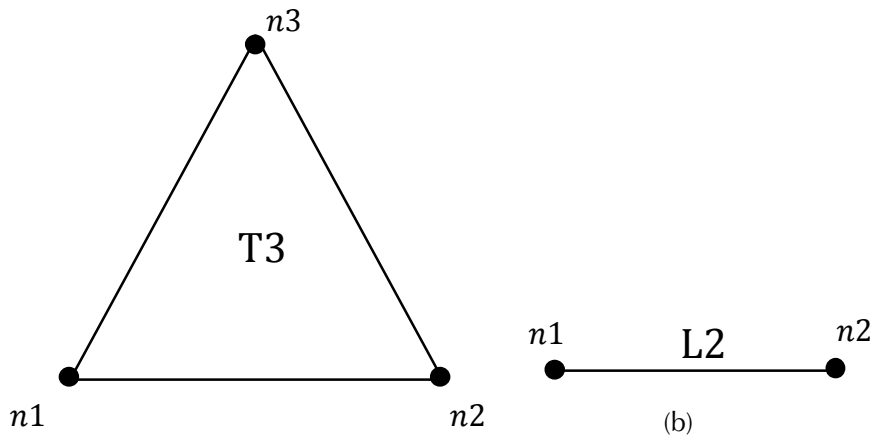
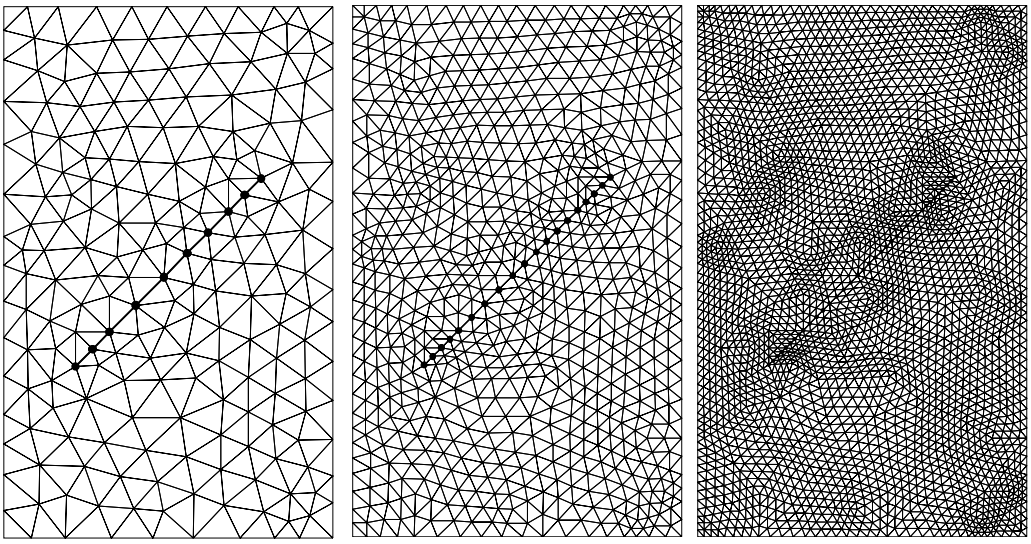


Fig. 3.13. Elements used for discrete fracture model. (a) Triangular element (T3) used to discretize the matrix and (b) line element (L2) used to discretize the fracture.

Prior to comparing DFM to the fracture mapping approach, the rate of convergence of DFM was determined. Three of the six meshes used to determine the rate of convergence of DFM are shown in Fig. 3.14.



(a) MeshA: 366 elements (b) MeshB: 1464 elements (c) MeshC: 5856 elements

Fig. 3.14. Mesh refinements used to determine rate of convergence of DFM.

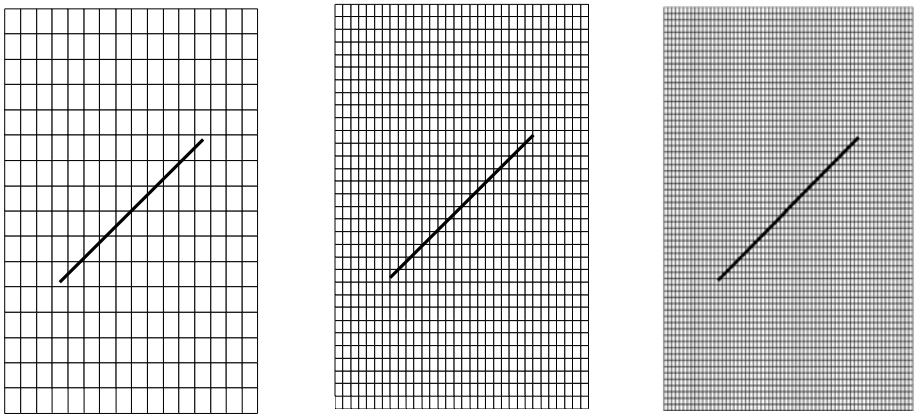
The mesh refinements used to determine the rate of convergence of DFM were obtained by dividing each element of the preceding mesh into four new elements using MeshA as the base mesh (Fig. 3.14.(a)). The mesh used as the reference mesh for the convergence study contained 374784 elements[3.1] (fifth refinement step). The third (23424 elements) and fourth (93696 elements) refinement steps were also used in determining the rate of convergence. The error was measured using the normalised L_2 norm defined by.

$$\frac{\|\mathbf{p}_c - \mathbf{p}_{ref}\|_0}{\|\mathbf{p}_{ref}\|_0} = \frac{\left(\int_\Omega (\mathbf{p}_c - \mathbf{p}_{ref})^2 d\Omega\right)^{1/2}}{\left(\int_\Omega (\mathbf{p}_{ref})^2 d\Omega\right)^{1/2}}, \tag{3.26}$$

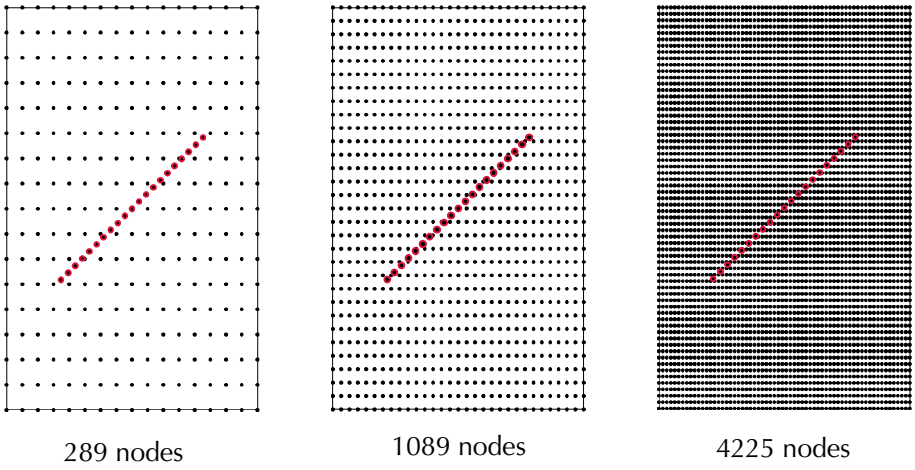
where $\mathbf{p}_c$ is the pressure determined from a coarse mesh and $\mathbf{p}_{ref}$ is the pressure determined on the reference mesh. The rate of convergence is given by the gradient of the log-log plot of the L_2 error norm vs. the nodal spacing h (Fig. 3.16). The rate of convergence was obtained using linear regression of the data points and is equal to 1.21.

Using the DFM reference solution the rates of convergence of both the FEM and Meshfree implementation of FM were also determined. The discretizations used to determine the rates of convergence are shown in Fig. 3.15. The number of nodes for the discretizations are 289, 1089, 4225 and 16641[3.1] respectively.

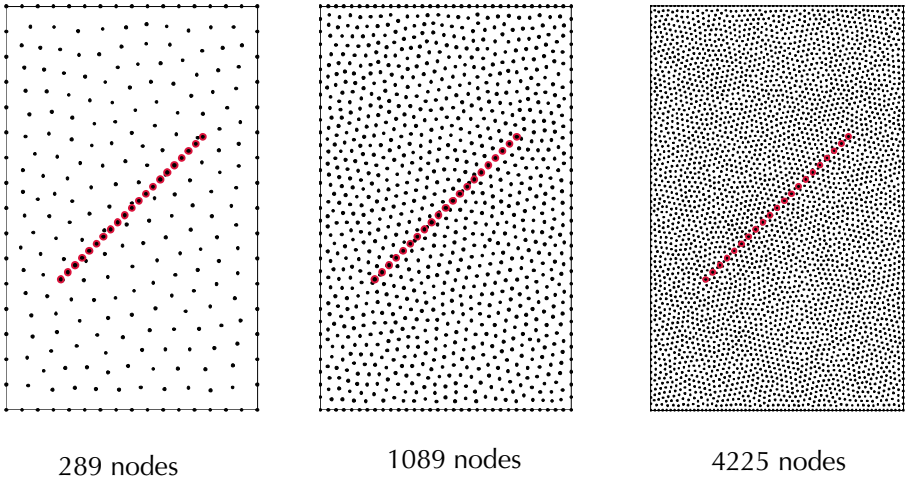
[3 1] At higher resolutions the discretizetion is not clearly shown and therefore has not been included.



(a) Finite element mesh resolutions.



(b) Meshfree regular node distribution resolutions.



(c) Meshfree irregular node distribution resolutions.

Fig. 3.15. Domain discretizations used to determine the rates of convergence of the FEM and Meshfree fracture mapping implementation.

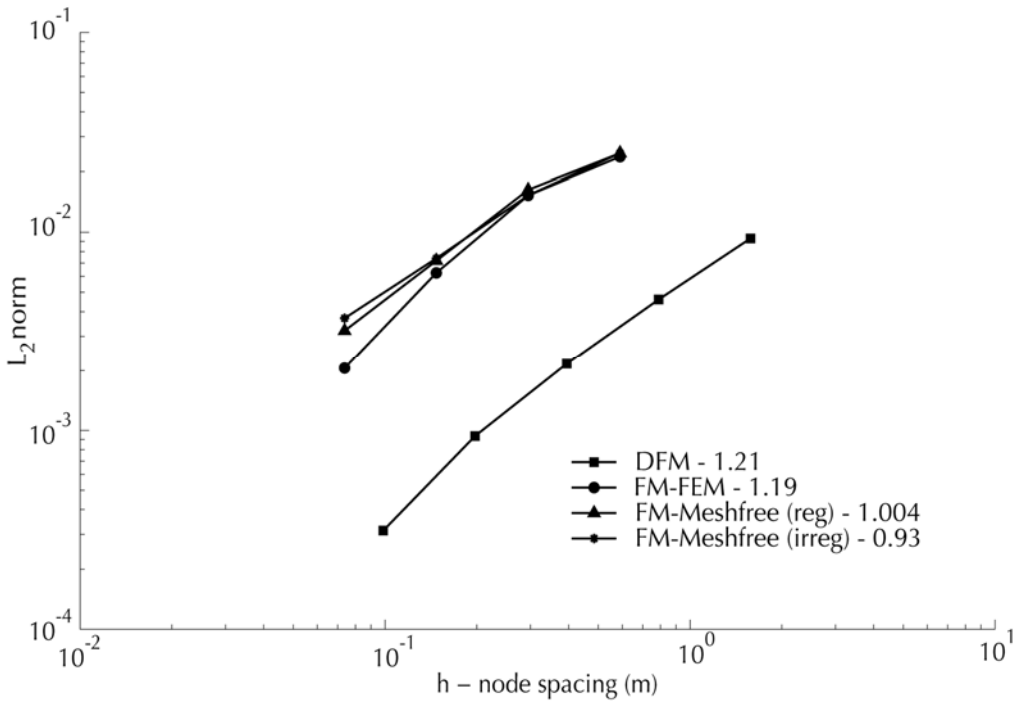
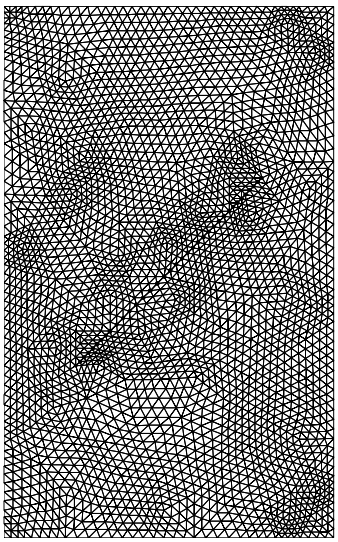


Fig. 3.16. Convergence rates for the discrete fracture model and the fracture mapping approach. The fracture mapping approach has been implemented using FEM and the Meshfree method for both regular and irregular nodal distributions.

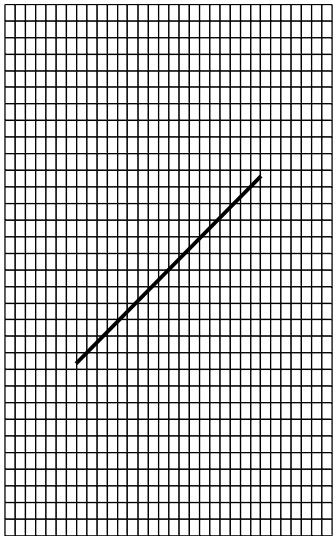
The rates of convergence of FM implementations are very similar to the rate of convergence of DFM. The meshfree implementation using an irregular nodal distribution provided the lowest convergence rate of 0.93 and indicates that the nodal distribution affects the rate of convergence of the meshfree numerical scheme. The comparative convergence rates obtained between DFM and FM implementations indicate that the fracture mapping approach solution accuracy is similar to that of DFM.

The pressure fields determined for the discrete fracture model and fracture mapping (FEM and meshfree) were also compared. The discretizations used to determine the pressure fields are shown in Fig. 3.17 and the resulting pressure fields are shown in Fig. 3.18. Since differences in the pressure fields obtained for the different methods are indistinguishable, a pressure profile along a

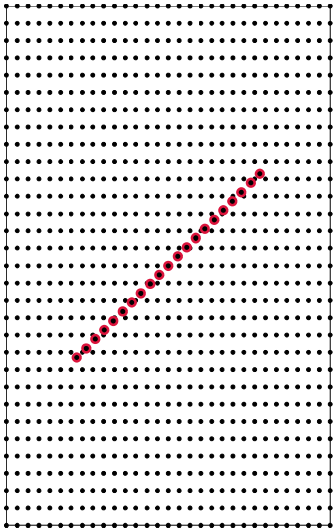
vertical section through the centre of the domain was created to compare the pressure change across the domain for each model.



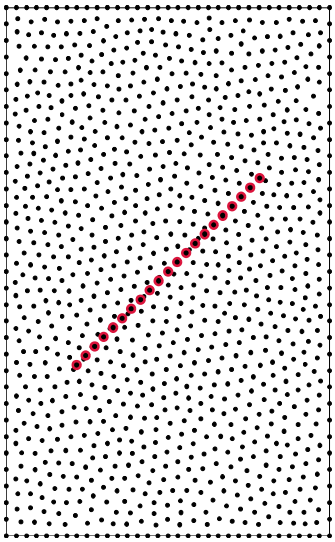
(a) DFM Mesh: 5856 elements



(b) FM-FEM Mesh: 1024 elements

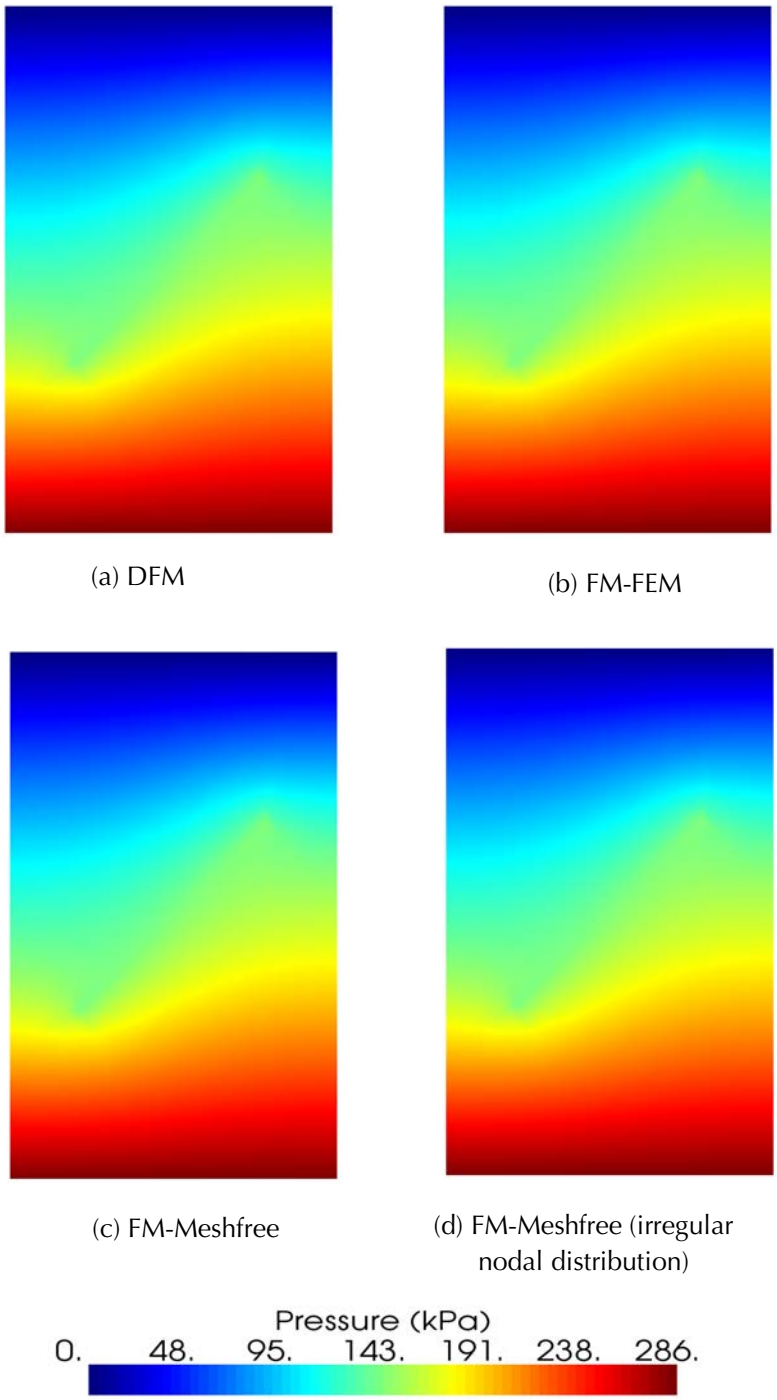


(c) FM-Meshfree: 1089 nodes



(d) FM-Meshfree (irregular nodal distribution): 1089 nodes

Fig. 3.17. Domain discretizations used to create comparative pressure fields.



(a) DFM

(b) FM-FEM

(c) FM-Meshfree

(d) FM-Meshfree (irregular nodal distribution)

Fig. 3.18. Pressure fields obtained for DFM and fracture mapping implementations.

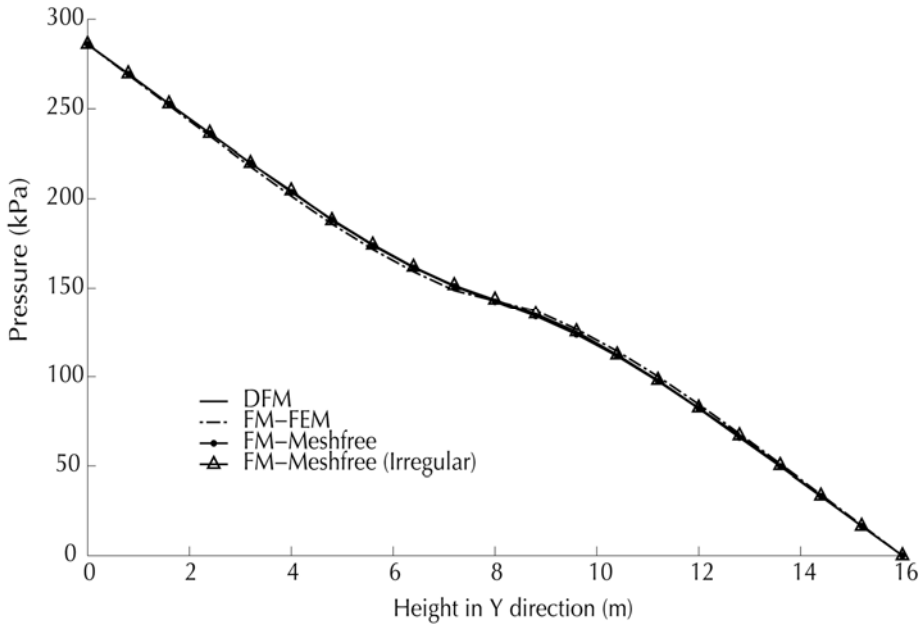


Fig. 3.19. Pressure profile for a vertical section taken through the centre of the fractured domain.

Using the discrete fracture model as a baseline for comparison, the fracture mapping method adequately describes the variation in pressure across the domain (Fig. 3.19). It should be noted that the fracture mapping models used to generate the pressure fields are all at discretization resolutions lower than that of DFM.

3.9.3 Intersecting Fractures

The model set-up for this example is the similar to the model given in section 3.9.1 with the inclusion of an additional fracture (Fig. 3.20). This example is used to compare the pressure fields obtained for a domain containing intersecting fractures using the discrete fracture model and the fracture mapping approach.

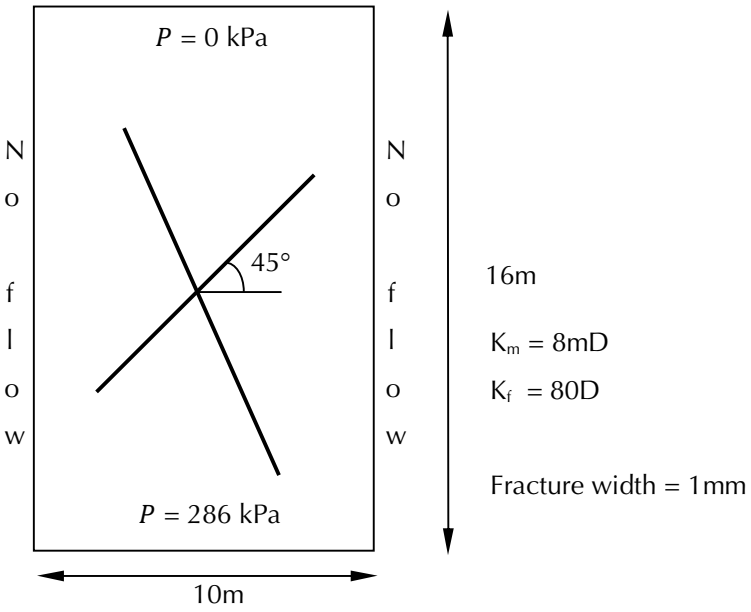
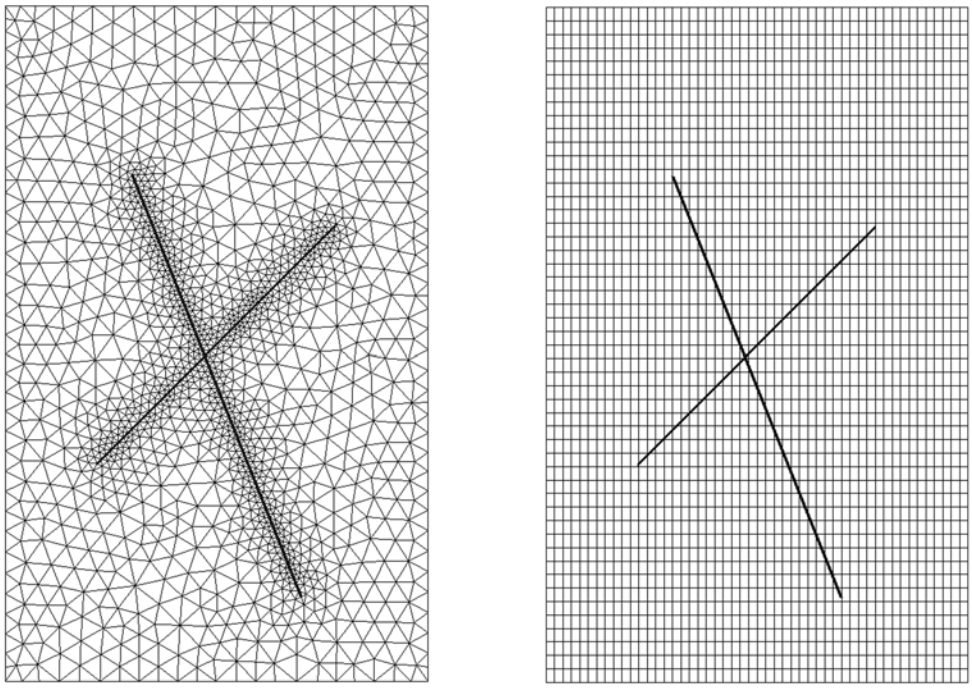


Fig. 3.20. 2D fractured domain containing intersecting fractures with applied pressures at top and bottom boundary.

The meshes used to generate DFM and FM solutions are shown in Fig. 3.21.



Discrete-fracture flow model:
3017 elements

Fracture mapping approach:
2500 elements

Fig. 3.21. Finite element meshes used for simulation.

The resulting pressure fields are shown in Fig. 3.22. The fracture mapping solution obtained for the intersecting fractures is also in good agreement with the discrete fracture model solution.

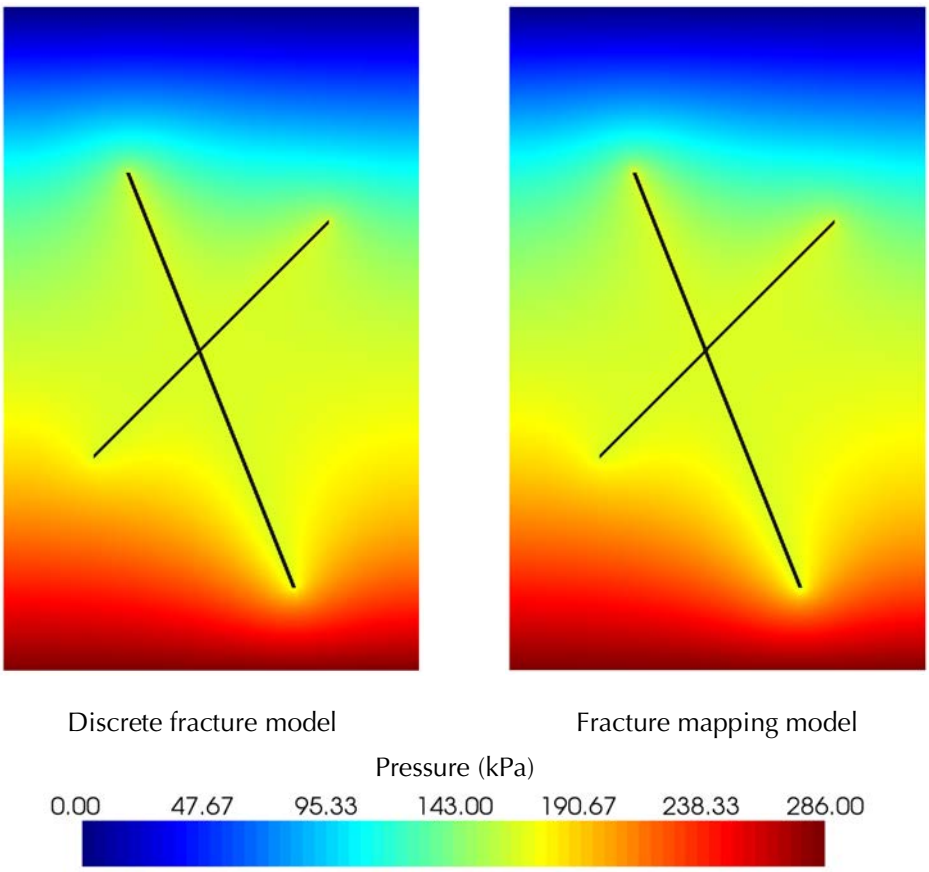
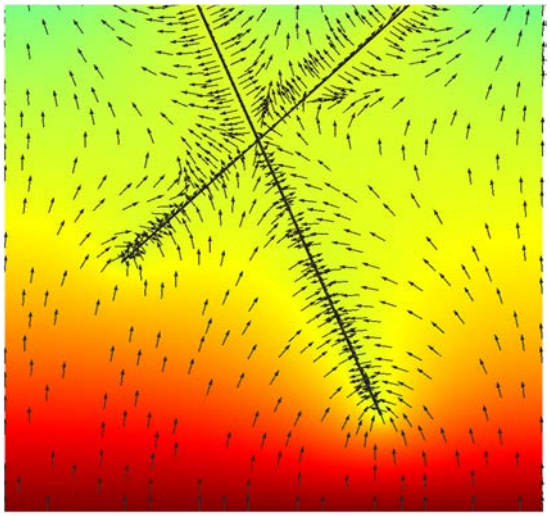


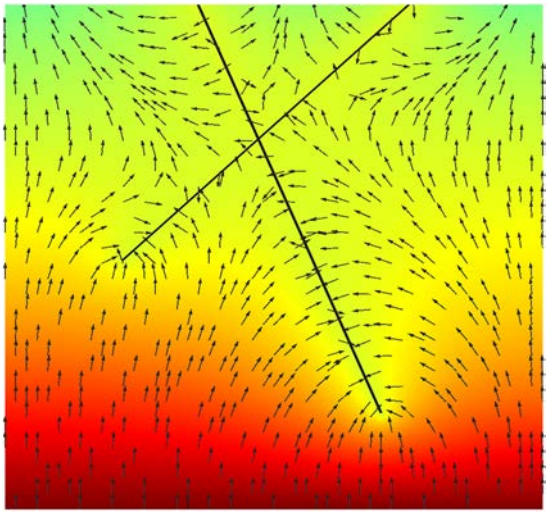
Fig. 3.22. Pressure fields for discrete fracture and fracture mapping models for intersecting fractures.

Fig. 3.23 shows a close up of the pressure fields shown in Fig. 3.22 and displays the flow velocity vectors within the domain. The figure is used to illustrate the influence of the highly permeable fractures on the direction of the fluid flow for both DFM and FM. The mapping adequately reproduces the flow field obtained

from the discrete fracture model. Fluid flow is directed towards the fractures in both models.



Discrete-fracture flow model



Fracture mapping model

Fig. 3.23. Influence of interacting fractures on the flow field direction for DFM and FM.

3.10 Conclusions

The fracture mapping presented is able to adequately simulate the fluid flow in a fractured porous domain without meshing the fracture. The method produces results that are in good agreement with analytical results presented in literature. Both the FEM and meshfree implementations perform adequately on coarse meshes when compared to analytical and discrete fracture model solutions. The developed method is also able to adequately resolve the fluid flow interaction between intersecting fractures and the matrix.

CHAPTER 4

FM-XFEM formulation

This chapter presents the fracture mapping-extended finite element method (FM-XFEM) for coupled deformation and fluid flow in fractured porous media. The method has been developed by modifying the single phase consolidation formulation discussed in chapter 2 using the fracture mapping approach presented in chapter 3 and the extended finite element method. The method performs well on coarse structured grids without having the computational mesh aligned with existing fractures. Comparison between the proposed method and the discrete fracture method (DFM) shows its ability to adequately determine the displacement and fluid pore pressure distribution within a fractured domain.

4.1 Modelling Discontinuities

Standard finite element approximations are piecewise differentiable polynomials that are not well suited for problems with strong and weak discontinuities (problems with a jump in the displacement and strain field respectively). To accurately model these discontinuities the finite element mesh must conform to the line of discontinuity (Fig. 4.1). This becomes a major difficulty for moving discontinuities (such as cracks, shear bands and faults)

because of the need to update the mesh to match the geometry of the discontinuity. The extended finite element method (XFEM) has been developed to address the difficulties associated with handling discontinuities within the finite element framework (Dolbow, 1999; Daux *et al.*, 2000; Chessa *et al.*, 2002; Moës *et al.*, 2002; Stazi *et al.*, 2003; Sukumar and Prévost, 2003; Budyn, 2004; Zi *et al.*, 2007; Bordas *et al.*, 2008).

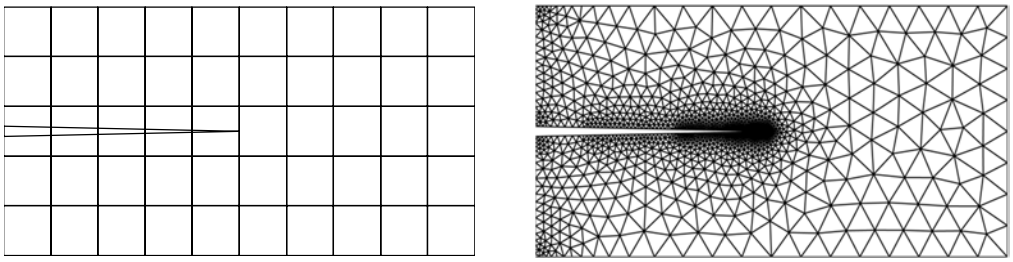


Fig. 4.1. Discretized fractured domain. (a) Fractures cutting across elements, (b) Elements aligned with fracture.

The extended finite element method (XFEM), introduced by Belytschko and Black (1999) and subsequently improved by Moës *et al.(*1999), exploits the partition of unity (PU) (Melenk and Babuska, 1996) property of finite elements. Local enrichment functions are added to the finite element shape functions which allow the approximated field and its derivatives to be discontinuous within an element. Therefore, a domain can be modelled by finite elements without explicitly meshing discontinuities. Belytschko *et al.* (2009) gives a comprehensive review of the extended finite element method.

The enrichment terms consist of modified Heaviside function and Branch functions which represent the Westergaard near tip asymptotic fields (Fleming

et al., 1997). The Heaviside function is applied to elements entirely cut by a fracture and is given by:

$$H(\mathbf{x}) = \begin{cases} +1 & \phi(\mathbf{x}) \geq 0, \\ -1 & \phi(\mathbf{x}) < 0, \end{cases} \tag{4.1}$$

where $\phi(\mathbf{x})$ is the signed distance function normal to the crack. Branch functions are used to enrich elements that contain fracture tips and are given by:

$$\left[\Phi_\gamma(\mathbf{x})\right]_{\gamma=1}^{4} = \left[\sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\theta\sin\frac{\theta}{2}, \sqrt{r}\sin\theta\cos\frac{\theta}{2}\right], \tag{4.2}$$

where r and θ are the polar coordinates of the point $\mathbf{x}$ (integration point or finite element node) in a local coordinate system centred on the tip of the fracture with the x-axis aligned with the fracture direction (Fig. 4.2).

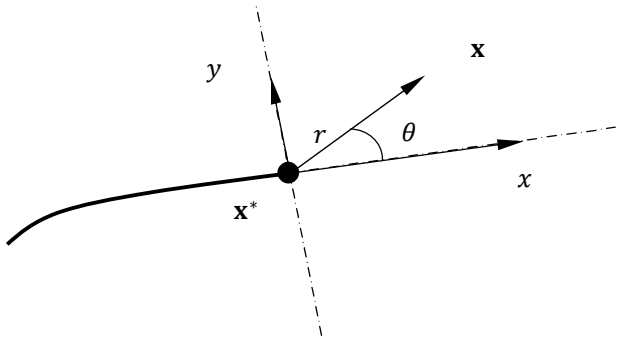


Fig. 4.2. Coordinate configuration for fracture front enrichment functions in 2D. $\mathbf{x}^*$ represents the fracture tip and $\mathbf{x}$ represents either an integration point or finite element node.

The addition of these enrichment terms results in the following displacement field within a fractured domain:

$$\mathbf{u}^h(\mathbf{x}) = \underbrace{\sum_{I\in\mathcal{N}} N_u^I(\mathbf{x})\bar{\mathbf{u}}_I}_{Standard} + \underbrace{\sum_{J\in\mathcal{N}_{\mathrm{cr}}} \widehat{N}_u^J(\mathbf{x})\left[H(\phi(\mathbf{x})) - H\left(\phi(\mathbf{x}_J)\right)\right]\mathbf{a}_J}_{Heaviside} + \underbrace{\sum_{K\in\mathcal{N}_{\mathrm{tip}}} \widehat{N}_u^K(\mathbf{x})\sum_{\gamma=1}^{4}(\mathbf{x})\left[\Phi_\gamma(\mathbf{x}) - \Phi_\gamma(\mathbf{x}_K)\right]\mathbf{b}_{\gamma K}}_{Branch}, \tag{4.3}$$

where $\bar{\mathbf{u}}_I$ is the nodal displacement vector associated with the continuous part of the finite element solution (standard approximation), N_u^I, $\widehat{N}_u^J$ and $\widehat{N}_u^K$ are finite element shape functions, $\mathcal{N}$ is the set of all nodes in the mesh; $\mathcal{N}_{\mathrm{cr}}$ is the set of nodes whose support element is cut by the fracture; $\mathcal{N}_{\mathrm{tip}}$ is the set of nodes whose support element contains a fracture tip; $\mathbf{a}_J$ is the nodal enriched degrees of freedom vector associated with the modified Heaviside function; and $\mathbf{b}_{\alpha K}$ is the nodal enriched degree of freedom vector associated with the elastic asymptotic fracture-tip functions. Both $\mathbf{a}_J$ and $\mathbf{b}_{\alpha K}$ represent additional degrees of freedom associated with enriched nodes.

The determination of both Heaviside and tip enriched nodes is illustrated in Fig. 4.3. Tip enriched nodes can be determined via topological or geometrical/fixed area enrichment. Only the nodes of elements containing the fracture tips are enriched if a topological enrichment is implemented. If geometrical enrichment is used nodes within a fixed area centred on the fracture tip become tip enriched nodes (see Fig. 4.3). Although topological enrichment is usually adequate, geometrical enrichment improves the method's rate of convergence (Bechet *et al.*, 2005; Laborde *et al.*, 2005; Belytschko *et al.*, 2009). Geometrical enrichment was used in this work with a fixed area of radius ($\boldsymbol{R}$) 1/20 of the domain width.

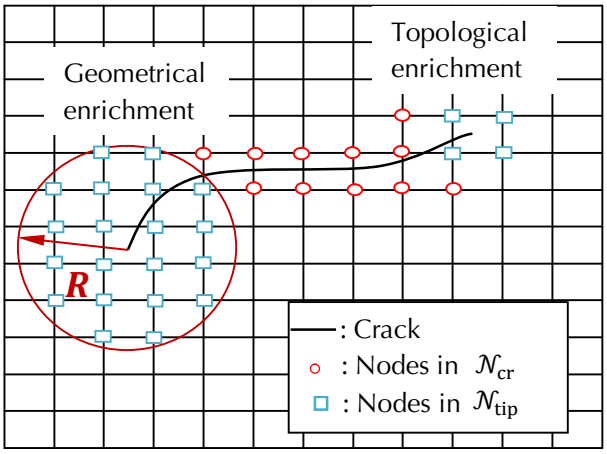


Fig. 4.3. Assignment of enriched nodes. ○ Heaviside enriched □ Tip enriched

4.1.1 Intersecting Fractures

The nodal enrichment of elements containing intersecting fractures was determined using the methodology presented by Daux *et al.* (2000). The method, which introduces a discontinuous branching function to (4.3), is described in this section.

Consider the case of a branched fracture which is modelled using the standard finite element method (Fig. 4.4(a)). This fracture can also be modelled in an equivalent discrete space using XFEM by enriching the standard finite element formulation with the discontinuous function $H(\mathbf{x})$ and another function which accounts for the junction between the branches of the fracture (Fig. 4.4(b)).

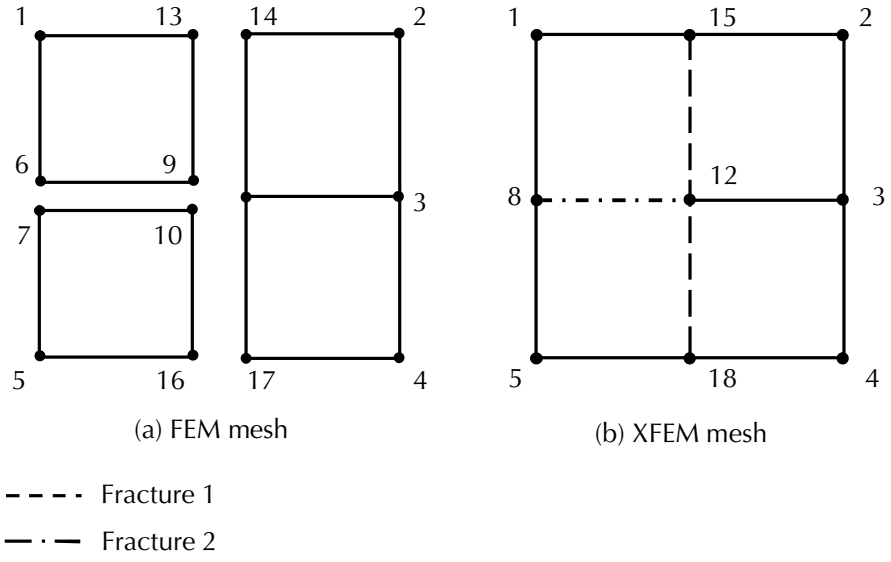


Fig. 4.4. Intersecting fractures modelled by: (a) finite element method and (b) the extended finite element method.

By defining the discontinuous functions of fractures 1 and 2 as $H_1(\mathbf{x})$ and $H_2(\mathbf{x})$ respectively, the branching function $Br(\mathbf{x})$ can now be expressed as:

$$Br(\mathbf{x}) = \begin{cases} H_2(\mathbf{x}) & \text{for } H_1(\mathbf{x}) < 0, \\ 0 & \text{for } H_1(\mathbf{x}) > 0, \end{cases} \tag{4.4}$$

A visual interpretation of the relationship between the discontinuous functions is given in Fig. 4.5.

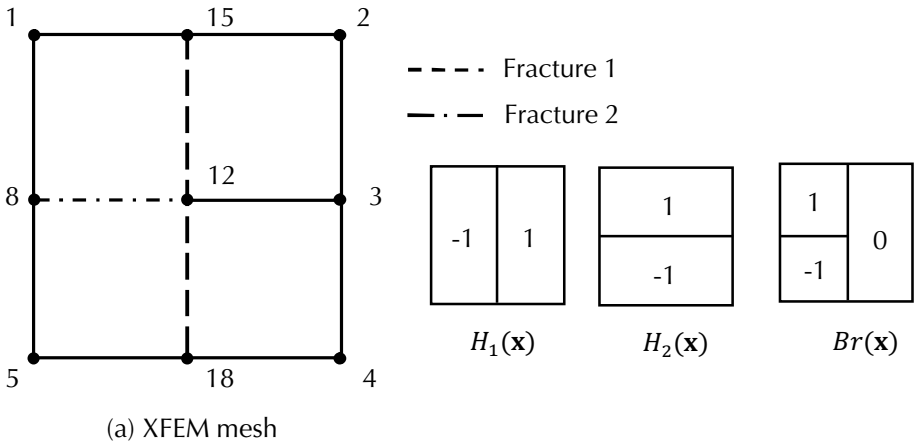


Fig. 4.5. Discontinuous enrichment functions, $H_1(\mathbf{x})$, $H_2(\mathbf{x})$ and $Br(\mathbf{x})$.

Daux *et al.* (2000) divides intersecting fractures into a main fracture and secondary branching fractures before applying the discontinuous functions to the formulation. This ensures that the XFEM discrete space remains equivalent to the standard finite element method discrete space, i.e., the degrees of freedom are the same for both numerical schemes. The enriched nodes for an intersecting fracture are defined as shown in Fig. 4.6. To simplify the figure, topological enrichment has been used for the fracture tips.

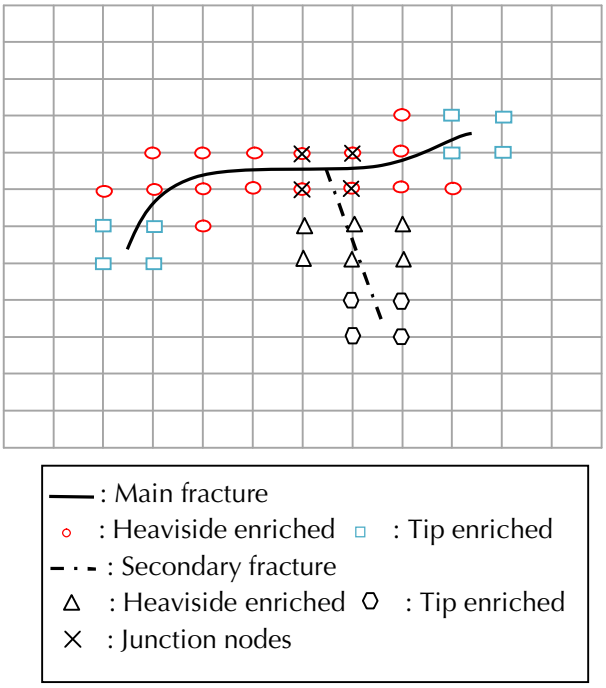


Fig. 4.6. Nodal enrichment for intersecting fracture.

For a main fracture with several connecting secondary fractures the displacement approximation can be obtained by adding the branching function to (4.3) and is given by:

$$\mathbf{u}^h(\mathbf{x}) = \sum_{I\in\mathcal{N}} N_u^I(\mathbf{x})\bar{\mathbf{u}}_I + \sum_{i=1}^{n_c}\sum_{J\in\mathcal{N}_{\mathrm{cr},i}} \hat{N}_u^J(\mathbf{x})\left[H_i(\phi(\mathbf{x})) - H_i\left(\phi(\mathbf{x}_{J,i})\right)\right]\mathbf{a}_{J,i} + \sum_{i=1}^{n_t}\sum_{K\in\mathcal{N}_{\mathrm{tip},i}} \hat{N}_u^K(\mathbf{x})\sum_{\gamma=1}^{4}(\mathbf{x})\left[\Phi_\gamma(\mathbf{x}) - \Phi_\gamma(\mathbf{x}_{K,i})\right]\mathbf{b}_{\gamma K,i} + \sum_{i=1}^{n_x}\sum_{M\in\mathcal{N}_{\mathrm{br},i}} \hat{N}_u^M(\mathbf{x})\left[Br_i(\phi(\mathbf{x})) - Br_i\left(\phi(\mathbf{x}_{M,i})\right)\right]\mathbf{c}_{M,i}\,, \tag{4.5}$$

where n_c is the number of fractures (main plus secondary fractures), n_t is the number of fracture tips, n_x is the number of intersections ($n_x = n_c - 1$) , $\mathcal{N}_{\mathrm{br}}$ is the set of nodes that are enriched with the branching function and $\mathbf{c}_M$ represents the associated enriched degrees of freedom.

4.2 Numerical Integration

Standard Gaussian integration causes numerical inaccuracies and creates ill-conditioned system matrices when performed on elements enriched by the discontinuous Heaviside function (4.1) and the Branch function (4.2) (Belytschko *et al.*, 2001; Sukumar and Prévost, 2003; Laborde *et al.*, 2005). To remedy the numerical issues associated with the integration of these discontinuous enrichment functions, elements intersected by a fracture are decomposed into sub-elements. The edges of these sub-elements are aligned along the discontinuity surface and a higher order of Gaussian quadrature is usually used in these elements to improve the accuracy of the numerical integration.

Element decomposition is done only for the purpose of numerical integration and does not introduce any additional degrees of freedom (nodes) Decomposition of finite elements that are intersected by the fracture is a clear-

cut procedure that is done using computational geometry since no restrictions are placed on the shape of the generated sub-elements. Fig. 4.7 illustrates element decomposition for quadrilateral and triangular elements. The sub-elements are determined by applying Delaunay triangulation to the vertex nodes of an element and the end nodes of the fracture segment contained within that element.

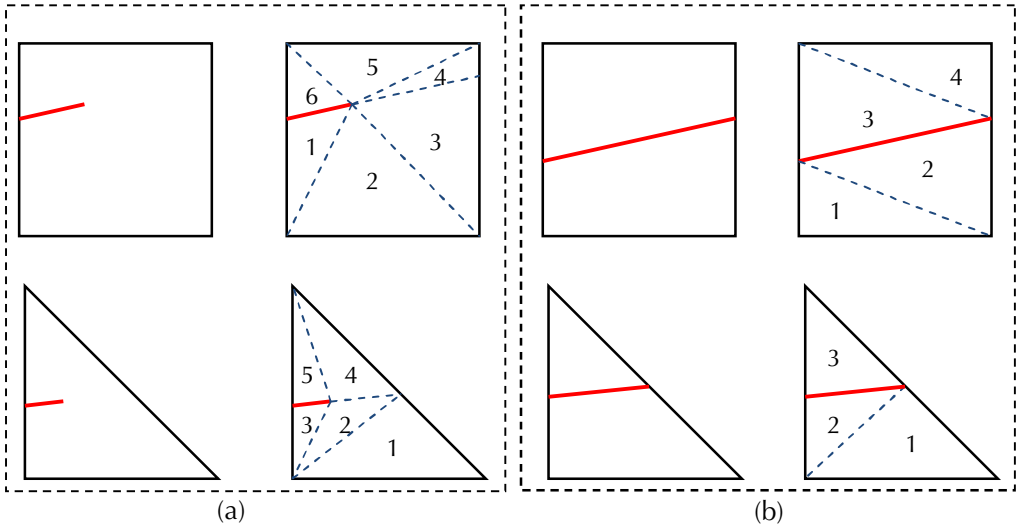


Fig. 4.7. Element decomposition into sub-elements: (a) elements containing fracture tip; (b) elements completely cut by fracture. Sub-elements are obtained by applying Delaunay triangulation to the vertex nodes of the element and end nodes of fracture segments contained within the element.

4.3 Fracture Closure

The influence of the matrix deformation around the fracture will undoubtedly have an influence on fractures within the domain. The faces of existing fractures at some instances may come into contact and close off the fracture. In this work fracture closures are accounted for using the well-established field of

contact mechanics. The use of a contact algorithm allows fractures within the domain to close and reopen during a simulation.

The bottom and top boundaries of a fracture may be defined by Γ_A and Γ_B respectively (Fig. 4.8). If Γ_A is defined as the master surface and Γ_B is defined as the slave surface, the penetration/contact condition between these two surfaces can be written as the motion of the top boundary relative to the bottom boundary. For a point P on Γ_B which has come into contact with the master surface Γ_A the penetration condition is defined as:

$$g_N = \begin{cases} (\mathbf{x}_A - \bar{\mathbf{x}}_P) \cdot \bar{\mathbf{n}}_A & \text{if } (x_A - x_P) \cdot n_A < 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

where $\mathbf{x}_A$ is the perpendicular distance from the surface Γ_A to the surface to point P. $\bar{\mathbf{n}}_A$ is the outward unit normal to Γ_A. In the case of geometrically linear kinematics, it is advantageous to write the penetration in terms of displacements.

$$g_N = (\mathbf{u}_A - \bar{\mathbf{u}}_P) \cdot \bar{\mathbf{n}}_A \tag{4.7}$$

where $\mathbf{u}_A$ and $\bar{\mathbf{u}}_P$ are the displacements at nodes A and P respectively.

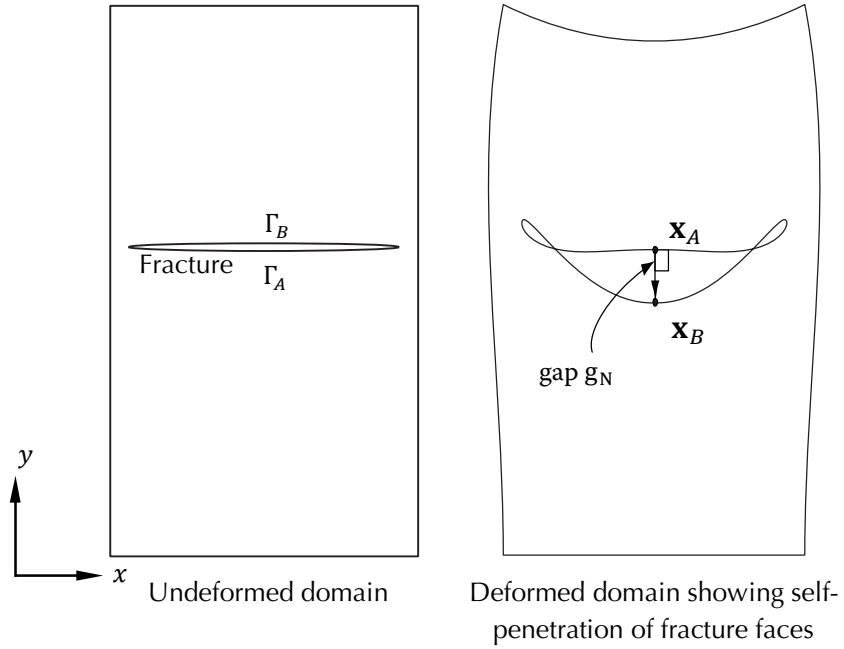


Fig. 4.8. Penetration condition of fracture faces.

For contact problems there are several different methods that can be used to ensure that the penetration constraint is satisfied, i.e., if interpenetration occurs between the fracture faces ($g_N < 0$) the displacement field must be iteratively re-calculated to ensure that (4.7) is reduced to $g_N = 0$. These methods include the Lagrange multiplier method (Hughes *et al.*, 1976; Papadopoulos and Solberg, 1998), the penalty method (Zhong, 1993; Laursen, 2002), the barrier method (Wriggers, 2006) and the Nitsche method (Nitsche, 1970; Becker and Hansbo, 1999). The penalty method, implemented here because of its simplicity, applies a penalty to the amount of penetration that can occur.

When interpenetration occurs between two fracture faces the gap, g_N, determined using (4.7) is less than 0. At this point contact forces are now

introduced into the system and the standard/extended finite element system equations which take the general form:

$$\mathbf{K}\bar{\mathbf{u}} = \mathbf{F}, \tag{4.8}$$

where $\mathbf{K}$ is the stiffness matrix, $\bar{\mathbf{u}}$ is the displacement vector and $\mathbf{F}$ is the external force vector, must be amended to include the contact force that exists between the two interacting faces. The system equations therefore become:

$$\mathbf{K}\bar{\mathbf{u}} + \mathbf{K}_c\bar{\mathbf{u}} = \mathbf{F}, \tag{4.9}$$

where $\mathbf{K}_c$ is the contact stiffness given by (Wriggers, 2006):

$$\mathbf{K}_c = \int_{\Gamma_C} \epsilon_N \left(N_I^A n_I^A - N_J^B n_J^B\right)^T \left(N_I^A n_I^A - N_J^B n_J^B\right) d\boldsymbol{\Gamma}_C, \tag{4.10}$$

where ϵ_N is a large number called the penalty parameter and can be thought of physically as a stiff spring between the contact surfaces. n^A and n^B represent the normal to the boundaries Γ_A and Γ_B respectively and N is the finite element shape function determined at the nodes along the boundaries Γ_A and Γ_B. The contact surface is defined as Γ_C and represents the portion of the fracture faces that touch, i.e., $\Gamma_C = \Gamma_A \cap \Gamma_B$. I and J represent the set of nodes boundaries Γ_A and Γ_B respectively.

When penetration occurs the system equation (4.9) must now be solved to determine a new displacement vector, $\bar{\mathbf{u}}$, such that the gap, g_N, determined using (4.7) returns to 0. Since the contact stiffness, $\mathbf{K}_c$, is non-zero only when $g_N < 0$, it is considered to be a function of the displacement $(\bar{\mathbf{u}})$ making equation (4.9) a non-linear system which is solved using the Newton-Raphson method (Belytschko and Fleming, 1999; Wriggers, 2006). The next section describes the implementation of the penalty method as part of this work.

4.3.1 Penalty Method Implementation

The implementation of the penalty method within this coupled deformation and fluid flow framework involves the following steps:

1. For each time step determine the displacement increment on either face of the fracture. As shown in Fig. 4.9, the fracture within an element is divided into segments and the incremental displacement is determined at fictional nodes along the segments on either side of the fracture using (4.3).

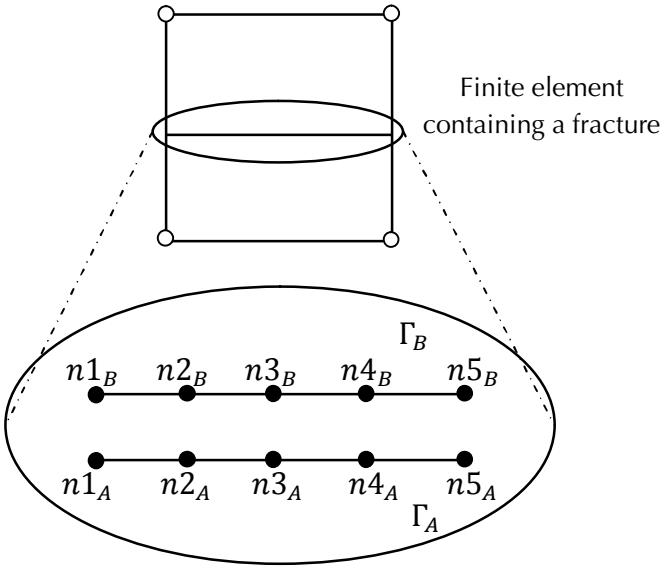


Fig. 4.9. Segments created to determine if contact has occurred between fracture faces.

2. Determine the total displacement obtained at the nodes shown in Fig. 4.9. The total displacement is equal to the sum of the incremental displacement at this time step and the cumulative displacement obtained over previous time steps.

3. The total displacements of the nodes on both faces of the fracture are used to determine the gap g_N using (4.7). If $g_N < 0$ interpenetration between the faces of the fractures has occurred and the incremental displacement must be minimized so that $g_N = 0$. If $g_N \geq 0$ no contact/penetration has occurred between the fracture faces and the simulation can proceed to the next time step.
4. If contact has occurred, the standard/extended finite element system equations take the general form

$$\mathbf{K}\Delta\bar{\mathbf{u}} + \mathbf{K}_c\Delta\bar{\mathbf{u}} = \Delta\mathbf{F}, \tag{4.11}$$

where $\mathbf{K}$ is the stiffness matrix, $\Delta\bar{\mathbf{u}}$ is the change in displacement vector and $\Delta\mathbf{F}$ is the change in the external force vector. Equation (4.11) is now solved to obtain the new incremental displacement vector using the Newton-Raphson algorithm shown in Box 4.1. This new incremental displacement vector ensures that $g_N = 0$, i.e., there is no interpenetration between the faces.

Box 4.1. The Newton-Raphson Iterative Procedure.

$\mathbf{K}_{int} = \mathbf{K} + \mathbf{K}_c$
$\Delta\mathbf{F}_{int} = \mathbf{K}_{int}\Delta\bar{\mathbf{u}}$
$\mathbf{R} = \Delta\mathbf{F}_{int} - \Delta\mathbf{F}_{ext}$ (applied force)
Do While $\left[\left\|\mathbf{R}^{i+1}\right\| < 1\text{x}10^{-6}$ or $i <$ maximum iteration$\right]$
  $\mathbf{K}^i\Delta\mathbf{x}^i = -\mathbf{R}^i$
  $\Delta\bar{\mathbf{u}}^{i+1} \leftarrow \Delta\bar{\mathbf{u}}^i + \Delta\mathbf{x}^i$
  $\Delta\mathbf{F}_{int}^{i+1} = \mathbf{K}_{int}\Delta\bar{\mathbf{u}}^{i+1}$
  $\mathbf{R}^{i+1} = \Delta\mathbf{F}_{int}^{i+1} - \Delta\mathbf{F}_{ext}$
  $i \leftarrow i + 1$
End Do

4.4 Spatial Discretization

Recall that the continuity equation 2.12 is written as:

$$\left(\frac{\alpha - n}{K_s} + \frac{n}{K_w}\right)\frac{\partial P_w}{\partial t} + \nabla \cdot \left(\frac{\mathbf{K}}{\mu_w}(-\nabla P_w + \rho \mathbf{g})\right) + \alpha \mathbf{I}^{\mathrm{T}} \frac{\partial \boldsymbol{\varepsilon}}{\partial t} = 0. \tag{4.12}$$

To simplify the system of equations the effect of gravity is neglected and the material grains are assumed incompressible, i.e., $1/K_s = 0, \alpha = 1$. Adding the transfer function term which accounts for the fluid flow interaction between the matrix and fracture, and a source sink term equation (4.12) takes the form:

$$\nabla \cdot \left(-\frac{\mathbf{K}}{\mu_w}\nabla P_w\right) + \frac{\bar{s}\mathbf{K}_f}{\mu_w}\left(P_{wm} - P_{wf}\right) + Q_w + \frac{n}{K_w}\frac{\partial P_w}{\partial t} + \alpha \mathbf{I}^{\mathrm{T}} \frac{\partial \boldsymbol{\varepsilon}}{\partial t} = 0. \tag{4.13}$$

After Ghafouri and Lewis (1996), the continuity equations can be written for both the porous matrix and the fractures as:

$$\nabla \cdot \left(-\frac{\mathbf{K}_m}{\mu_w}\nabla P_{wm}\right) + \frac{\bar{s}\mathbf{K}_f}{\mu_w}\left(P_{wm} - P_{wf}\right) + Q_w + \frac{n_m}{K_w}\frac{\partial P_{wm}}{\partial t} + \mathbf{I}^{\mathrm{T}}\frac{\partial \varepsilon}{\partial t} = 0, \tag{4.14}$$

for the porous matrix, and,

$$\nabla \cdot \left(-\frac{\mathbf{K}_f}{\mu_w}\nabla P_{wf}\right) - \frac{\bar{s}\mathbf{K}_f}{\mu_w}\left(P_{wm} - P_{wf}\right) + \frac{n_f}{K_w}\frac{\partial P_{wf}}{\partial t} = 0, \tag{4.15}$$

for the fractured zone.

The discretized form of the equations which describe coupled deformation and fluid flow in porous media is obtained using the Galerkin based finite element approach (see Appendix B for the standard procedure). The weighted residuals used in the integral equation (B6) must be amended to account for the presence of a discontinuity within the domain. The weighting functions therefore take the forms:

$$\mathbf{w}_u = \mathbf{w}_u^* + H\mathbf{w}_u^* + \Phi\mathbf{w}_u^*, \tag{4.16}$$

for the domain and

$$\overline{\mathbf{w}}_u = \overline{\mathbf{w}}_u^* + H\overline{\mathbf{w}}_u^* + \Phi\overline{\mathbf{w}}_u^*, \tag{4.17}$$

for the boundary, where H and Φ represent the Heaviside and Branch terms given by (4.1) and (4.2) respectively. The nodal displacements are approximated using (4.3) and the pressure variables are approximated using

$$P_{wm} \approx P_{wm}^h = \mathbf{N}_p \overline{\mathbf{P}}_{wm}, \tag{4.17}$$

and

$$P_{wf} \approx P_{wf}^h = \mathbf{N}_p \overline{\mathbf{P}}_{wf}, \tag{4.18}$$

where $\overline{\mathbf{P}}_{wm}$ and $\overline{\mathbf{P}}_{wf}$ are the vectors of nodal unknowns of matrix and fracture excess pressures respectively and $\mathbf{N}_p$ is the approximating shape function. The resulting discrete equations take the form:

$$\mathbf{AX} + \mathbf{B}\frac{\mathbf{dX}}{\mathbf{dt}} = \mathbf{C}, \tag{4.19}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{H}_m + \mathbf{H}_{T_r} & -\mathbf{H}_{T_r} \\ 0 & 0 & 0 & -\mathbf{H}_{T_r} & \mathbf{H}_f + \mathbf{H}_{T_r} \end{bmatrix} \tag{4.20}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{Ke}_{uu} & \mathbf{Ke}_{ua} & \mathbf{Ke}_{ub} & -\mathbf{L} & 0 \\ \mathbf{Ke}_{au} & \mathbf{Ke}_{aa} & \mathbf{Ke}_{ab} & -\mathbf{L}_a & 0 \\ \mathbf{Ke}_{bu} & \mathbf{K}_{ba} & \mathbf{Ke}_{bb} & -\mathbf{L}_b & 0 \\ \mathbf{L}^{\mathrm{T}} & \mathbf{L}_a^{\mathbf{T}} & \mathbf{L}_b^{\mathbf{T}} & \mathbf{S}_m & 0 \\ 0 & 0 & 0 & 0 & \mathbf{S}_f \end{bmatrix} \tag{4.21}$$

$$\mathbf{X} = \begin{bmatrix} \overline{\mathbf{u}} \\ \mathbf{a} \\ \mathbf{b} \\ \overline{\mathbf{P}}_{wm} \\ \overline{\mathbf{P}}_{wf} \end{bmatrix} \tag{4.22}$$

$$\mathbf{C} = \begin{bmatrix} \mathrm{d}\mathbf{F} \\ \mathrm{d}\mathbf{F}_a \\ \mathrm{d}\mathbf{F}_b \\ \bar{\mathbf{f}} \\ 0 \end{bmatrix} \tag{4.23}$$

The elements of the above listed matrices are given in Appendix D:

4.5 Temporal Discretization

Using the incremental loading scheme discussed in section 2.5.2, the final discrete equations can be written as:

$$\begin{bmatrix} \mathbf{Ke}_{uu} & \mathbf{Ke}_{ua} & \mathbf{Ke}_{ub} & -\mathbf{L} & 0 \\ \mathbf{Ke}_{au} & \mathbf{Ke}_{aa} & \mathbf{Ke}_{ab} & -\mathbf{L}_a & 0 \\ \mathbf{Ke}_{bu} & \mathbf{Ke}_{ba} & \mathbf{Ke}_{bb} & -\mathbf{L}_b & 0 \\ -\mathbf{L}^{\mathrm{T}} & -\mathbf{L}_a^{\mathbf{T}} & -\mathbf{L}_b^{\mathbf{T}} & \mathbf{S}_m - \Delta t\theta(\mathbf{H}_m + \mathbf{H}_{T_r}) & -\mathbf{H}_{T_r}\Delta t\theta \\ 0 & 0 & 0 & -\mathbf{H}_{T_r}\Delta t\theta & \mathbf{S}_f - (\mathbf{H}_f + \mathbf{H}_{T_r})\Delta t\theta \end{bmatrix} \begin{bmatrix} \Delta\bar{\mathbf{u}} \\ \Delta\mathbf{a} \\ \Delta\mathbf{b} \\ \Delta\overline{\mathbf{P}}_{wm} \\ \Delta\overline{\mathbf{P}}_{wf} \end{bmatrix} = \begin{bmatrix} \Delta\mathbf{F} \\ \Delta\mathbf{F}_a \\ \Delta\mathbf{F}_b \\ (\mathbf{H}_m + \mathbf{H}_{T_r})\overline{\mathbf{P}}_{wm_n} + -\mathbf{H}_{T_r}\overline{\mathbf{P}}_{wf_n} + \bar{\mathbf{f}} \\ -\mathbf{H}_{T_r}\overline{\mathbf{P}}_{wm_n} + (\mathbf{H}_f + \mathbf{H}_{T_r})\overline{\mathbf{P}}_{wf_n} \end{bmatrix} \tag{4.24}$$

The value of θ determines the accuracy and stability of the time integration scheme. A fully implicit scheme, i.e., $\theta = 1$, imposes no requirements on the time step size (Δt) chosen for both stability and the elimination oscillatory effects in the solution.

4.6 FM/XFEM Implementation

This section summarizes the implementation procedure for the FM/XFEM method. The method implementation has the following steps:

1. Mesh generation: The domain is divided into a series of quadrilateral finite elements;
2. Calculate the level sets of nodes in the vicinity of the line segments which describe the fracture geometry, see section 3.4. Nodal enrichment is also assigned at this step;

3. Determine fracture elements using the equality conditions presented in section 3.4.1;
4. Loop over all elements (matrix assembly):
    a. Determine element contributions to discrete system equation (4.24): if an element is a fracture element, calculate element permeability using the procedure outlined in section 3.3. Element contributions are also made to the fracture permeability and transfer matrix ($\mathbf{H}_f$ and $\mathbf{H}_{T_r}$) for fracture elements;
    b. Add element contributions to the global system matrix
5. Apply boundary conditions;
6. Solve the system equations given by (4.24) to obtain field variables, nodal displacements and pressures, at each time step. If the contact module is enabled, determine whether or not contact between the fracture faces has occurred for time step n. When contact occurs calculate the displacement and pressure fields that result in a stable configuration at time step n using Box 4.1. As discussed in section 4.3, a stable configuration is obtained when the displacement field calculated results in a penetration gap (g_N) equal to zero.

This implementation procedure assumes that the fracture geometry remains constant over the simulation period. Both the matrix permeability and fracture permeability are also assumed constant.

4.7 Numerical Examples

This section presents two numerical examples which validate the FM-XFEM algorithm developed as part of this work. The first numerical example is used to verify the implemented contact algorithm module. The second example compares the FM-XFEM approach with a discrete fracture model in which both the fracture and matrix are discretized using T6/T3 (Taylor-Hood) elements. A third example is used to illustrate the coupled deformation and fluid flow in a domain containing intersecting fractures.

4.7.1 Contact Algorithm Example

The model used to test the contact algorithm within the FM-XFEM framework is shown in Fig. 4.10. The model consists of a block of homogenous porous material and its material properties are given in Table 4.1. The domain contains a 1mm wide, 7m long horizontal fracture. The domain is fully saturated and allowed to freely drain at the top (excess pore water pressure is equal to zero) and is assumed to be sealed on the right, left and bottom boundaries. The mesh used for the simulation is shown in Fig. 4.11 along with the deformed meshes for simulations conducted with and without the contact algorithm being enforced. The deformed mesh obtained for the simulation which ignored contact between the fracture faces shows that interpenetration has occurred and is illustrated by the overlapping mesh. In reality once the fracture faces come into contact the fracture will close and the displacement will be uniform along the length of the fracture as illustrated by the deformed

mesh obtained when contact between the fracture faces is considered. This is also clearly shown in the corresponding displacement fields given in Fig. 4.12.

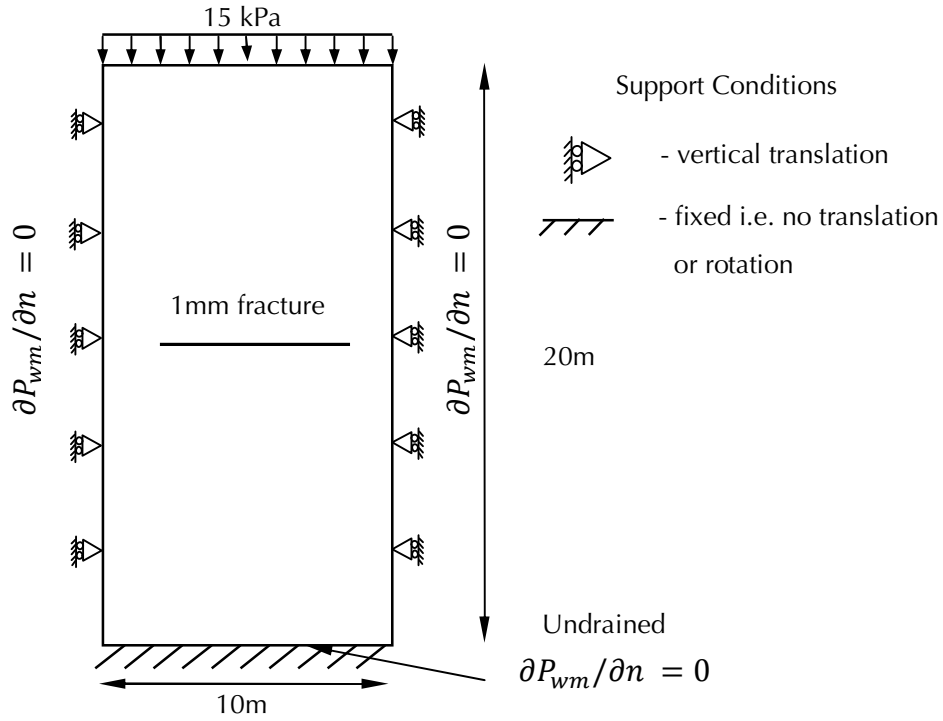
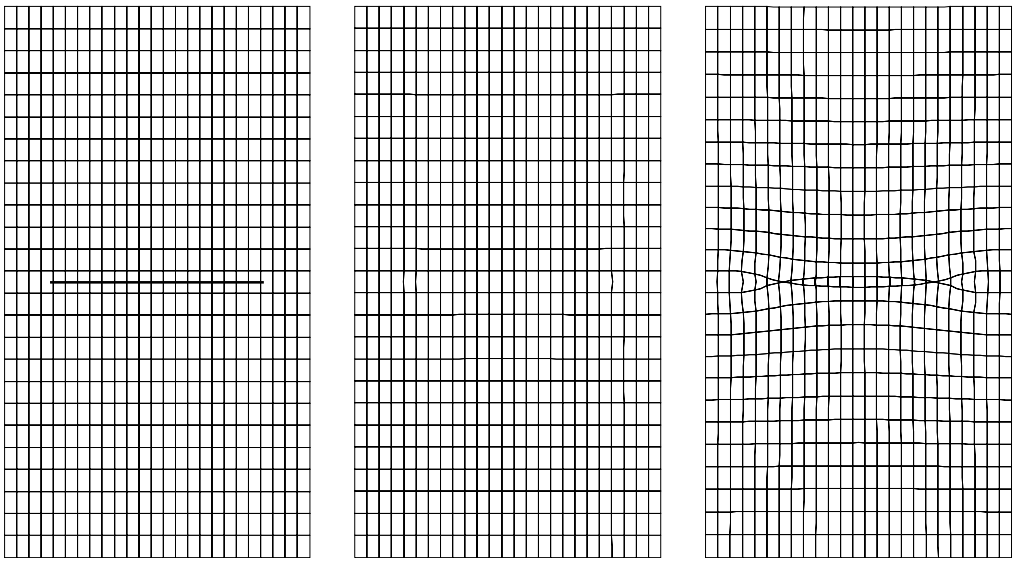


Fig. 4.10. 2D fractured domain with assigned boundary conditions.

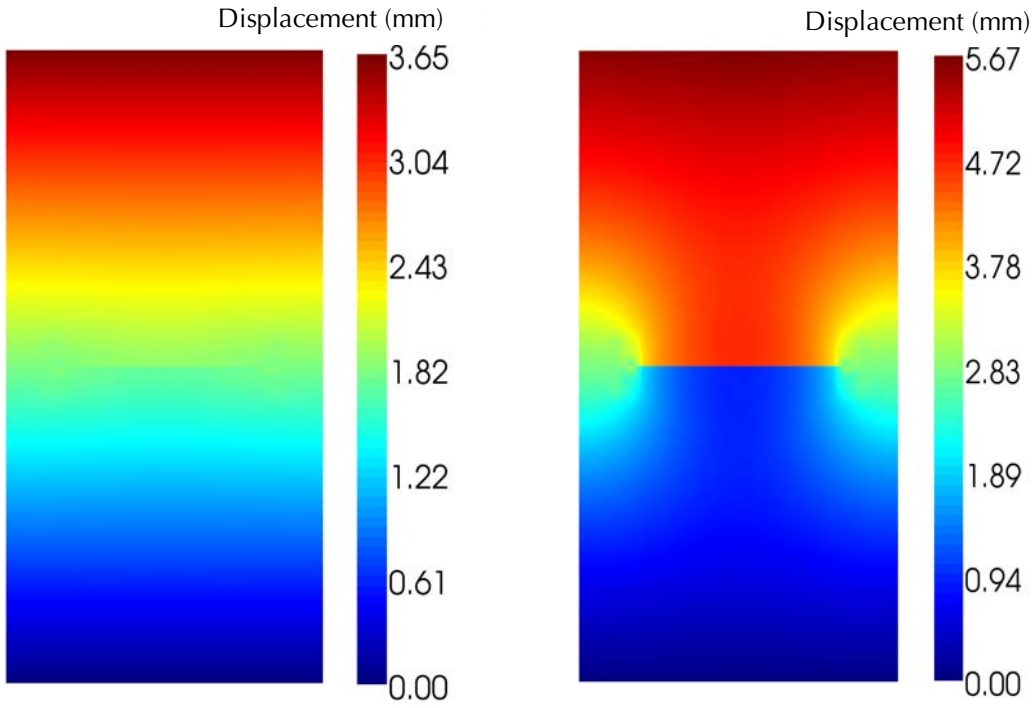
Table 4.1. Material properties for contact algorithm example.

Parameter	Definition	Magnitude	Units
E	Modulus of Elasticity	40	MPa
ν	Poisson's ratio	0.25	-
ϕ_1	Matrix porosity	0.1	-
ϕ_2	Fracture porosity	0.05	-
$\mathbf{K}_1/\mu$	Matrix mobility	1.75×10^{-3}	$m^2/(MPa{\cdot}s)$
$\mathbf{K}_2/\mu$	Fracture mobility	1.75	$m^2/(MPa{\cdot}s)$



FEM Mesh (a) Deformed mesh with contact (b) Deformed mesh without contact

Fig. 4.11. Initial FEM mesh along with deformed meshes: (a) contact between fracture faces is considered; (b) contact between fracture faces is ignored.



(a) Displacement field with contact (b) Displacement field without contact

Fig. 4.12. Displacement field: (a) contact between fracture faces is considered; (b) contact between fracture faces is ignored.

4.7.2 Coupled Numerical Example

The validity of FM-XFEM is demonstrated on the 2D plane strain problem shown in Fig. 4.13. The domain is fully saturated and allowed to freely drain at the top (excess pore water pressure is equal to zero) and is assumed to be sealed on the right, left and bottom boundaries. A static load of 10kN/m^2 is applied at the top of the block at time t=0 and maintained throughout the duration of the simulation. The block is homogenous and assumed to behave in linearly elastic manner and material properties are shown in Table 4.2. The fracture is 8m long, inclined at 45 degrees and is centred in the middle of the block.

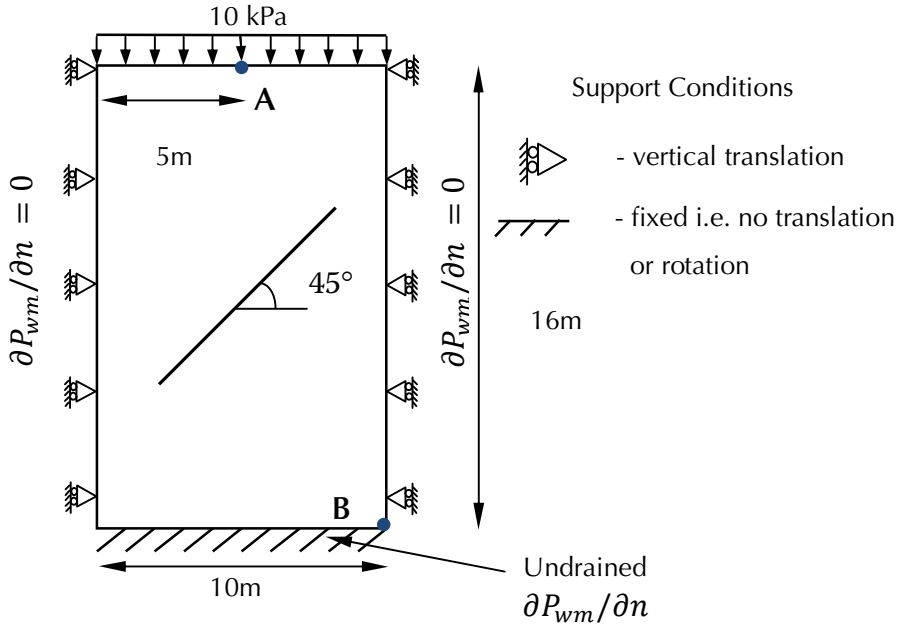


Fig. 4.13. 2D fractured domain with assigned boundary conditions.

Table 4.2. Material properties for coupled example.

Parameter	Definition	Magnitude	Units
E_m	Modulus of Elasticity	40	MPa
ν	Poisson's ratio	0.3	-
n_m	Matrix porosity	0.1	-
n_f	Fracture porosity	0.05	-
$\mathbf{K}_m/\mu$	Matrix mobility	1×10^{-6}	$m^2/(MPa{\cdot}s)$
$\mathbf{K}_f/\mu$	Fracture mobility	1×10^{-1}	$m^2/(MPa{\cdot}s)$

The current model is compared to a discrete fracture model (DFM) in which the fracture is modelled as an elliptical soft inclusion with its interior meshed along with the domain using T6/T3 triangular elements. The inclusion has an aspect ratio of ~1:40 with a Young's modulus three orders-of-magnitude lower than the surrounding matrix material and a Poisson's ratio of $\nu = 0$. Since the fracture geometry remains constant throughout the simulation for the FM-XFEM model, i.e., the fracture does not close or open during the simulation, the nodes of the elements used to model the soft inclusion are assigned a cross-

displacement of zero to ensure consistency between the DFM and FM-XFEM models.

The computational meshes used for the both the DFM and FM-XFEM are shown in Fig. 4.14 along with the level of refinement used around the tip of the fracture. A uniform mesh is used for the FM-XFEM to demonstrate the methods ability to produce fairly accurate results in the vicinity of the discontinuity without the need for mesh refinement.

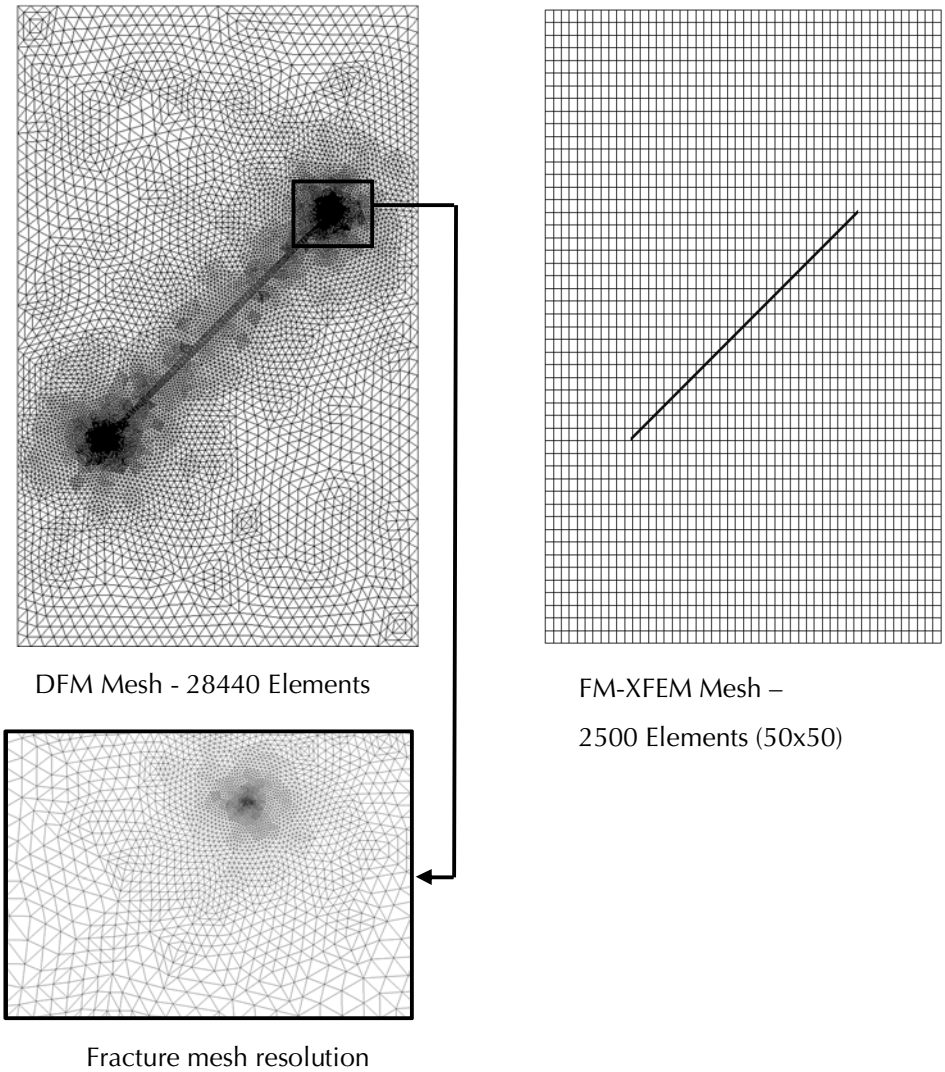


Fig. 4.14. Computational meshes used to conduct the comparison between DFM and FM-XFEM.

The resulting displacement fields after 100 days are shown in Fig. 4.15 and they are qualitatively similar. The vertical displacement at A (point located at the surface shown in Fig. 4.13) during 100 days is given in Fig. 4.16. The plot also illustrates the displacement variation for the DFM model and the current method at varying mesh resolutions. The FM-XFEM results adequately match the results of the DFM model and show very little change in the resulting displacement at higher mesh resolutions.

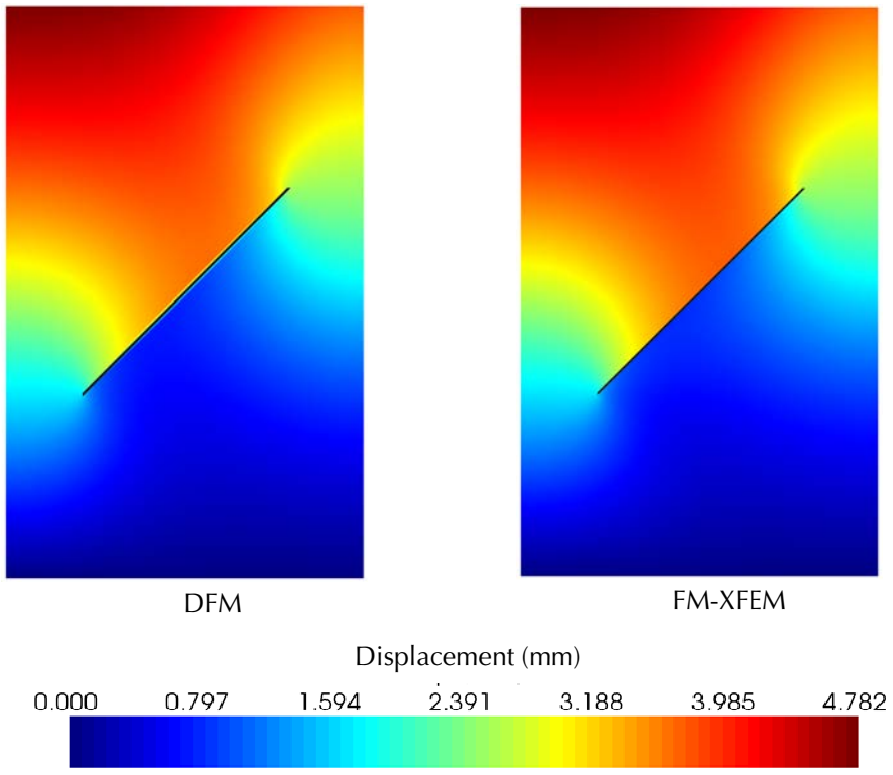


Fig. 4.15. Displacement field for fractured domain after 100days for DFM and FM-XFEM.

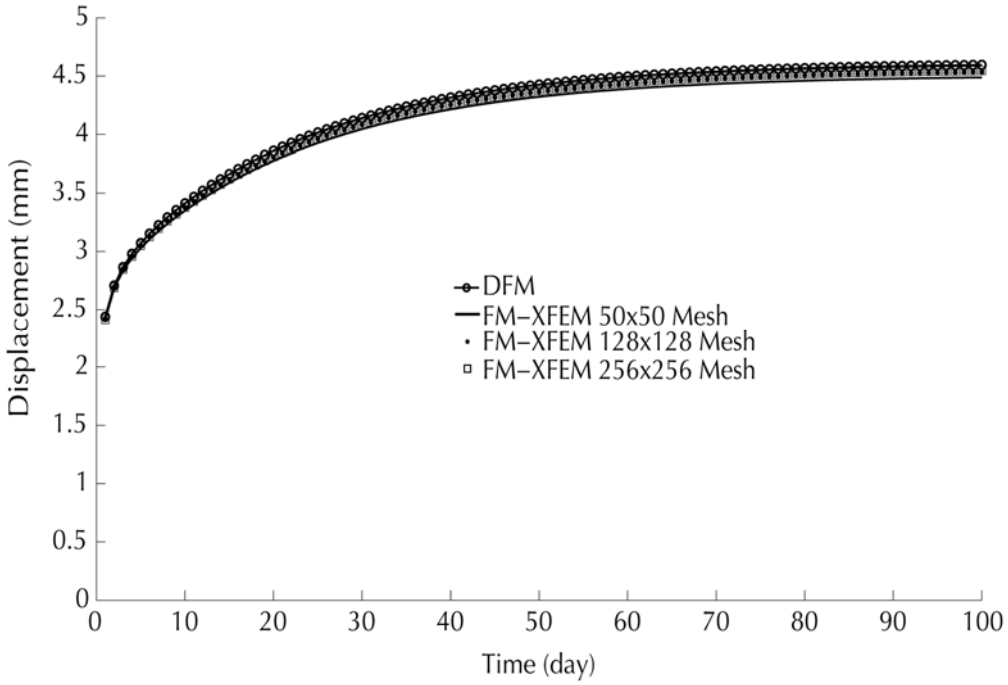


Fig. 4.16. Surface displacement at point A obtained for DFM and FM-XFEM.

The excess pore pressure fields are shown in Fig. 4.17 and they are also qualitatively similar. The variation in the excess pore pressure within the domain at point B (point located at bottom boundary shown in Fig. 4.13) has been plotted in Fig. 4.18. The plot shows the variation in pressure for the DFM and the current model at varying mesh resolutions. The pressure variation is very similar for all mesh resolutions used for the FM-XFEM and is also similar to the DFM results.

A point of interest observed on the pore pressure plot of both models is the increase in the initial pressure before it starts to decrease. The increase in pressure from the initial pressure and then the decrease is described as the Mandel-Cryer effect which has been discussed in section 2.6.2 of this thesis. The initial contraction of the domain that occurs as the water escapes the

domain causes an increase in the excess pore pressure above the initial value. The results obtained illustrate that on a coarse mesh, FM-XFEM is able to provide a physical description of the pressure dissipation comparable to that of the DFM.

Using a 512x512 reference mesh the convergence rate of FM-XFEM was determined with mesh resolutions of 32x32, 64x64, 128x128 and 256x256. The convergence rate, calculated using the error measure of the sum of the L_2-norm of the displacement gradients and the L_2-norm of the pressure given in section 2.6.2 of this thesis, is 1.05. A log-log plot of the L_2 error norm vs. the maximum nodal spacing h is shown in Fig. 4.19. The convergence rate obtained for FM-XFEM is similar to the rate of convergence calculated for the finite element implementation of fracture mapping approach obtained in chapter 3. It should be noted that this rate of convergence represents a 50% reduction in the rate obtained for the standard finite element consolidation formulation presented in chapter 2.

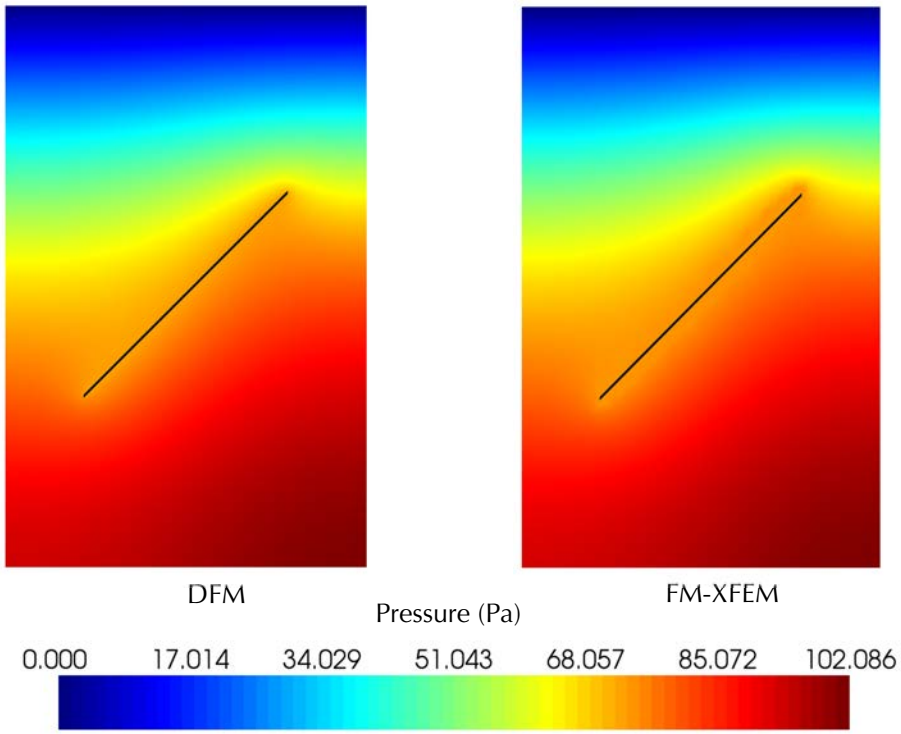


Fig. 4.17. Pressure field for fractured domain after 100days for DFM and FM-XFEM.

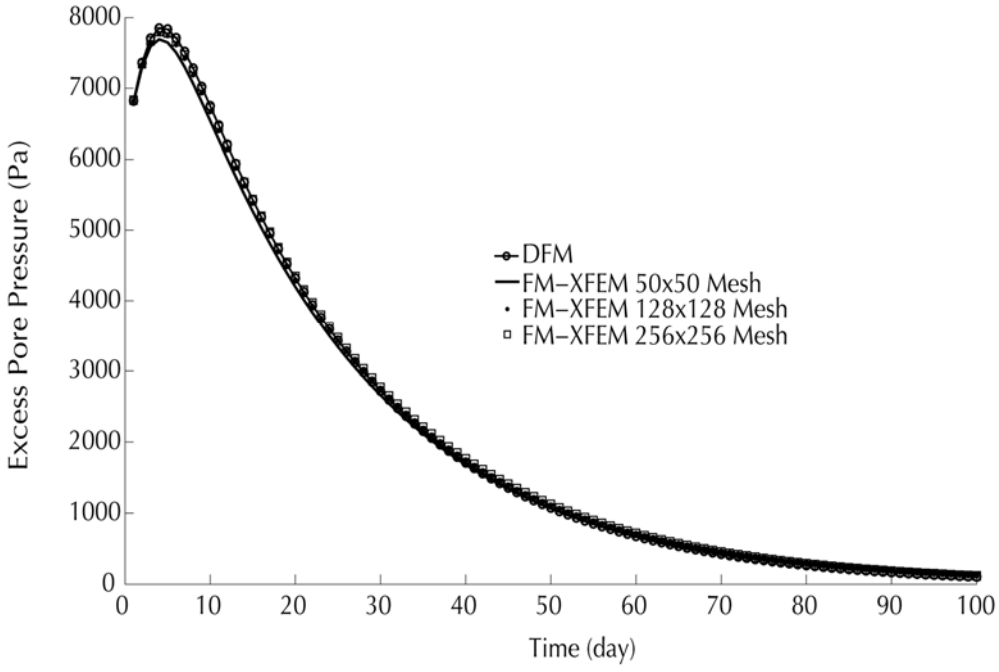


Fig. 4.18. Excess pore pressure at point B obtained for DFM and FM-XFEM.

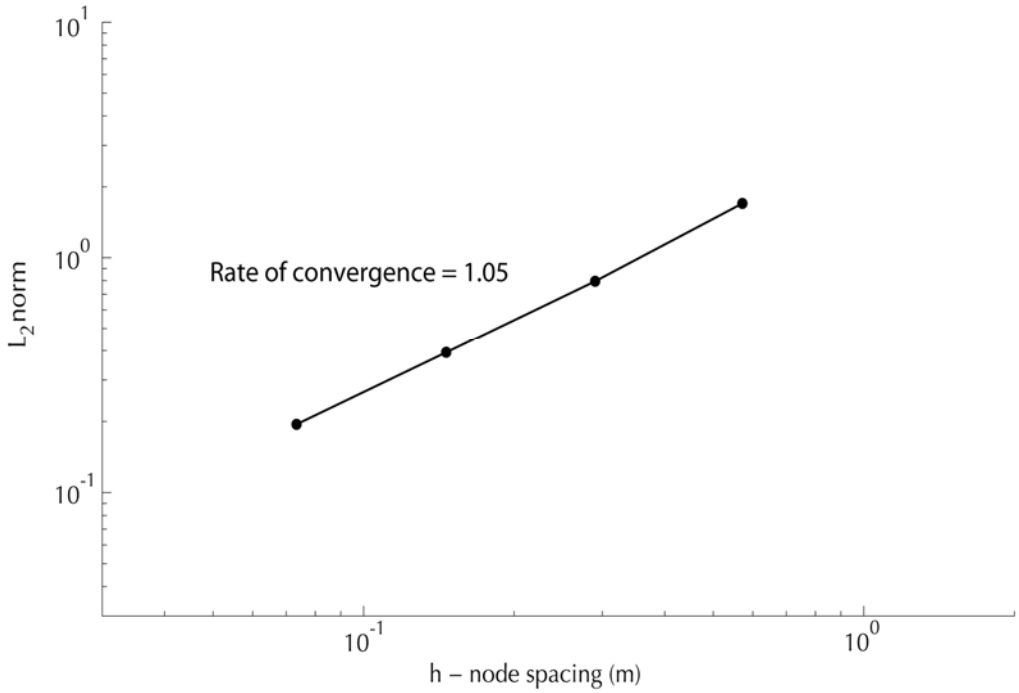


Fig. 4.19. Convergence rate of FM-XFEM.

4.7.3 Intersecting Fractures Example

This example is used to demonstrate the FM-XFEM's ability to resolve the displacement and pressure fields around intersecting fractures on a coarse, structured mesh. The model used for this example is the same as the previous example; however, the fracture is replaced by four intersecting fractures (Fig. 4.20). For this simulation, contact between the fracture faces has been ignored and the branching function presented by Daux *et al.* (2000) is used to enrich the nodes of elements containing fracture junctions.

Fig. 4.20 shows the computational mesh along with the displacement field and pressure contours. The figure illustrates discontinuities within the displacement field across the individual fractures and fracture intersections. Since contact has been neglected the load is not transferred across the fracture

faces and gives rise to the sharp changes observed in the displacement field. Displacement across the fractures is not well resolved if contact between the fracture faces is not considered.

The pressure contours are discontinuous and aligned to the fractures within the domain. The influence of the fractures on the pressure field is consistent with results obtained for the evaluation of the fracture mapping methodology for intersecting fractures presented in section 3.9.3.

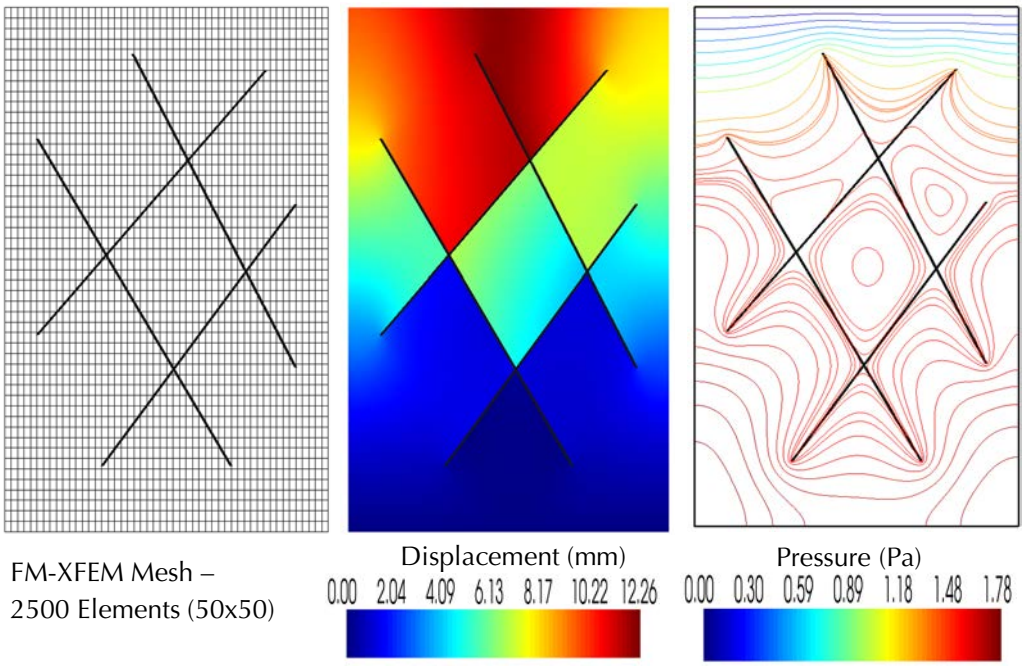


Fig. 4.20. Computational mesh along with resulting displacement field and pressure contours for domain containing intersecting fractures.

4.8 Conclusions

FM-XFEM is able to adequately determine the fluid flow and deformation in porous media without meshing fractures along with the computational domain. Although the method has been developed within a dual continuum framework, the flow through individual fractures is retained through permeability mapping. The validation exercise performed illustrates the method's potential as an alternative to the discrete fracture method. The method performs well on a coarse mesh and also adequately resolves both the displacement and pressure fields for a domain containing intersecting fractures.

CHAPTER 5

Fracture Propagation

The preceding chapter presents a method developed for simulating deformation and fluid flow in fractured porous media called FM-XFEM. This chapter extends the scheme to facilitate the propagation of existing fractures within the domain. The displacement field determined from FM-XFEM is used to determine the stress intensity factors at the fracture tip using the *J*-integral approach. The stress intensity factor provides a measure of the stress concentration at the fracture tip and is used in a mixed mode fracture criterion to determine whether or not existing fractures will propagate. The fracture extension length is determined using an empirical growth rate developed for fracturing in geological media. The mode I stress intensity factor determined from the *J*-integral is compared to an analytical solution and a propagation example is also presented.

5.1 Stress Intensity Factor Calculation

Stress intensity factors can be extracted from XFEM results using direct or energy approaches. Direct approaches calculate the SIFs from the XFEM displacements directly while energy approaches extract the SIFs from energy release rates computed using the displacements. Although energy approaches

are more accurate, the simplicity of the expressions used in direct approaches make them ideal for verifying energy approach results (Ingraffea and Wawrzynek, 2003). The displacement correlation technique (DCT) and the quarter-point displacement technique (QPDT) (Chan *et al.*, 1970; Fehl and Truman, 1999) are two direct approaches that are commonly used within the finite element framework.

The *J*-integral, an energy approach, has been employed to determine the SIFs in this work due to its prior successful application in conjunction with XFEM (Moës *et al.*, 1999; Dolbow *et al.*, 2000; Gravouil *et al.*, 2002; Nagashima *et al.*, 2003). The *J*-integral is a nonlinear fracture mechanics parameter introduced by Rice (1968), and is equivalent to the energy release rate, G, under linear elastic material assumptions.

The integral relates the energy release rate for a 2D body to a contour integral and is defined by:

$$J = \lim_{\Gamma \to 0} \int_{\Gamma} \left[W dx_2 - \sigma_{ij} \frac{\partial u_i}{\partial x_1} n_j \right] d\Gamma \,, \tag{5.1}$$

where W is the strain energy density, σ is the stress tensor and u is the displacement vector. **n** is the unit outward normal to the contour generated at the fracture tip where the x_1 axis is tangential to the crack and the x_2 axis is perpendicular to the crack (Fig. 5.1).

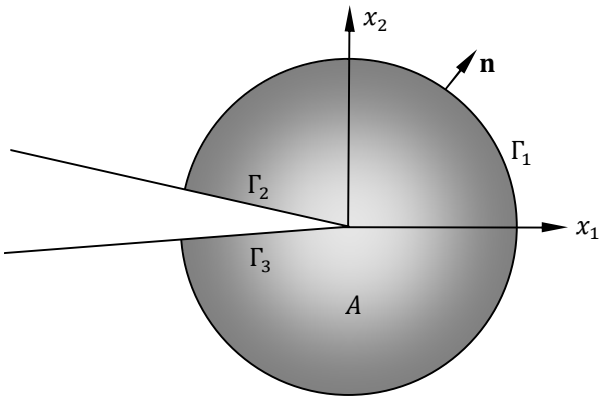


Fig. 5.1. Integration domain for evaluation of *J*-integral.

The contour integral in this form is path independent providing there are no body forces inside the integration area, no tractions on the crack surface and the material behaviour is elastic. To simplify the use of the *J*-integral in the finite element context, the contour integral is transformed into an area integral and is given by:

$$J = \int_{\mathrm{A}} \left[\sigma_{ij} \frac{\partial u_i}{\partial x_1} - W \delta_{1j} \right] \frac{\partial q_1}{\partial x_j} d\mathrm{A}, \tag{5.2}$$

where δ is Kronecker delta and q is a weighting function defined over the domain of integration which is defined by a circular region that surrounds the crack tip (Fig. 5.1). The integral is performed on all elements contained within the domain centred on the fracture tip. The weighting function q is defined by prescribing nodal values that are interpolated over elements in the integration domain using standard shape functions:

$$q = \sum_i N_i q_i \quad \text{and} \quad \frac{\partial q}{\partial x_j} = \sum_i \frac{\partial N_i}{\partial x_j} q_i \tag{5.3}$$

The function q takes a value of 1 inside the domain and decreases linearly to zero on the external edge of the internal domain.

The *J*-integral as defined by (5.2) gives the total energy release rate for the crack, and can be written as:

$$J = G = (K_I^2 + K_{II}^2)/E' \tag{5.4}$$

where K_I and K_{II} are the SIFs for modes I and II failure modes respectively. The fracture tip displacement is generally classified as mode I (opening mode), mode II (sliding or shearing mode and mode III (tearing mode) (see Fig. 5.2). E' is defined as:

$$E' = \begin{cases} E & \text{for plane strain,} \\ \dfrac{E}{1-\nu^2} & \text{for plane stress,} \end{cases} \tag{5.5}$$

where E and ν are the Young's modulus and Poisson's ratio of the material respectively.

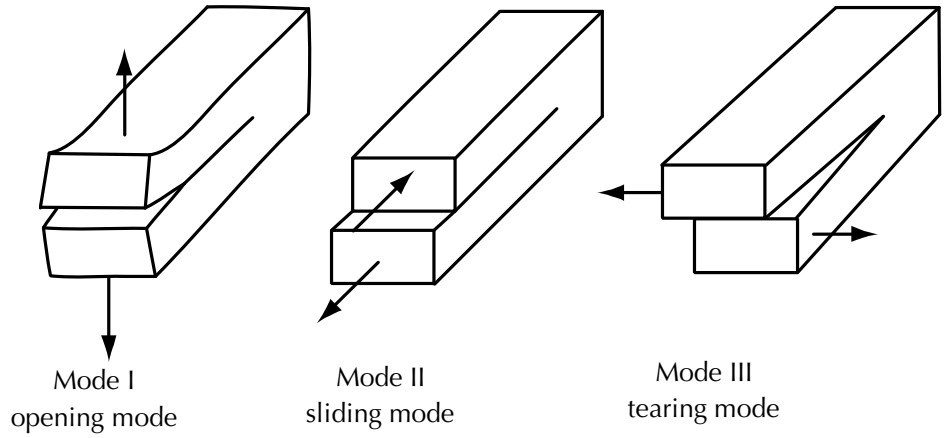


Fig. 5.2. Fundamental fracture tip displacement modes.

The *J*-integral in equation (5.4) on its own cannot be used to determine the values of K_I and K_{II} in a mixed mode crack problem. The SIFs are computed by writing the integral as the combination of two stress states and defining an interaction integral after Yau *et al.* (1980). The stress states considered are state

1, $\left(\sigma_{ij}^{(1)}, \varepsilon_{ij}^{(1)}, u_i^{(1)}\right)$, which represents the present state and state 2, $\left(\sigma_{ij}^{(2)}, \varepsilon_{ij}^{(2)}, u_i^{(2)}\right)$, is an auxiliary state. The *J*-integral for the sum of the two stress states is given by:

$$J^{(1,2)} = J^{(1)} + J^{(2)} + I^{(1,2)}, \tag{5.6}$$

where $I^{(1,2)}$ is the interaction integral for states 1 and 2 and is given by:

$$I^{(1,2)} = \int_{\mathrm{A}} \left[\sigma_{ij}^{(1)} \frac{\partial u_i^{(2)}}{\partial x_1} + \sigma_{ij}^{(2)} \frac{\partial u_i^{(1)}}{\partial x_1} - W^{(1,2)} \delta_{1j} \right] \frac{\partial q_1}{\partial x_j} d\mathrm{A}\,. \tag{5.7}$$

$W^{(1,2)}$ represents the interaction strain energy and is given by,

$$W^{(1,2)} = \sigma_{ij}^{(1)} \varepsilon_{ij}^{(2)} = \sigma_{ij}^{(2)} \varepsilon_{ij}^{(1)}. \tag{5.8}$$

Using the interaction integral the stress intensity factors are given by (Moës *et al.*, 1999):

$$K_I^{(1)} = \frac{E'}{2} I^{(1,\text{mode } I)} \text{ and } K_{II}^{(1)} = \frac{E'}{2} I^{(1,\text{mode } II)}. \tag{5.9}$$

5.2 Fracture Propagation

Fracture propagation/extension occurs once an applicable crack extension criterion has been satisfied. Fracture propagation referred to in this work represents quasi-static crack growth, in which inertial effects are neglected. This approach assumes that the body remains in equilibrium throughout the numerical simulation. Similar to brittle elastic solids, porous elastic media fracture propagation is assumed to occur when the stress intensity factors at the fracture tip satisfies the following condition:

$$K_I = K_{IC}, \tag{5.10}$$

where K_{IC} is the critical stress intensity factor in the crack opening mode. Equation (5.10) can be generalized to include the influence of mode II shearing effects. The mixed mode fracture criterion presented by Erdogan and Sih (1963) has been implemented in this work and takes the form:

$$\cos\left(\frac{\theta_p}{2}\right)\left[\frac{K_I}{K_{IC}}\cos^2\frac{\theta_p}{2}-\frac{3K_{II}}{2K_{IC}}\sin\theta_p\right]\leq 1, \tag{5.11}$$

where θ_p is the fracture propagation angle. The propagation angle is determined via the maximum hoop stress criterion which states that the fracture will propagate from its tip in the direction θ_p where the circumferential stress, $\sigma_{\theta\theta}$, is maximum. After Sukumar and Prévost (2003) the propagation angle is given by:

$$\theta_p = 2\arctan\left[-\frac{2K_{II}/K_I}{1+\sqrt{1+8(K_{II}/K_I)^2}}\right]. \tag{5.12}$$

5.2.1 Fracture Extension Length

Two common approaches have been used to determine the fracture extension length within the XFEM framework. The first approach assumes a constant fracture extension increment which is used to update the fracture geometry (Moës *et al.*, 1999). The second approach relies on external criteria to predict the increment of fracture extension. This work uses the extension criterion for geological material presented by Renshaw and Pollard (1994). This criterion relates the energy accumulated around a specific fracture tip with the maximum energy of all existing fracture tips and restricts fracture extension by applying an empirical velocity index, ϱ. The incremental fracture length is given by:

$$l_{adv} = l_{max}\left(\frac{G}{G_{max}}\right)^{\varrho}, \tag{5.13}$$

where G_{max} is the maximum energy release rate in the fracture set at the current growth iteration. After Paluszny and Matthai (2008), l_{max} is taken as the radius of the plastic zone that occurs at the fracture tip (Irwin, 1958). This radius is given by:

$$l_{max} = \begin{cases} \frac{1}{6\pi}\left(\frac{K_I}{\sigma_{ys}}\right)^2 & \text{for plane strain,} \\ \frac{1}{2\pi}\left(\frac{K_I}{\sigma_{ys}}\right)^2 & \text{for plane stress,} \end{cases} \tag{5.14}$$

where σ_{ys} is the material yield stress.

5.3 Fracture Propagation Algorithm

The fracture propagation algorithm flow chart is shown in Fig. 5.3. This algorithm assumes that the domain permeability remains constant during the simulation. Items shown in red indicate algorithm steps required for a stress dependent domain permeability, i.e., the matrix deformation causes changes in the matrix permeability. This permeability can be adjusted using the stress dependent relationship presented by Bai *et al.* (1997):

$$\Delta\mathbf{K} = \mathbf{K}_0\left(1 \pm \frac{1}{2}\left[\frac{9(1-\nu^2)}{2}(\pi\Delta\varepsilon)^2\right]^{\frac{1}{3}}\right)^2 \tag{5.15}$$

where $\mathbf{K}_0$ represents the initial permeability of the domain, ν is Poisson's ratio and ε is the volumetric strain of the domain. In equation (5.15), the positive sign is used when a tension load is applied to the domain and the negative sign is used when a compression load is applied to the domain.

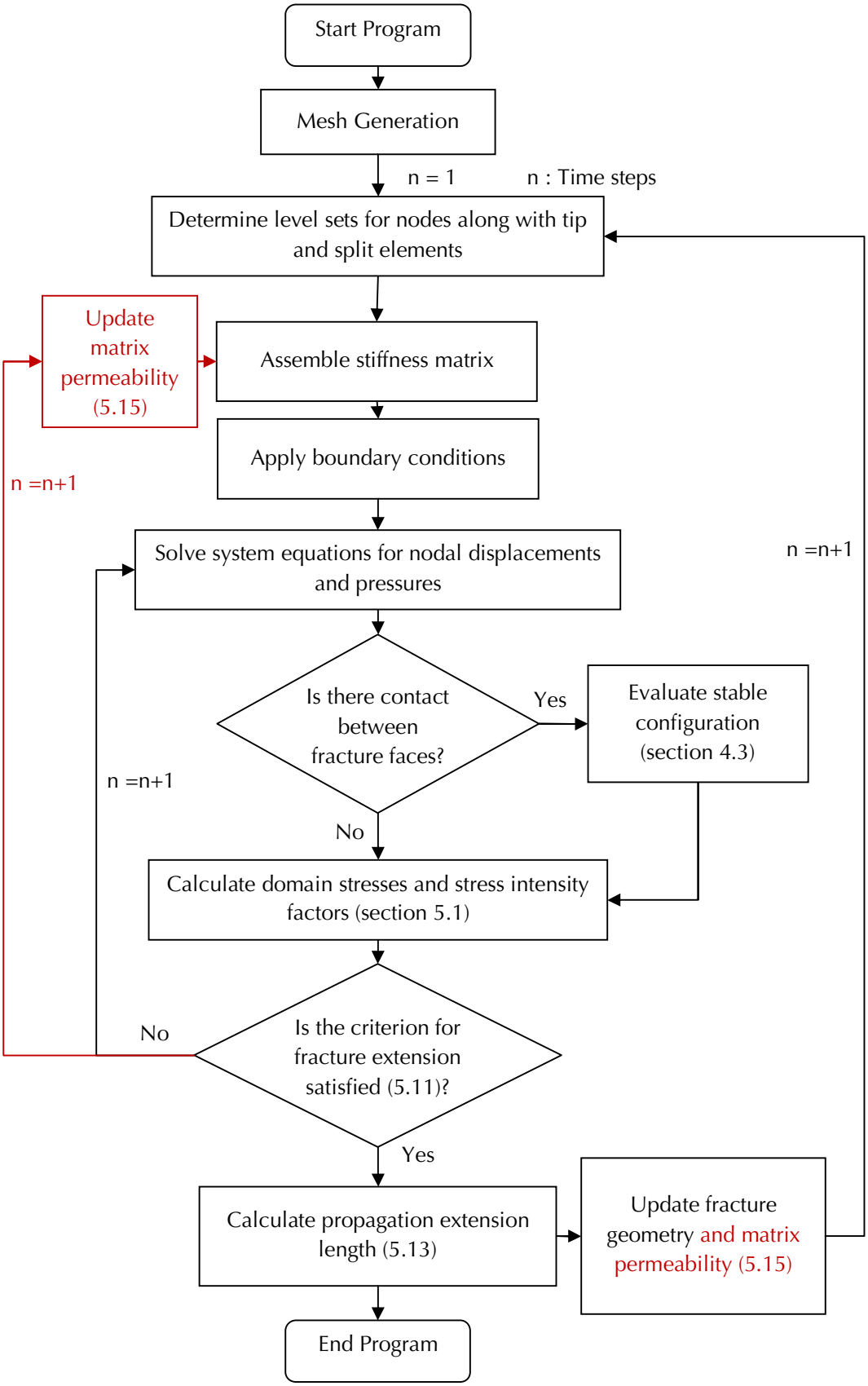


Fig. 5.3. Flow chart of fracture propagation algorithm.

5.4 Numerical Examples

The first example is divided into two parts. The first part is used to build confidence in the mode I stress intensity factor, K_I, determined using FM-XFEM and the J-integral method, by comparing the numerically determined SIF to a closed form solution found in literature. The next part of the example presents the extended configuration of the fracture obtained using the fracture propagation algorithm given in section 5.3. Plots of the vertical and Von Mises stresses within the fractured domain are also included to demonstrate the strain smoothing method presented in Appendix E.

The second example illustrates the effect an additional fracture within the domain has on the final fracture geometry after propagation. The final example shows the resulting fracture geometry obtained by allowing the fracture given in section 4.7.2 to propagate during the simulation.

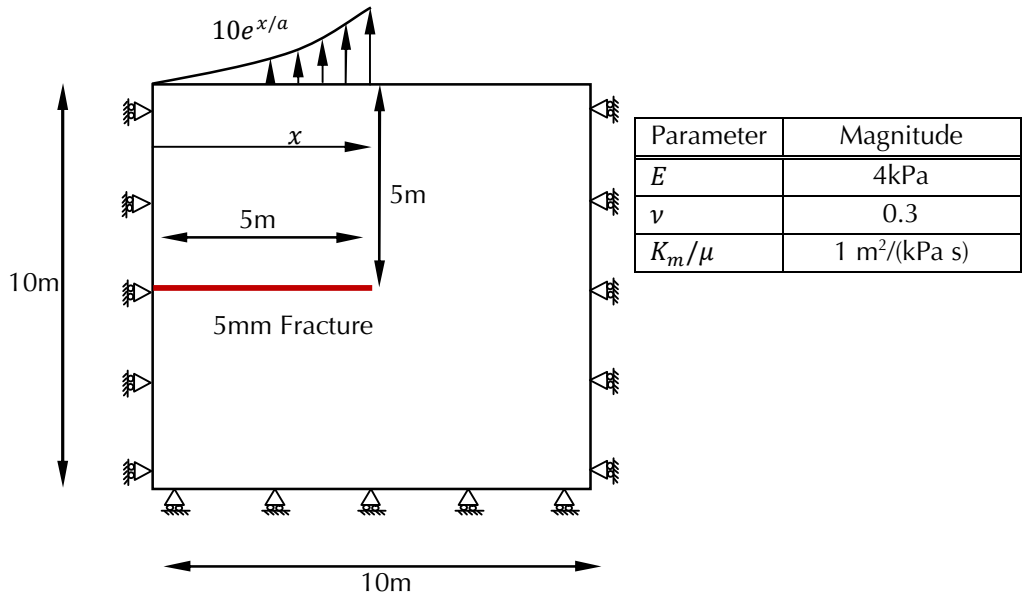
5.4.1 SIF Example

The mode I SIF, K_I obtained from FM-XFEM is compared to the analytical solution developed by Atkinson and Craster (1991) in the first part of this example. The domain is assumed fully saturated and the simulation is conducted until the system reaches equilibrium. The equilibrium state here is defined as the point where the excess pore pressure returns to zero. For the first part of the example the fracture is assumed static, i.e., no propagation occurs during the simulation. For a permeable semi-infinite fracture loaded

internally under tension by an impulse load of $\tau_0 e^{x/a}$, the elastic stress intensity factor is given by (Atkinson and Craster, 1991):

$$K_e = \tau_0 \sqrt{2a}\,, \tag{5.16}$$

where a is the fracture length (5m). The model used for this example is illustrated in Fig. 5.4:



Parameter	Magnitude
E	4kPa
ν	0.3
K_m/μ	1 m²/(kPa s)

Fig. 5.4. Model for stress intensity simulation.

The domain is allowed to freely slide vertically on the left and right boundaries and allowed to slide horizontally at the base. The domain is also allowed to freely drain on the left, right and bottom boundaries. Numerical simulations were conducted using varying radii for the geometrical enrichment at the fracture tip, 0.5m, 1m and 2m. (Geometrical enrichment for fracture tips is dicussed in section 4.1 of this document)

The mesh resolutions used for this numerical example are given below:

Mesh A: 1089 Elements

Mesh B: 4225 Elements

Mesh C: 16641 Elements

Mesh D: 66049 Elements

Mesh E: 263169 Elements

A comparison of the numerical SIF, K_I, with the analytical elastic SIF, K_e, is given in Table 5.1 for the different mesh refinements and fixed enrichment radii. The numerical values are in good agreement with the analytical SIF. A plot of the error against node spacing, h, is given in Fig. 5.5. The average rate of convergence for the current methodology is 2.08. The rate of convergence of the SIF obtained was 2.08. This convergence rate indicates that FM-XFEM is suitable for determining the stress intensity factors in fractured porous media.

Table 5.1. Comparison between analytical and numerical stress intensity factor

Mesh resolution	max K_I kPam$^{1/2}$			% Error = $\|K_I - K_e\|/\|K_e\|$ K_e =31.623 kPam$^{1/2}$		
	R=0.5	R=1.0	R=2.0	R=0.5	R=1.0	R=2.0
Mesh A	31.688	31.819	31.685	0.2083	0.6212	0.1960
Mesh B	31.635	31.675	31.639	0.0364	0.1653	0.0512
Mesh C	31.625	31.635	31.627	0.0056	0.0363	0.0119
Mesh D	31.624	31.626	31.624	0.0018	0.0089	0.0029
Mesh E	31.623	31.624	31.623	0.0006	0.0022	0.0007

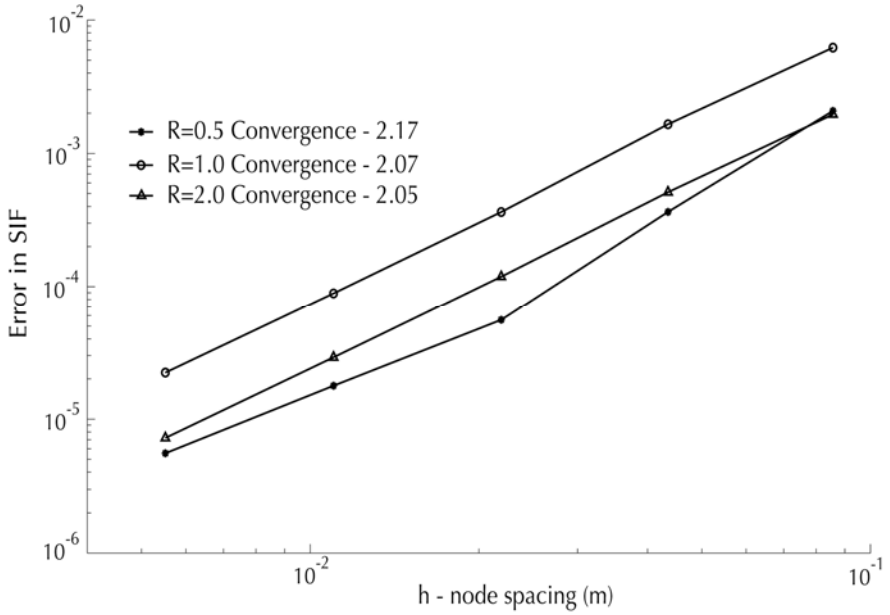


Fig. 5.5. Error in numerically determined stress intensity factor.

In the second part of this example the simulation is also conducted until the system reaches equilibrium, however, the fracture is allowed to propagate during the simulation. The values of the yield stress of the material, σ_{ys}, and the critical stress intensity factor, K_{IC}, are 25kPa and 1kPam$^{1/2}$ respectively. The final fracture configuration along with domain stresses are shown in Fig. 5.6. The lack of symmetry in the final fracture configuration is attributed to the asymmetrically applied loading. The stresses within the domain are determined using an extended global recovery procedure presented in Appendix E.

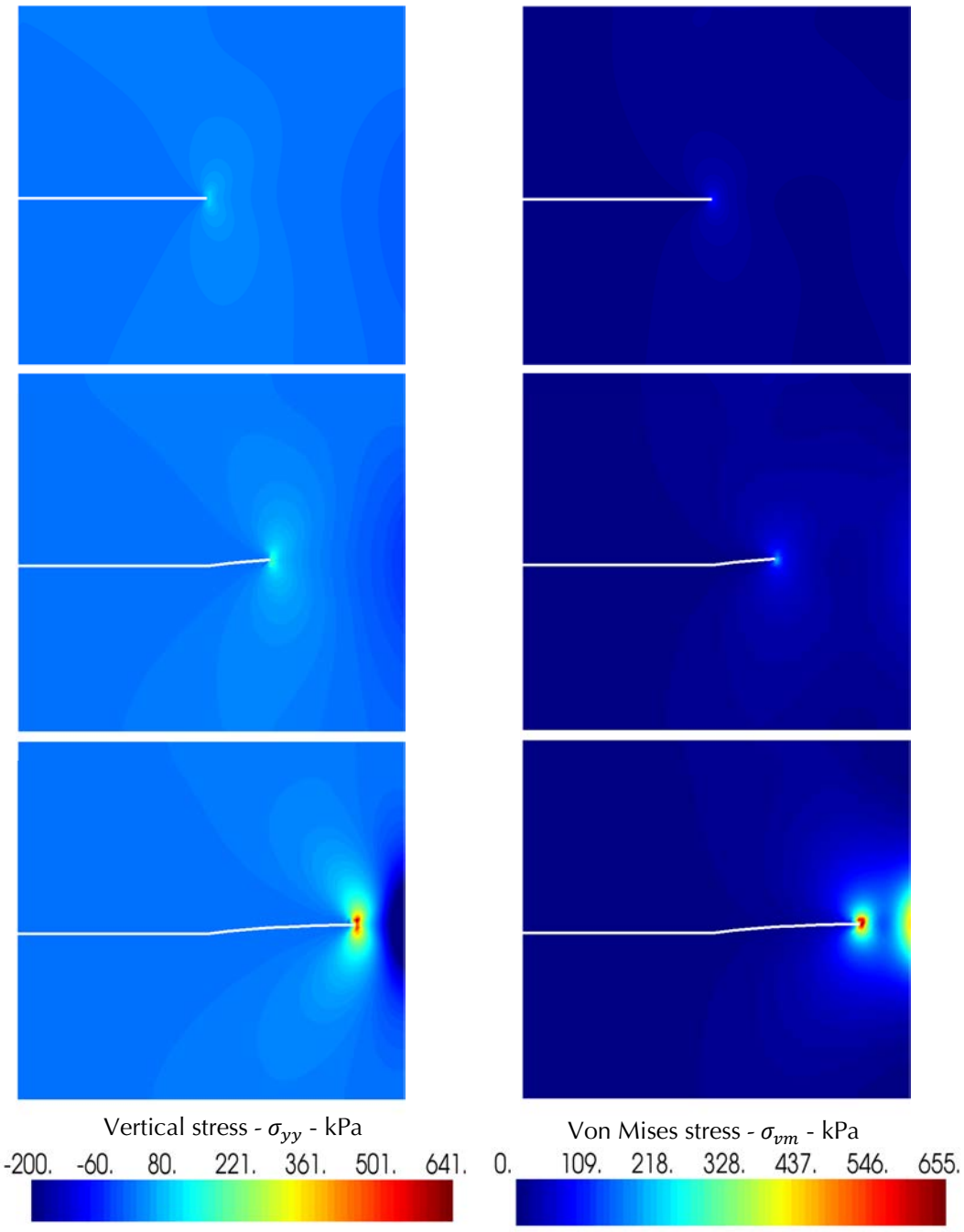


Fig. 5.6. Final fracture configuration and domain stress fields.

5.4.2 Fracture Interaction

This example is used to examine the influence an additional fracture has on the final fracture configuration obtained in the previous example. The model dimensions, loading and boundary conditions are the same as those specified in the previous example. The initial and final fracture configurations are shown in Fig. 5.7 along with the mesh used to conduct the simulation. The figure shows the fracture on the left moving towards the lower fracture on the right to form a curved pattern. This contrasts the left to right movement of the fracture observed in the previous example which contains a single fracture within the domain. The close proximity of the fracture tips during propagation influences the orientation of the fractures as they propagate. Interaction of the stress fields at the tips caused the propagation path to deviate from the straight course shown in Fig. 5.7.

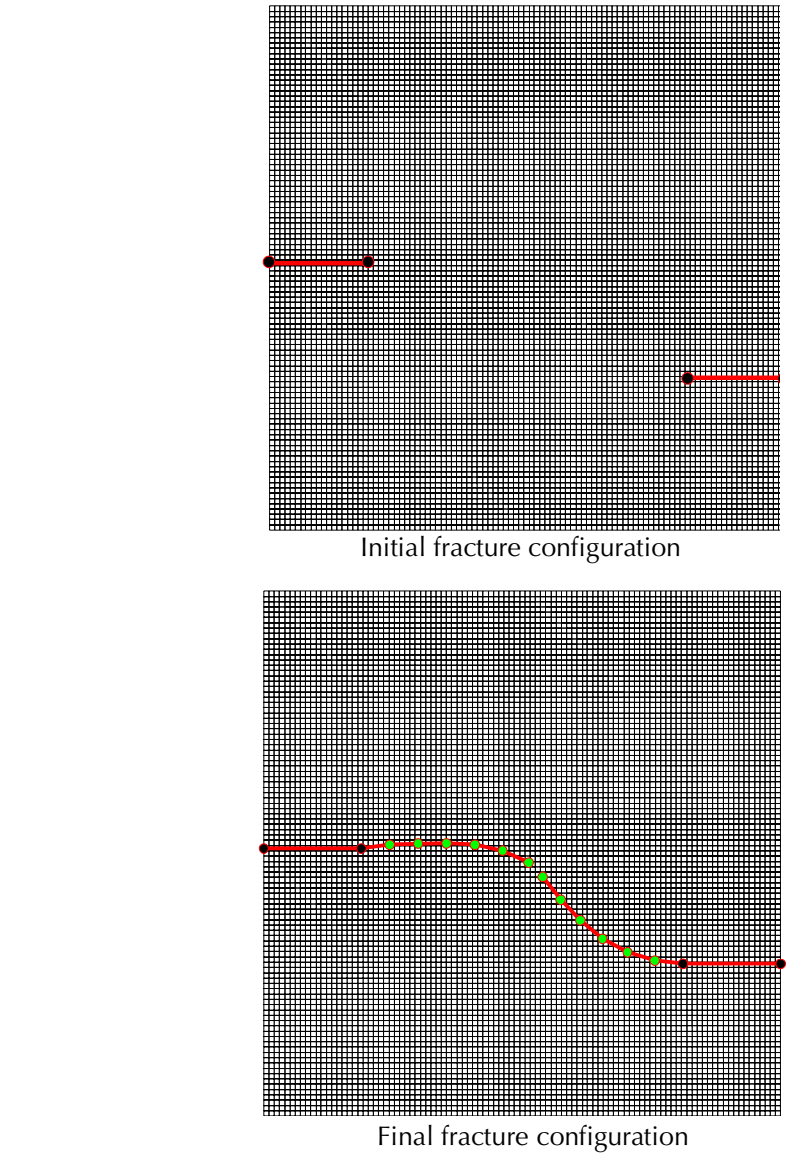


Fig. 5.7. Initial and final fracture configurations along with mesh.

5.4.3 Fracture Extension Example

This example illustrates the fracture configuration obtained by allowing the fracture in example 4.7.2 of chapter 4 to propagate during the simulation. The final fracture geometry along with domain stresses are shown in Fig. 5.8. The values of the yield stress of the material, σ_{ys}, and the critical stress intensity factor, K_{IC}, are 25kPa and 1kPam$^{1/2}$ respectively. The fracture extends horizontally at both fracture tips during the simulation. This indicates that the fracture propagation was controlled by the fluid pressure and not the applied vertical load. If the fracture propagation was dictated by the applied loading the extension would have been oriented in the vertical direction.

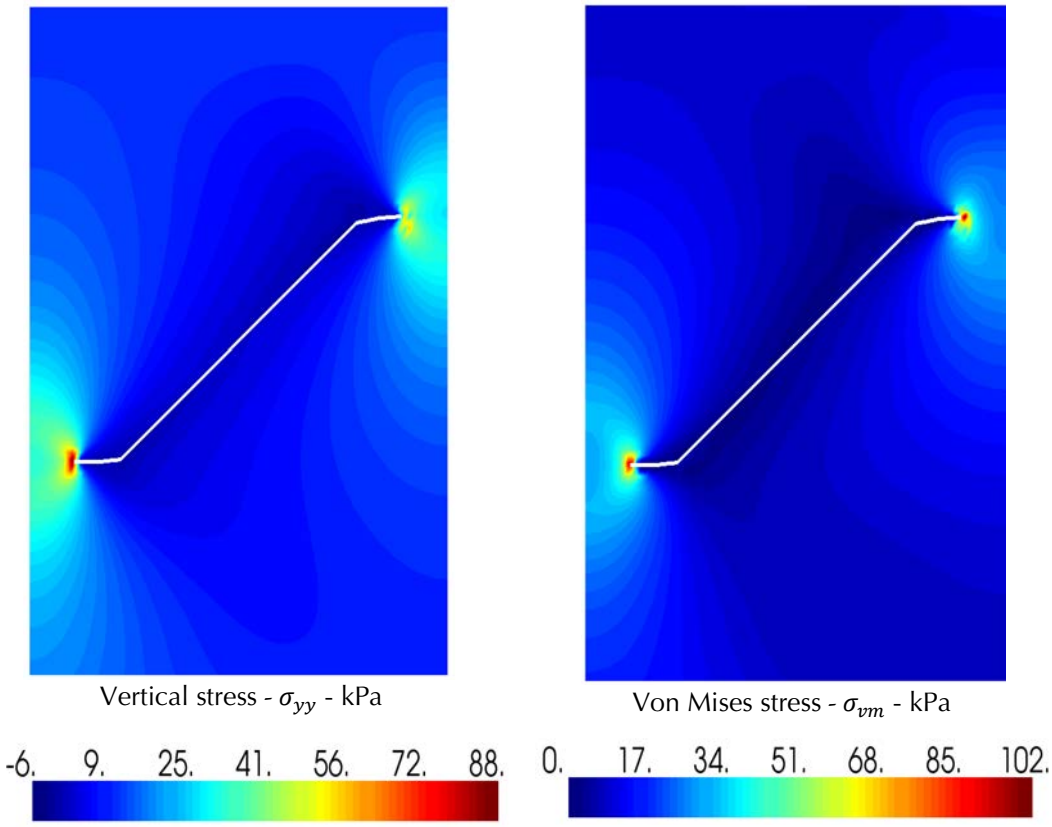


Fig. 5.8. Final fracture configuration and domain stress fields.

5.5 Conclusion

The mode I stress intensity factor calculated using the *J*-integral approach show good agreement with the analytical value provided in literature. Since the SIF is calculated using the displacements obtained from the FM-XFEM method it demonstrates the method's ability to accurately determine the discontinuous displacement field within a fractured porous domain.

CHAPTER 6

Conclusions and Future Work

This chapter, firstly, summarizes the work presented in this thesis and then presents recommendations for future research work.

6.1 Conclusions

This thesis presents a finite element based method for coupled deformation, fluid flow and fracture propagation in porous media. The numerical framework has been developed with the intention of creating a method that can be extended to test hypothesized mechanisms for polygonal fault generation. The method is a combination of a fracture mapping approach, which allows fluid flow interaction between the porous matrix and existing fractures and the extended finite element method, which is able to adequately model the discontinuous displacements across existing fractures without explicitly meshing the fracture geometry.

The fracture mapping approach is presented in chapter 3 along with the FEM and NI-RPIM (meshfree) implementation procedure. The FEM implementation separates finite elements which contain a fracture into two overlapping elements; a matrix element and a fracture element. The fracture properties are mapped onto the fracture element and the flow interaction

between the fracture and matrix elements is done using a transfer function. The transfer function adopted assumes that the flow within an element is directly proportional to the difference between the matrix pore pressure and the fracture fluid pressure. The meshfree implementation presented uses nodes to discretize the fracture geometry. Both the FEM and meshfree implementations perform adequately on coarse meshes when compared to analytical and discrete fracture model solutions. The mapping approach also adequately resolves the fluid flow interaction between intersecting fractures and the matrix.

The standard single-phase consolidation formulation was modified using the fracture mapping approach and the extended finite element method (FM-XFEM) in chapter 4. This formulation allows the inclusion of discontinuities within the domain without having the mesh conform to the discontinuity. In the standard finite element method the mesh must be aligned with existing discontinuities in order to obtain accurate results. This formulation reduces the need for refinement in areas where the solution is expected to possess singularities or large gradients as the domain undergoes compaction. The FM approach used within the formulation ensures an adequate representation of the fluid flow with the fractured domain. Also, the proposed method compared well with a discrete fracture model.

The numerical experiments presented in chapters 3 and 4 were used to develop confidence in the performance of FM and FM-XFEM respectively. Both methods have demonstrated they are able to adequately simulate both the fluid flow and deformation in fractured porous media. The displacement field

generated from FM-XFEM is used to determine the stress intensity factors at the fracture tip using the *J*-integral approach in chapter 5. The stress intensity factor provides a measure of the stress concentration at the fracture tip and is used in a mixed mode fracture criterion to determine whether or not existing fractures will propagate. The stress intensity factor obtained from the numerical method shows good agreement with the analytical solution.

6.2 Future work

The thesis acts as a proving ground for the proposed FM-XFEM. Extensions to the method are required before the method can be applied to real world applications; more specifically the evolution of polygonal faults. This section provides recommendations for possible future research.

6.2.1 Extension to Multiphase Flow

The current work uses the FM-XFEM to modify the single phase consolidation formulation. In order to extend this work to simulate polygonal faulting, coupled flow and deformation in hydrocarbon reservoirs or CO_2 storage simulations the method must be extended to handle multiphase flow components; water, oil and gas. This extension can be achieved by modifying the multiphase consolidation formulation presented by Lewis and Shrefler (1998) using FM-XFEM. Their formulation incorporates non-linear stress strain behaviour for the consolidation phenomenon and is capable of simulating multi-phase flow having relative permeability contrasts, complex

rock and fluid properties, the effects of capillary pressure and saturations. The model is enhanced further by its capability in simulating reservoirs with differing types of boundary conditions.

6.2.2 Extension to 3D

The proposed methodology has been limited to 2D space. If the evolution of polygonal faults is to be pursued the formulation must be extended to 3D space. XFEM has previously been implemented for 3D fracture propagation in linear elastic fracture mechanics (Sukumar *et al.*, 2000; Huang *et al.*, 2003; Sukumar *et al.*, 2003); this previous work provides a suitable template for extending the XFEM portion of the current formulation to 3D.

The validity of the fracture mapping approach in 3D must be assessed before FM-XFEM can be extended. Within 3D space a fracture exhibits local permabilities (k_x, k_y and k_z) which are aligned to the local axes of the fracture. The rotation and subsequent scaling of these permabilities within a 3D finite element may or may not adequately capture the flow interaction between the matrix and fracture and warrants further investigation.

6.2.3 Meshfree Implementation

The meshfree method has been used only for the fracture mapping approach in this study. A proposed research avenue is the investigation of a partition of unity enriched meshfree scheme similar to XFEM. The element free Galerkin method has been previously enriched using the partition of unit (Ventura *et al.*, 2002; Rabczuk and Belytschko, 2004; Nguyen *et al.*, 2008), however due to the difficulties in implementing essential boundary conditions and the complexity of the algorithms used to construct shape functions (Belytschko *et al.*, 1996; Liu, 2002; Liu and Gu, 2005) this scheme was not pursued in the present work. Although RPIM implemented in the fracture mapping does not have the above shortcomings, attempts made to incorporate the partition of unity into the formulation as part of this study found that the nodal displacements obtained from the scheme were in good agreement with analytical values; however the discontinuous strain field was not accurately calculated. This therefore warrants further investigation before the scheme can be fully developed and used in conjunction with the fracture mapping approach.

Appendix A

This appendix presents MATLAB® Executable (MEX) files developed for hsl_m57a, a direct matrix solver (Duff, 2004) and Voro++ (Rycroft *et al.*, 2006), a C++ library for the computation of 2D and 3D Voronoi cells.

A1 Ma57Mex.f90

```
!  This subroutine solves a linear system of equations
!  using hsl_ma57 direct sparse solver.
!  http://www.hsl.rl.ac.uk/catalog.html
!
!
!  Usage:
!
!  mex Ma57Mex (Ensure that source files are in the same folder)
!
!  x = Ma57Mex(A,b)
!  x = Ma57Mex(A,b,iter)
!
!  Input:
!  A - Sparse symmetric matrix
!  b - RHS vector (this vector must not be sparse)
!  iter - Number of iterations to be used for iterative refinement
!
!  Output:
!  x   - Solution vector
!
!  Jan 05 2010 Anthony R. Lamb
!  $Revision: 1.0 $


#include "fintrf.h"

      SUBROUTINE MEXFUNCTION(NLHS, PLHS, NRHS, PRHS)
      USE HSL_MA57_DOUBLE
      IMPLICIT NONE
      TYPE(ZD11_TYPE) MATRIX
      TYPE(MA57_CONTROL) CONTROL
      TYPE(MA57_AINFO) AINFO
      TYPE(MA57_FINFO) FINFO
      TYPE(MA57_SINFO) SINFO
      TYPE(MA57_FACTORS) FACTORS
      MWSIZE,PARAMETER::iwp=SELECTED_REAL_KIND(15)
```

```
      MWPOINTER :: PLHS(*), PRHS(*)
      INTEGER :: NLHS, NRHS, INFO
      MWPOINTER :: A_pr, b_pr,x_pr,x_res,max_it_pr,tol_pr,&
      flag_pr,iter_conv_pr,MXGETIR,MXGETJC, &
      MXGETNZMAX,A_ja, A_ia,MXCREATENUMERICMATRIX,&
      MXISSPARSE,MXCLASSIDFROMCLASSNAME, MXGETPR, &
      MXCREATEDOUBLEMATRIX, temp_pr
      MWSIZE MXGETM, MXGETN,N,Nnz, M
      MWSIZE  full, dig,max_it,ord
      MWINDEX i,j
      DOUBLE PRECISION MXGETSCALAR, ord_r
      MWINDEX , DIMENSION(:), ALLOCATABLE :: ip_in,irn_in,irn,jcn
      DOUBLE PRECISION, ALLOCATABLE, DIMENSION (:) :: X,B,a_val


!     For Windows only!
!     This resets the floating point exception to allow divide by
      zero, overflow and invalid numbers.

#if defined MSWIND
      INTEGER(2) CONTROL
      CALL GETCONTROLFPQQ(CONTROL)
      CONTROL = CONTROL .OR. FPCW$ZERODIVIDE
      CONTROL = CONTROL .OR. FPCW$INVALID
      CONTROL = CONTROL .OR. FPCW$OVERFLOW
      CALL SETCONTROLFPQQ(CONTROL)
#endif

!     CHECK FOR PROPER NUMBER OF ARGUMENTS
!
      IF   (NRHS .LT. 2)  THEN
        CALL MEXERRMSGTXT('Ma57ERROR: Ma57Mex REQUIRES TWO INPUT
                           ARGUMENTS MIN')
      ENDIF


!     CHECK IF MATRIX IS SPARSE
      full = MXISSPARSE(PRHS(1))
      IF (full .EQ. 0) THEN
       CALL MEXERRMSGTXT('Ma57ERROR: INPUT MATRIX MUST BE SPARSE')
      ENDIF


!     ASSIGN POINTERS TO THE VARIOUS PARAMETERS
      A_pr = MXGETPR(PRHS(1))
      A_ia = MXGETIR(PRHS(1))
      A_ja = MXGETJC(PRHS(1))
      b_pr = MXGETPR(PRHS(2))


      M     = MXGETM(PRHS(1)) ! ORDER OF MATRIX (Square matrix)
      N     = MXGETN(PRHS(2))
      Nnz    = MXGETNZMAX(PRHS(1)) ! NUMBER OF NON_ZERO ENTRIES


      ALLOCATE(ip_in(M+1))
```

```
      ALLOCATE(irn_in(Nnz))
      ALLOCATE(MATRIX%VAL(Nnz), MATRIX%ROW(Nnz), MATRIX%COL(Nnz))

      CALL mxCopyPtrToINTEGER8(A_ja, ip_in, M+1)
      CALL mxCopyPtrToINTEGER8(A_ia, irn_in, Nnz)
      CALL mxCopyPtrToReal8(A_pr,MATRIX%VAL,Nnz)

      DO i=1,M
        DO j=ip_in(i)+1,ip_in(i+1)
           MATRIX%COL(j) = i
           MATRIX%ROW(j) = irn_in(j)+1
        END DO
      END DO


!     Read matrix order and number of entries
      MATRIX%N  = M
      MATRIX%NE = Nnz

 !    Allocate arrays of appropriate sizes
      ALLOCATE(B(M), X(M))
      CALL MXCOPYPTRTOREAL8(b_pr, B, M) ! Copy rhs into B

!     INITIALIZE STRUCTURES
      CALL MA57_INITIALIZE(FACTORS,CONTROL)

!     ANALYSE
      CALL MA57_ANALYSE(MATRIX,FACTORS,CONTROL,AINFO)
      IF(AINFO%FLAG<0) THEN
       CALL MEXERRMSGTXT('Ma57ERROR: Failure of MA57_ANALYSE with
                         AINFO%FLAG=', AINFO%FLAG)
      END IF

!     FACTORIZE
      CALL MA57_FACTORIZE(MATRIX,FACTORS,CONTROL,FINFO)
      IF(FINFO%FLAG<0) THEN
        CALL MEXERRMSGTXT('Ma57ERROR: Failure of MA57_FACTORIZE
                          with FINFO%FLAG=', FINFO%FLAG)
      END IF

      IF  (NRHS .EQ. 4)  THEN
        ord = MXGETSCALAR(PRHS(4))
        CONTROL%ORDERING = INT(ord)
      ENDIF

!     SOLVE WITHOUT REFINEMENT
      X = B
      CALL MA57_SOLVE(MATRIX,FACTORS,X,CONTROL,SINFO)

      IF   (NRHS .GE. 3) THEN
        max_it = MXGETSCALAR(PRHS(3))
      ENDIF

!     SOLVE WITH REFINEMENT IF ITERATION VALUE GIVEN
      IF   ((NRHS .GE. 3) .AND. (INT(max_it).NE.0))  THEN
            DO i= 1,max_it
```

```
                CALL MA57_SOLVE(MATRIX,FACTORS,X,CONTROL,SINFO,B)
              ENDDO
        ENDIF

!       SET UP OUTPUT POINTERS
        PLHS(1) = MXCREATEDOUBLEMATRIX(M, N, 0)
        x_res = MXGETPR(PLHS(1))

        IF(SINFO%FLAG==0) THEN
          CALL MXCOPYREAL8TOPTR(X,x_res, M)
        ELSE

          CALL MEXERRMSGTXT('Ma57ERROR: No Solution obtained with
                            SINFO%FLAG=', SINFO%FLAG)
        ENDIF

        INFO = INT(INFO)
!       CLEAN UP

DEALLOCATE(MATRIX%VAL,MATRIX%ROW,MATRIX%COL,B,X,irn_in,ip_in)
        CALL MA57_FINALIZE(FACTORS,CONTROL,INFO)

        RETURN

        CALL MXFREE (x_res)
        CALL MXFREE (A_pr)
        CALL MXFREE (A_ia)
        CALL MXFREE (A_ja)
        CALL MXFREE (b_pr)
        CALL MXDESTROYARRAY(PRHS)

      END SUBROUTINE MEXFUNCTION
```

A2 VoroMex.cpp

```
/*This subroutine serves as an interface to Voro++ by Chris
  Rycroft.
  http://math.lbl.gov/voro++/about.html
  Using the 2D version supplied by Chris

  Usage:

  mex voro++Mex (Ensure that source files are in the same folder)

  [vIndex nID] = voroCMex(nodes,box,lscale)

  Input:
  nodes - particle coordinates
  box - bounding box [minX maxX; minY maxY]
  lscale - lengthscale
```

```
  Output:
  vIndex   - Voronoi cell coordinates
  nID      - particle id of associated Voronoi cell

  May 30 2010 Anthony R. Lamb
  $Revision: 1.0 $*/


#include <stdio.h>
#include <search.h>
#include <malloc.h>
#include "mex.h"
#include "matrix.h"
#include <cstring>
#include "cell_2d.cc"
#include "container_2d.cc"


/*A guess for the memory allocation per region*/
const int memory=8;

/*A maximum allowed number of regions, to prevent enormous amounts
of memory  being allocated*/
const int max_regions=16777216;


void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray *prhs[] )
{

      bool xperiodic=false,yperiodic=false;
      char buffer[256];
      double *nodes;
      double *box;
      double lscale;
      double *xValues, *yValues;
      double *pointer;
      int nPts,i;
      double x,y,minX, maxX, minY, maxY;

      /* Check for proper number of arguments. */
      if(nrhs!=3) {
            mexErrMsgTxt("Three input arguments are required.");
      } else if(nlhs!=2) {
            mexErrMsgTxt("Two output arguments are required.");
      }

      /* Assign pointers to each input.*/
      nodes   = mxGetPr(prhs[0]);
      box     = mxGetPr(prhs[1]);
      lscale  = mxGetScalar(prhs[2]);

      /* Determine number of points*/
      nPts = mxGetM(prhs[0]);
```

```
/*Assign values to bounding box*/
  minX = box[0];
  maxX = box[2];
  minY = box[1];
  maxY = box[3];

  lscale =1.8/lscale ;


/*Compute the number regions based on the length scale
  provided. If the total number exceeds a cutoff then bail
  out,to prevent making a massive memory allocation. Do this
  test using floating point numbers, since huge integers could
  potentially wrap around to negative values.*/

  double nxf=(maxX-minX)*lscale +1;
  double nyf=(maxY-minY)*lscale +1;


  if (nxf*nyf>max_regions){
        mexErrMsgTxt("voro++: Number of computational blocks
                      exceeds the maximum allowed of: " );}

/*Now that we are confident that the number of regions is
  reasonable,create integer versions of them.*/
  int nx=int(nxf);
  int ny=int(nyf);



/*Create a container with the geometry given above, and make
  it non-periodic in each of the three coordinates. Allocate
  space for eight particles within each computational block*/

  container_2d con(minX,maxX,minY,maxY,nx,ny,
                   xperiodic,yperiodic,memory);

  /* Add particles to container*/
  for(i=0;i<nPts;i++) {
        x= nodes[i];
        y= nodes[i+nPts];
        con.put(i,x,y);}

  /* Creation of Voronoi cells*/
  voronoicell_2d vCell;
  int j,k,ij=0,q,ct=0,nVert,vPt;
  int* nID = new int [nPts];


  /*Create cell array to store vCell indicies*/
  mxArray* vCell_out;
  /*Create pointer to vCell indicies*/
  double *pr;
  /*Create return cell arary*/
  plhs[0] = mxCreateCellMatrix(nPts, 1);
  /* Create a nPts-by-1 mxArray to store nodal ID*/
```

```
    plhs[1] = mxCreateNumericMatrix(nPts, 1, mxINT32_CLASS,
                                    mxREAL);
    pointer = mxGetPr(plhs[1]);

    for(j=0;j<ny;j++) for(i=0;i<nx;i++,ij++)
      for(q=0;q<con.getco()[ij];q++){
        x=con.getp()[ij][2*q];y=con.getp()[ij][2*q+1];
          if(!con.compute_cell_sphere(vCell,i,j,ij,q,x,y))
          continue;
          nVert = vCell.p;
          /* Only if nVert greater than zero will we store
             data*/
          if (nVert>0){
                mwIndex k=0;
                nID[ct] = (con.getID()[ij][q])+ 1 ;
              vCell_out = mxCreateDoubleMatrix(2,nVert, mxREAL);
              pr = mxGetPr(vCell_out);
                do {
                      *pr++ = x+0.5*vCell.pts[2*k];
                      *pr++ = y+0.5*vCell.pts[2*k+1];
                      k = vCell.ed[k][0];
                } while (k!=0);
                      /*Attach cell data to Matlab cell array*/
                      mxSetCell(plhs[0],ct,vCell_out);
          ct++;
          }
    }

  /* Copy data into the mxArray */
  memcpy(pointer,nID,nPts*sizeof(int));

  /* Clean up arrays */
  mxFree(nID);
  mxDestroyArray(vCell_out);
}
```

Appendix B

B1 Galerkin Method

Mathematical models are used to describe many systems in nature whether mechanical, geological or biological. These models take the form of governing equations with associated boundary conditions and/or initial conditions. Governing equations usually take the form of ordinary differential, partial differential or integral equations. Corresponding boundary and/or initial conditions are also required for the determining the field variables described by the governing equations.

Obtaining the exact solution of derived governing equations is very difficult for many of the complex phenomena that occur in nature. In contrast an approximate solution can be obtained by introducing an integral operation to the governing equations. The resulting equations are known as the weak-form and usually produce very stable discretized system equations that give accurate results.

The Galerkin method, a weighted residual method, is used to develop the integral form of a set of governing equations. Consider the following boundary value problem:

$$\mathbf{A}(\mathbf{u}) = \mathbf{C}(\mathbf{u}) + \mathbf{p} = \mathbf{0} \text{ in problem domain } \Omega, \tag{B1}$$

where $\mathbf{C}$ is a differential operator and $\mathbf{p}$ is a known function independent of the field variable $\mathbf{u}$. The boundary condition is given as:

$$\mathbf{B}(\mathbf{u}) = \mathbf{M}(\mathbf{u}) + \mathbf{q} = \mathbf{0} \text{ on the boundary } \Gamma, \tag{B2}$$

where **M** is also a differential operator and **q** is a known function independent of **u** which is the exact solution of the boundary value problem. Only in very simple instances can an exact solution be obtained for the system described by (B1) and (B2). For most practical problems only an approximate solution can be obtained. Using the Galerkin method the field variable **u** is firstly approximated by:

$$\mathbf{u} \approx \mathbf{u}^h = \sum_i^n N_i a_i = \mathbf{Na}, \tag{B3}$$

where N_i is the ith term basis function or shape function (usually a polynomial of x, y, z), a_i is the unknown coefficient for the ith term basis function and n is the number of basis functions used. The introduction of the approximate solution $\mathbf{u}^h$ introduces an error called the residual error. This residual error **R**, is given by:

$$\mathbf{R} = \mathbf{R}_\Omega + \mathbf{R}_\Gamma = \mathbf{A}(\mathbf{u}^h) + \mathbf{B}(\mathbf{u}^h). \tag{B4}$$

The approximate method revolves around minimising the residual error over the entire domain Ω and boundary Γ. This requires that the sum of the integrals of the error over Ω and Γ be equal to zero. Therefore:

$$\int_\Omega \mathbf{w}^{\mathrm{T}}\, \mathbf{A}(\mathbf{u}^h)\, d\Omega + \int_\Gamma \bar{\mathbf{w}}^{\mathrm{T}}\, \mathbf{B}(\mathbf{u}^h)\, d\Gamma = 0, \tag{B5}$$

where **w** and $\bar{\mathbf{w}}$ are weighting functions applied to the domain and the boundary respectively. It should be noted that if (B5) is satisfied for a large number of arbitrary weighting functions, the approximate solution will approach the exact solution.

B2 Application of Galerkin method

This section illustrates the application of the Galerkin method in the development of the finite element formulation for the equations describing the mechanics of coupled deformation and fluid flow in porous media outlined in Chapter 2. Applying the integral equation (B5) to (2.2) and (2.17) gives:

$$\int_{\Omega} \mathbf{w}_u^{\mathrm{T}} \left(\nabla_{\mathrm{s}}^{\mathrm{T}} \boldsymbol{\sigma} + \rho \mathbf{g}\right) d\boldsymbol{\Omega} + \int_{\Gamma} \bar{\mathbf{w}}_u^{\mathrm{T}} (\mathbf{n} \cdot \sigma - \bar{\mathbf{t}}) \, d\boldsymbol{\Gamma} = 0, \tag{B6}$$

in which $\mathbf{w}_u$ and $\overline{\mathbf{w}}_u$ are the weighting functions. In this case the weighting functions are given such that:

$$\begin{aligned} \mathbf{w}_u &= 0 \text{ on } \Gamma_u, \\ \bar{\mathbf{w}}_u &= -\mathbf{w}_u \text{ on } \Gamma_u^q. \end{aligned} \tag{B7}$$

Green's theorem, given below, is now applied the first part of (B6).

$$\int_{\Omega} \phi \frac{\partial \Psi}{\partial x} d\Omega = -\int_{\Omega} \frac{\partial \phi}{\partial x} \Psi d\Omega + \int_{\Gamma} \phi \Psi n_x d\Gamma. \tag{B8}$$

Equation (B6) then becomes

$$-\int_{\Omega} (\nabla_{\mathrm{s}} \mathbf{w}_u)^{\mathrm{T}} \boldsymbol{\sigma} \, d\Omega + \int_{\Omega} \mathbf{w}_u^{\mathrm{T}} \rho \mathbf{g} \, d\Omega + \int_{\Gamma} \mathbf{w}_u^{\mathrm{T}} \bar{\mathbf{t}} \, d\Gamma = 0. \tag{B9}$$

The weighted residual approach is now applied to the continuity equations for fluid flow (2.9) and to the natural boundary conditions (2.13) and (2.14). The weighting functions for the domain and boundary are now represented by $\mathbf{w}_p$ and $\overline{\mathbf{w}}_p$ respectively and satisfy conditions similar to those given by (B7). After applying Green's theorem to the divergence operator the resulting equation takes the form:

$$\begin{aligned} \int_{\Omega} \Bigg[-\left(\nabla \mathbf{w}_p\right)^{\mathrm{T}} \left(\frac{\mathbf{K}}{\mu_w} (-\nabla P_w + \rho_w \mathbf{g}) \right) + \mathbf{w}_p{}^{\mathrm{T}} \alpha \mathbf{I}^{\mathrm{T}} \nabla_{\mathrm{s}} \frac{\partial \mathbf{u}}{\partial t} \\ + \mathbf{w}_p{}^{\mathrm{T}} \left(\frac{\alpha - n}{K_s} + \frac{n}{K_w} \right) \frac{\partial P_w}{\partial t} \Bigg] d\Omega + \int_{\Gamma_w^q} \mathbf{w}_p{}^{\mathrm{T}} \frac{q_w}{\rho_w} \, \mathrm{d}\Gamma = 0. \end{aligned} \tag{B10}$$

The following expressions for $\mathbf{u}$ and P_w are now substituted into (B9) and (B10):

$$\begin{aligned} \mathbf{u} &\approx \mathbf{u}^h = \mathbf{N}_u \bar{\mathbf{u}}, \\ P_w &\approx P_w^h = \mathbf{N}_p \overline{\mathbf{P}}_w, \end{aligned} \tag{B11}$$

where $\bar{\mathbf{u}}$ and $\overline{\mathbf{P}}_w$ are the vectors of nodal unknowns of displacement and excess pressures respectively. The weighting functions $\mathbf{w}_u$ and $\mathbf{w}_p$ are also replaced by the shape functions $\mathbf{N}_u$, the displacement shape function and $\mathbf{N}_p$, the pressure shape function respectively. The resulting equations are given below:

$$\int_\Omega (\nabla_s \mathbf{N}_u)^{\mathrm{T}} \boldsymbol{\sigma}\, d\Omega = \int_\Omega \mathbf{N}_u^{\mathrm{T}} \rho \mathbf{g}\, d\Omega + \int_\Gamma \mathbf{N}_u^{\mathrm{T}} \bar{\mathbf{t}}\, d\Gamma \tag{B12}$$

$$\begin{aligned} \int_\Omega \Bigg[-\left(\nabla \mathbf{N}_p\right)^{\mathrm{T}} \left(\frac{\mathbf{K}}{\mu_w} \left(-\nabla \mathbf{N}_p \overline{\mathbf{P}}_w + \rho_w \mathbf{g} \right) \right) + \mathbf{N}_p{}^{\mathrm{T}} \alpha \mathbf{I}^{\mathrm{T}} \nabla_s \mathbf{N}_u \frac{\partial \bar{\mathbf{u}}}{\partial t} \\ + \mathbf{N}_p{}^{\mathrm{T}} \left(\frac{\alpha - n}{K_s} + \frac{n}{K_w} \right) \mathbf{N}_p \frac{\partial \overline{\mathbf{P}}_w}{\partial t} \Bigg] d\Omega \\ + \int_{\Gamma_w^q} \mathbf{N}_p{}^{\mathrm{T}} \frac{q_w}{\rho_w}\, \mathrm{d}\Gamma = 0. \end{aligned} \tag{B13}$$

Considering equation (2.2), which describes force equilibrium of the soil-water mixture, and equation (2.4), which describes the total stress of the system, equations (B12) and (B13)can be rewritten as:

$$\mathbf{Ke}\, \bar{\mathbf{u}}\, d\Omega - \mathbf{L} \overline{\mathbf{P}}_w = \mathbf{F}^u, \tag{B14}$$

$$\mathbf{H} \overline{\mathbf{P}}_w + \mathbf{L}^{\mathrm{T}} \frac{\partial \bar{\mathbf{u}}}{\partial t} + \mathbf{S} \frac{\partial \overline{\mathbf{P}}_w}{\partial t} = \mathbf{F}^p. \tag{B15}$$

Equations (B14) and (B15) represent the fully coupled discretized equations describing deformation and fluid flow in porous media.

Definitions of the terms given in (B14) and (B15) are given below.

Ke is the elastic stiffness matrix,

L is the coupling matrix,

H is the permeability matrix,

S is the compressibility matrix,

$\mathbf{F}^u$ is the compressibility matrix,

and $\mathbf{F}^p$ is the force vector due to the flow field.

The elements of these matrices are:

$$\mathbf{Ke} = \int_{\Omega} \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B}\, d\Omega, \tag{B16}$$

$$\mathbf{L} = \int_{\Omega} \mathbf{B}^{\mathrm{T}} \alpha \mathbf{I} \mathbf{N}_p\, d\Omega, \tag{B17}$$

$$\mathbf{H} = \int_{\boldsymbol{\Omega}} \left(\nabla \mathbf{N}_p\right)^{\mathrm{T}} \frac{\mathbf{K}}{\mu_w} \nabla \mathbf{N}_p\, d\Omega, \tag{B18}$$

$$\mathbf{S} = \int_{\Omega} \mathbf{N}_p^{\mathrm{T}} \left(\frac{\alpha - n}{K_s} + \frac{n}{K_w}\right) \mathbf{N}_p\, d\Omega, \tag{B19}$$

$$\mathbf{F}^u = \int_{\Omega} \mathbf{N}_u^{\mathrm{T}} \rho \mathbf{g}\, d\Omega + \int_{\Gamma_u^q} \mathbf{N}_u^{\mathrm{T}} \bar{\mathbf{t}}\, d\Gamma, \tag{B20}$$

$$\mathbf{F}^p = \int_{\Omega} \left(\nabla \mathbf{N}_p\right)^{\mathrm{T}} \frac{\mathbf{K}}{\mu_w} \rho^w \mathbf{g}\, d\Omega - \int_{\Gamma_w^q} \mathbf{N}_p^{\mathrm{T}} \frac{q^w}{\rho^w}\, d\Gamma, \tag{B21}$$

$$\mathbf{B} = \nabla_{\mathrm{s}} \mathbf{N}_u = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \mathbf{N}_u, \tag{B22}$$

$$\nabla \mathbf{N}_p = \begin{bmatrix} \frac{\partial}{\partial x} \mathbf{N}_p \\ \frac{\partial}{\partial y} \mathbf{N}_p \end{bmatrix}. \tag{B23}$$

The discretized equations describing coupled deformation and fluid flow can be conveniently cast into matrix form and re-written as:

$$\begin{bmatrix} \mathbf{Ke} & -\mathbf{L} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}} \\ \bar{\mathbf{P}}_w \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{S} \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \bar{\mathbf{u}} \\ \bar{\mathbf{P}}_w \end{bmatrix} = \begin{bmatrix} \mathbf{F}^u \\ \mathbf{F}^p \end{bmatrix}. \tag{B24}$$

Since (B24) is not symmetric, symmetry can be restored by time differentiating the lhs of (B14) and multiplying by -1 (Lewis and Shrefler, 1998). This shifts the component matrices horizontally in the global matrix. The final equations take the form:

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{u}} \\ \bar{\mathbf{P}}_w \end{bmatrix} + \begin{bmatrix} -\mathbf{Ke} & \mathbf{L} \\ \mathbf{L}^{\mathrm{T}} & \mathbf{S} \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \bar{\mathbf{u}} \\ \bar{\mathbf{P}}_w \end{bmatrix} = \begin{bmatrix} -\mathbf{F}^u \\ \mathbf{F}^p \end{bmatrix}. \tag{B25}$$

Appendix C

C1 Nodal Integration

The integrals of any discrete variable are numerically determined over individual elements and summed for all elements to obtain the global integration. Using the Guass quadrature scheme the numerical integration can be written as:

$$\int_{\Omega} f(x,y)\, d\Omega = \sum_{k}^{ne} \sum_{i=1}^{ng} w_i f(x_{ip}, y_{ip})\, |\boldsymbol{J}_{ik}| \tag{C1}$$

where $f(x,y)$ is the integrand applicable over domain Ω, w_i is the Gauss weighting factor for the ith Gauss point at (x_{ip}, y_{ip}) and J_{ik} is the Jacobian matrix for the finite element k. ne and ng are the number of elements and Gaussian integration points respectively.

NI-RPIM uses the Taylor series extension to carry out the integration over an integration cell. The integral function is extended to include additional terms of the Taylor series, and the integration will be approximately performed on these terms. Therefore the Taylor series extension of the integrand $f(x,y)$ can be written as a two-dimensional (2D) continuous function approximated in the vicinity of a point (x_0, y_0) as follows:

$$\begin{aligned} f(x,y) \approx f(x_0,y_0) &+ \left(x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}\right) f(x_0,y_0) \\ &+ \frac{1}{2!}\left(x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}\right)^2 f(x_0,y_0) \end{aligned} \tag{C2}$$

The integral over the nodal integration domain Ω_i (Voronoi cell i) can be expressed as:

$$\int\int_{\Omega_i} f(x,y)\,d\Omega \approx \int\int_{\Omega_i}\left(f(x_0,y_0) + \left(x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}\right) f(x_0,y_0) + \frac{1}{2!}\left(x\frac{\partial}{\partial x} + y\frac{\partial}{\partial y}\right)^2 f(x_0,y_0)\right) d\Omega$$

$$\begin{aligned} &= f(x_0,y_0)\int\int_{\Omega_i} 1\,d\Omega + f_{\partial x}(x_0,y_0)\int\int_{\Omega_i} x\,d\Omega \\ &\quad + f_{\partial x}(x_0,y_0)\int\int_{\Omega_i} y\,d\Omega \\ &\quad + \frac{1}{2} f_{\partial xx}(x_0,y_0)\int\int_{\Omega_i} x^2 d\Omega \\ &\quad + f_{\partial xy}(x_0,y_0)\int\int_{\Omega_i} xy\,d\Omega \\ &\quad + \frac{1}{2} f_{\partial xx}(x_0,y_0)\int\int_{\Omega_i} x^2 d\Omega \end{aligned} \tag{C3}$$

$$\begin{aligned} = f(x_0,y_0)A_i + f_{\partial x}(x_0,y_0)M_{yi} + f_{\partial y}(x_0,y_0)M_{xi} \\ + \frac{1}{2} f_{\partial xx}(x_0,y_0)M_{yyi} + f_{\partial xy}(x_0,y_0)M_{xyi} \\ + \frac{1}{2} f_{\partial yy}(x_0,y_0)M_{xxi} \end{aligned}$$

where $A_i$is the area of the nodal integration domain of the i^{th} node,

$$M_{xi} = \int\int_{A_i} y\,dA_i \qquad M_{yi} = \int\int_{A_i} x\,dA_i \tag{C4}$$

are the area moments of the first-order for the integration domain of the i^{th} node, and

$$M_{xxi} = \int\int_{A_i} y^2 dA_i \quad M_{yyi} = \int\int_{A_i} x^2 dA_i \quad M_{xyi} = \int\int_{A_i} xy\,dA_i \tag{C5}$$

are the area moments of second-order for the integration domain of the ith node.

C2 RPIM shape functions

The radial point interpolation method (RPIM) is formulated based on the Galerkin weak form using shape functions that satisfy the Kronecker delta function property. These shape functions are created through an interpolation using local nodes with both radial and polynomial basis functions. Such an interpolation is often used by many researchers for curve or surface fitting and function approximation (Golberg, 1996). The procedure of constructing RPIM shape functions is as follows.

The function $u(\mathbf{x})$ is approximated using radial and polynomial basis functions in the form of

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{n} R_i(\mathbf{x}) a_i + \sum_{j=1}^{m} P_j(\mathbf{x}) b_j = \mathbf{R}^{\mathrm{T}}(\mathbf{x})\mathbf{a} + \mathbf{P}^{\mathrm{T}}(\mathbf{x})\mathbf{b}, \tag{C6}$$

where $R_i(\mathbf{x})$ and $P_j(\mathbf{x})$ are radial polynomial basis functions in two dimensional space $\mathbf{x}^{\mathrm{T}} = [x, y]$, n is the number of field nodes in the local support domain of point $\mathbf{x}$, m is the number of terms of polynomial basis functions, and a_i and b_j are coefficients for $R_i(\mathrm{x})$ and $P_j(\mathrm{x})$ respectively. These vectors are defined as:

$$\begin{aligned} \mathbf{R}(\mathbf{x}) &= [R_1(\mathbf{x}), R_2(\mathbf{x}), R_3(\mathbf{x}), \dots, R_n(\mathbf{x})]^{\mathrm{T}}, \\ \mathbf{P}(\mathbf{x}) &= [P_1(\mathbf{x}), P_2(\mathbf{x}), P_3(\mathbf{x}), \dots, P_n(\mathbf{x})]^{\mathrm{T}}, \\ \mathbf{a} &= [a_1, a_2, a_3, \dots, a_n]^{\mathrm{T}}, \\ \mathbf{b} &= [b_1, b_2, b_3, \dots, b_m]^{\mathrm{T}}. \end{aligned} \tag{C7}$$

The radial basis function $R_i(\mathbf{x})$ has the following general form:

$$R_i(\mathrm{x}) = R(r_i), \tag{C8}$$

where r_i is the distance between the interpolating point $\mathbf{x}$ and field node $\mathbf{x}_i$ and is expressed as:

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad \text{for 2-D problems.} \tag{C9}$$

The polynomial basis vector $\mathbf{P}(\mathbf{x})$ has the following form for two dimensional problems:

$$\mathbf{P}(\mathbf{x}) = [1, x, y, x^2, xy, y^2, \dots]^{\mathrm{T}}. \tag{C10}$$

In order to determine a_i and b_j of (C6), a support domain is constructed for the point of interest at $\mathbf{x}$, with n field nodes included in the support domain (see Fig. 3.6). The coefficient vectors $\mathbf{a}$ and $\mathbf{b}$ are then determined by ensuring that the displacement of all the n field nodes within the local support domain satisfy (C6). This leads to n linear equations, one for each node and can be expressed in the following matrix form:

$$\mathbf{U}_s = \mathbf{R}_0\mathbf{a} + \mathbf{P}_m\mathbf{b}, \tag{C11}$$

where $\mathbf{U}_s$ is the vector of approximate function values is given. $\mathbf{R}_0$ and $\mathbf{P}_m$ represent the moment matrix of radial basis functions and polynomial moment matrix respectively. To ensure that the linear equations have a unique solution the following constraint is also added:

$$\sum_{i=1}^{n} P_j(x_i, y_i)a_i = 0 \qquad j = 1,2, \dots, m. \tag{C12}$$

The system equations are now expressed as:

$$\begin{bmatrix}\mathbf{U}_s \\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\mathbf{R}_0 & \mathbf{P}_m \\ \mathbf{P}_m^{\mathrm{T}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{a} \\ \mathbf{b}\end{bmatrix} = \mathbf{G}\begin{bmatrix}\mathbf{a} \\ \mathbf{b}\end{bmatrix}. \tag{C13}$$

The function approximation is now expressed as:

$$\mathbf{u}(\mathbf{x}) = [\mathbf{R}^{\mathrm{T}}(\mathbf{x})\mathbf{P}^{\mathrm{T}}(\mathbf{x})]\mathbf{G}^{-1}\begin{bmatrix}\mathbf{U}_s \\ \mathbf{0}\end{bmatrix} = \mathbf{N}(\mathbf{x})\mathbf{U}_s, \tag{C14}$$

where the matrix of shape functions $\mathbf{N}(\mathbf{x})$ is defined by:

$$\mathbf{N}(\mathbf{x}) = [N_1(\mathbf{x}), N_2(\mathbf{x}), N_3(\mathbf{x}), \dots, N_n(\mathbf{x})], \tag{C15}$$

where the k^{th} shape function is defined as:

$$N_k(\mathbf{x}) = \sum_{i=1}^{n} B_i(\mathbf{x})\bar{G}_{i,k} + \sum_{j=1}^{m} P_j(\mathbf{x})\bar{G}_{n+j,k}. \tag{C16}$$

where $\bar{G}_{i,k}$ is the (i,k) element of $\mathbf{G}^{-1}$. Using the inverse of matrix $\mathbf{G}$ the shape functions are obtained via:

$$\begin{aligned} \frac{\partial N_k}{\partial x} &= \sum_{i=1}^{n} \frac{\partial B_i}{\partial x}\bar{G}_{i,k} + \sum_{j=1}^{m} \frac{\partial P_j}{\partial x}\bar{G}_{n+j,k} \\ \frac{\partial N_k}{\partial y} &= \sum_{i=1}^{n} \frac{\partial B_i}{\partial y}\bar{G}_{i,k} + \sum_{j=1}^{m} \frac{\partial P_j}{\partial y}\bar{G}_{n+j,k} \end{aligned} \tag{C17}$$

The radial basis function used in this work is the multi-quadratics radial basis function (MQ-RBF) given by:

$$R_i(x,y) = (r_i^2 + (\alpha_c d_c)^2)^q, \tag{C18}$$

where d_c is the characteristic length that relates to the nodal spacing in the support domain of the field node $\mathbf{x}$ and is usually taken as the average nodal spacing for all the nodes in the support domain, q and α_c are shape parameters. Liu and Gu (2005) has demonstrated that MQ-RBF produce acceptable results for 2D solid and fluid mechanics problems when compared to the Gaussian, thin plate spline and logarithmic radial basis functions.

Appendix D

D1 XFEM discrete equations

This appendix presents the expanded form of the discrete system equations given in Chapter 4.

The discrete system is given by:

$$\mathbf{AX} + \mathbf{B}\frac{\mathbf{dX}}{\mathbf{dt}} = \mathbf{C} \tag{D1}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{H}_m + \mathbf{H}_{T_r} & -\mathbf{H}_{T_r} \\ 0 & 0 & 0 & -\mathbf{H}_{T_r} & \mathbf{H}_f + \mathbf{H}_{T_r} \end{bmatrix}, \tag{D2}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{Ke}_{uu} & \mathbf{Ke}_{ua} & \mathbf{Ke}_{ub} & -\mathbf{L} & 0 \\ \mathbf{Ke}_{au} & \mathbf{Ke}_{aa} & \mathbf{Ke}_{ab} & -\mathbf{L}_a & 0 \\ \mathbf{Ke}_{bu} & \mathbf{Ke}_{ba} & \mathbf{Ke}_{bb} & -\mathbf{L}_b & 0 \\ \mathbf{L}^{\mathrm{T}} & \mathbf{L}_a^{\mathbf{T}} & \mathbf{L}_b^{\mathbf{T}} & \mathbf{S}_m & 0 \\ 0 & 0 & 0 & 0 & \mathbf{S}_f \end{bmatrix}, \tag{D3}$$

$$\mathbf{X} = \begin{bmatrix} \overline{\mathbf{u}} \\ \mathbf{a} \\ \mathbf{b} \\ \overline{\mathbf{P}}_{wm} \\ \overline{\mathbf{P}}_{wf} \end{bmatrix}, \tag{D4}$$

$$\mathbf{C} = \begin{bmatrix} \mathrm{d}\mathbf{F} \\ \mathrm{d}\mathbf{F}_a \\ \mathrm{d}\mathbf{F}_b \\ \bar{\mathbf{f}} \\ 0 \end{bmatrix}. \tag{D5}$$

The elements of the above listed matrices are given by:

$$\mathbf{Ke}_{uu} = -\int_{\Omega} \mathbf{B}^{\mathrm{T}}\mathbf{DB}\, d\boldsymbol{\Omega}_m, \tag{D6}$$

$$\mathbf{Ke}_{ua} = -\int_{\Omega} \mathbf{B}^{\mathrm{T}}\mathbf{DB}^{a}\, d\boldsymbol{\Omega}_m, \tag{D7}$$

$$\mathbf{Ke}_{ub} = -\int_{\Omega} \mathbf{B}^{\mathrm{T}} \mathbf{D} \mathbf{B}^{b} \, d\boldsymbol{\Omega}_{m}, \tag{D8}$$

$$\mathbf{Ke}_{au} = -\int_{\Omega} \mathbf{B}^{a\,\mathrm{T}} \mathbf{D} \mathbf{B} \, d\boldsymbol{\Omega}_{m}, \tag{D9}$$

$$\mathbf{Ke}_{aa} = -\int_{\Omega} \mathbf{B}^{a\,\mathrm{T}} \mathbf{D} \mathbf{B}^{a} \, d\boldsymbol{\Omega}_{m}, \tag{D10}$$

$$\mathbf{Ke}_{ab} = -\int_{\Omega} \mathbf{B}^{a\,\mathrm{T}} \mathbf{D} \mathbf{B}^{b} \, d\boldsymbol{\Omega}_{m}, \tag{D11}$$

$$\mathbf{Ke}_{bu} = -\int_{\Omega} \mathbf{B}^{b\,\mathrm{T}} \mathbf{D} \mathbf{B} \, d\boldsymbol{\Omega}_{m}, \tag{D12}$$

$$\mathbf{Ke}_{ba} = -\int_{\Omega} \mathbf{B}^{b\,\mathrm{T}} \mathbf{D} \mathbf{B}^{a} \, d\boldsymbol{\Omega}_{m}, \tag{D13}$$

$$\mathbf{Ke}_{bb} = -\int_{\Omega} \mathbf{B}^{b\,\mathrm{T}} \mathbf{D} \mathbf{B}^{b} \, d\boldsymbol{\Omega}_{m}, \tag{D14}$$

$$\mathbf{L} = \int_{\Omega} \mathbf{B}^{\mathrm{T}} \mathbf{I} \mathbf{N}_{p} \, d\boldsymbol{\Omega}_{m}, \tag{D15}$$

$$\mathbf{L}_{a} = \int_{\Omega} \mathbf{B}^{a\,\mathrm{T}} \mathbf{I} \mathbf{N}_{p} \, d\boldsymbol{\Omega}_{m}, \tag{D16}$$

$$\mathbf{L}_{b} = \int_{\Omega} \mathbf{B}^{b\,\mathrm{T}} \mathbf{I} \mathbf{N}_{p} \, d\boldsymbol{\Omega}_{m}, \tag{D17}$$

$$\mathbf{F} = -\int_{\Omega} \mathbf{N}_{u}^{\mathrm{T}} \rho \mathbf{g} \, \mathrm{d}\boldsymbol{\Omega}_{m} - \int_{\Gamma} \mathbf{N}_{u}^{\mathrm{T}} \bar{\mathbf{t}} \, d\boldsymbol{\Gamma}_{m}, \tag{D18}$$

$$\mathbf{F}_{a} = -\int_{\Omega} H \mathbf{N}_{u}^{\mathrm{T}} \rho \mathbf{g} \, d\boldsymbol{\Omega}_{m} - \int_{\Gamma} H \mathbf{N}_{u}^{\mathrm{T}} \bar{\mathbf{t}} \, d\boldsymbol{\Gamma}_{m}, \tag{D19}$$

$$\mathbf{F}_{b} = -\int_{\Omega} \Phi_{\alpha} \mathbf{N}_{u}^{\mathrm{T}} \rho \mathbf{g} \, d\boldsymbol{\Omega} - \int_{\Gamma} \Phi_{\alpha} \mathbf{N}_{u}^{\mathrm{T}} \bar{\mathbf{t}} \, d\boldsymbol{\Gamma}_{m}, \tag{D20}$$

$$\mathbf{H}_{m} = \int_{\Omega} \left(\nabla \mathbf{N}_{p}\right)^{\mathrm{T}} \frac{\mathbf{K}_{m}}{\mu_{w}} \nabla \mathbf{N}_{p} \, d\boldsymbol{\Omega}_{m}, \tag{D21}$$

$$\mathbf{H}_{f} = \int_{\Omega} \left(\nabla \mathbf{N}_{p}\right)^{\mathrm{T}} \frac{\bar{\mathrm{s}} \mathbf{K}_{f}}{\mu_{w}} \nabla \mathbf{N}_{p} \, d\boldsymbol{\Omega}_{f}, \tag{D22}$$

$$\mathbf{H}_{T_r} = \int_{\Omega} \mathbf{N}_{p}^{\mathrm{T}} \frac{\mathbf{K}_{f}}{\mu_{w}} \mathbf{N}_{p} d\boldsymbol{\Omega}_{m\&f}, \tag{D23}$$

$$\mathbf{S}_{m} = \int_{\Omega} \mathbf{N}_{p}^{\mathrm{T}} \frac{n_{m}}{K_{w}} \mathbf{N}_{p} \, d\boldsymbol{\Omega}_{m}, \tag{D24}$$

$$\mathbf{S}_{f} = \int_{\Omega} \mathbf{N}_{p}^{\mathrm{T}} \frac{n_{f}}{K_{w}} \mathbf{N}_{p} \, d\boldsymbol{\Omega}_{f}, \tag{D25}$$

$$\bar{\mathbf{f}} = \int_{\Gamma_w^q} \mathbf{N}_p^{\mathrm{T}} \frac{\mathbf{K}_m}{\mu_w} \rho_w \mathbf{g} \cdot \mathbf{n} \, d\boldsymbol{\Gamma}_m - \int_{\Gamma_w^q} \mathbf{N}_p^{\mathrm{T}} \frac{q_w}{\rho_w} \, d\boldsymbol{\Gamma}_m, \tag{D26}$$

$$\mathbf{B} = \nabla_{\mathrm{s}} \mathbf{N}_u = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \mathbf{N}_u, \tag{D27}$$

$$\begin{aligned} \mathbf{B}^a &= \nabla_{\mathrm{s}} \widehat{\mathbf{N}}_u \Big(H\big(\phi(\mathbf{x})\big) - H\big(\phi(\mathbf{x})\big) \Big) \\ &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \widehat{\mathbf{N}}_u \Big(H\big(\phi(\mathbf{x})\big) - H\big(\phi(\mathbf{x})\big) \Big), \end{aligned} \tag{D28}$$

$$\begin{aligned} \mathbf{B}^b &= \nabla_{\mathrm{s}} \widehat{\mathbf{N}}_u \Big(\Phi_\gamma(\mathbf{x}) - \Phi_\gamma(\mathbf{x}_K) \Big) \\ &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \widehat{\mathbf{N}}_u \Big(\Phi_\gamma(\mathbf{x}) - \Phi_\gamma(\mathbf{x}_K) \Big), \end{aligned} \tag{D29}$$

$$\nabla \mathbf{N}_p = \begin{bmatrix} \frac{\partial}{\partial x} \mathbf{N}_p \\ \frac{\partial}{\partial y} \mathbf{N}_p \end{bmatrix}. \tag{D30}$$

Appendix E

E1 Global strain recovery in XFEM

This appendix presents a global strain recovery scheme implemented as part of this work. The method incorporates an enhanced discontinuous strain field presented by Duflot and Bordas (2008) into the standard least squares fit global strain recovery procedure (L_2 projection) to improve the accuracy of the recovered nodal strains obtained from XFEM. The procedure is presented within this appendix and the analytical solution of an infinite plate containing a fracture is used to illustrate the methods performance.

Introduction

Zienkiewicz *et al.* (1971) has demonstrated that strains determined at the Gaussian integration points, also known as superconvergent points, produce values that closely match the exact strains. The same level of accuracy of these fields is also required at element nodes and elsewhere within the element when energy norm and other norms are to be evaluated in error estimation calculations (Zienkiewicz *et al.*, 2005). To obtain accurate values elsewhere within an element local or global recovery techniques can be used.

Local recovery techniques are commonly used because they are computationally less intensive. Strains are calculated for the nodes of each element separately, or patches of elements formed by several adjacent

elements. Therefore, local strain recovery does not produce unique values of at the nodal points and calculation of nodal averages is required.

In global recovery techniques, values are calculated for the entire finite element domain globally. Strains are calculated at the Gaussian integration points, optimal calculation locations, within each element and are extrapolated to the domain nodes using a least squares fit technique. The resulting strains are more accurate using these superconvergent points as oppose to the nodal points.

Standard FEM yields a strain field that is continuous within its elements but discontinuous across interelement boundaries. Global recovery techniques are therefore used to project a continuous field onto the entire domain. When using XFEM the strain field within the elements containing a fracture or fracture tip becomes discontinuous and the standard global recovery no longer accurately reproduces a continuous strain field throughout the domain. To improve the accuracy of the recovered strain field this work uses an enhanced discontinuous strain field in the least squares fit of the strains determined at the element Gauss points.

E2 Least Squares Fit

This section outlines the least squares fit procedure. Let $\boldsymbol{\chi} = [\chi_1\ \chi_2 \dots \chi_n]^{\mathrm{T}}$ represent the element nodal values to be determined using a least squares fit. If $\mathbf{s} = [s_1\ s_2 \dots s_n]^{\mathrm{T}}$ represents the vector of interpolated values at the Gaussian integration points, and $\mathbf{a} = [a_1\ a_2 \dots a_n]^{\mathrm{T}}$ represents the actual values of the variable calculated at the integration points, the error may be defined as:

$$\boldsymbol{\epsilon} = \sum_{element} (\mathbf{s} - \mathbf{a})^{\mathrm{T}} (\mathbf{s} - \mathbf{a}). \tag{E1}$$

The relationship between the nodal values $\boldsymbol{\chi}$ and the interpolated values $\mathbf{s}$ is defined as:

$$\mathbf{s} = \mathbf{N}\boldsymbol{\chi}, \tag{E2}$$

where $\mathbf{N}$ represents the shape functions and takes the form:

$$\mathbf{N} = \begin{bmatrix} N_1^1 & N_2^1 & \dots & N_j^1 \\ N_1^2 & N_2^2 & \dots & N_j^2 \\ \vdots & \vdots & \dots & \vdots \\ N_1^i & N_2^i & \dots & N_j^i \end{bmatrix}, \tag{E3}$$

where N_j^i represents the shape function N_j evaluated at the interior point i. Inserting (E2) into (E1) the error becomes:

$$\boldsymbol{\epsilon} = \sum_{element} \boldsymbol{\chi}^{\mathrm{T}} \mathbf{N}^{\mathrm{T}} \mathbf{N} \boldsymbol{\chi} - 2\boldsymbol{\chi}^{\mathrm{T}} \mathbf{N}^{\mathrm{T}} \mathbf{a} + \mathbf{a}^{\mathrm{T}} \mathbf{a}. \tag{E4}$$

Setting the derivatives of $\boldsymbol{\epsilon}$ with respect to $\boldsymbol{\chi}$ equal zero, the following equation is obtained:

$$\mathbf{N}^{\mathrm{T}} \mathbf{N} \boldsymbol{\chi} = \mathbf{N}^{\mathrm{T}} \mathbf{a}. \tag{E5}$$

The unknown values of $\boldsymbol{\chi}$ are obtained by solving (E5). The matrix $\mathbf{N}^{\mathrm{T}}\mathbf{N}$ is similar to the finite element stiffness matrix and $\mathbf{N}^{\mathrm{T}}\mathbf{a}$ is similar to the finite element force vector.

E3 Enhanced Strain Field

The strain field calculated at the integration points is given by:

$$\bar{\boldsymbol{\varepsilon}}(\mathbf{x}) = \nabla_{\mathrm{s}} \bar{\mathbf{u}}(\mathbf{x}), \tag{E6}$$

where $\bar{\mathbf{u}}$ represents enriched finite element approximation for the displacement field and is given by (4.3) and the superscript s denotes the symmetric part of

the gradient operator. After Duflot and Bordas (2008) the enhanced strain field approximated at the element nodes is taken as:

$$\boldsymbol{\varepsilon}^{en}(\mathbf{x}) = \underbrace{\sum_{I\in\mathcal{N}} N_u^I(\mathbf{x})\mathbf{d}_I}_{Standard} + \underbrace{\sum_{J\in\mathcal{N}_{\mathrm{cr}}} N_u^J(\mathbf{x})\,\left[H\big(\phi(\mathbf{x})\big) - H\left(\phi(\mathbf{x}_J)\right)\right]\mathbf{e}_J}_{Heaviside} + \underbrace{\sum_{K\in\mathcal{N}_{\mathrm{tip}}} N_u^K(\mathbf{x})\sum_{\gamma=1}^{4}(\mathbf{x})\big[G_\gamma(\mathbf{x}) - G_\gamma(\mathbf{x}_K)\big]\,\mathbf{f}_{\gamma K}}_{Branch}, \tag{E7}$$

where $\mathbf{d}_I$, $\mathbf{e}_J$ and $\mathbf{f}_{\gamma K}$ are the coefficients that are to be evaluated to define the enhanced strain field. N_u^I, N_u^J and N_u^K are finite element shape functions, $\mathcal{N}$ is the set of all nodes in the mesh; $\mathcal{N}_{\mathrm{cr}}$ is the set of nodes whose support element is cut by the fracture; $\mathcal{N}_{\mathrm{tip}}$ is the set of nodes whose support element contains a fracture tip. G_γ is a component of the asymptotic near-tip strain function given by:

$$\mathbf{G} = \begin{bmatrix} \frac{1}{\sqrt{r}}\cos\left(\frac{\theta}{2}\right), \frac{1}{\sqrt{r}}\sin\left(\frac{\theta}{2}\right), \frac{1}{\sqrt{r}}\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\cos\left(\frac{3\theta}{2}\right), \\ \frac{1}{\sqrt{r}}\cos\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\sin\left(\frac{3\theta}{2}\right) \end{bmatrix} \tag{E8}$$

Following the procedure outlined in section E1 the system of equations obtained takes the form:

$$\begin{bmatrix} N_u^i N_u^I & N_u^i N_u^J h & N_u^i N_u^K g_L \\ N_u^j N_u^I h & N_u^j N_u^J & N_u^j h N_u^K g_L \\ N_u^k g_l N_u^I & N_u^k g_l N_u^J h & N_u^k g_l N_u^K g_L \end{bmatrix} \begin{pmatrix} \mathbf{d}_I \\ \mathbf{e}_J \\ \mathbf{f}_{\gamma K} \end{pmatrix} = \begin{pmatrix} N_u^i \bar{\boldsymbol{\varepsilon}} \\ N_u^j h \bar{\boldsymbol{\varepsilon}} \\ N_u^k g_l \bar{\boldsymbol{\varepsilon}} \end{pmatrix}, \tag{E9}$$

where $h = \left[H\big(\phi(\mathbf{x})\big) - H\left(\phi(\mathbf{x}_J)\right)\right]$ and $g_L = \left[G_\gamma(\mathbf{x}) - G_\gamma(\mathbf{x}_K)\right]$. The components $\mathbf{d}_I$, $\mathbf{e}_J$ and $\mathbf{f}_{\gamma K}$ are global strains recovered at the domain nodes.

E4 Numerical Example

This section presents a linear elastic fracture mechanics example to illustrate the performance of the strain recovery procedure using the enhanced strain field. The domain under consideration is an infinite plate containing a straight crack of length 2a and loaded by a remote stress field σ (Fig. E.1). In practical terms, "infinite" means that the width of the plate is ≫ 2a.

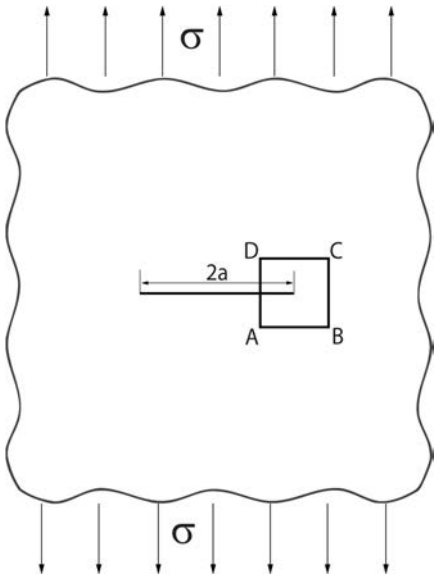


Fig. E.1. Infinite cracked plate loaded by remote tensile stress.

Along ABCD the closed form solution for mode I displacement in terms of polar coordinates centred at the crack tip is given by (Anderson, 2005):

$$u_x = \frac{K_I}{2G_s}\sqrt{\frac{r}{2\pi}}\cos\left(\frac{\theta}{2}\right)\left[\kappa - 1 + 2\sin^2\left(\frac{\theta}{2}\right)\right], \tag{E.10}$$

$$u_y = \frac{K_I}{2G_s}\sqrt{\frac{r}{2\pi}}\sin\left(\frac{\theta}{2}\right)\left[\kappa + 1 - 2\cos^2\left(\frac{\theta}{2}\right)\right], \tag{E.11}$$

where K_I is the mode I stress intensity factor given by $\sigma\sqrt{\pi a}$, G_s is the shear modulus given by $E/2(1+\nu)$ and κ is given by $(3-\nu)/(1+\nu)$ for the plane stress condition. The closed form strain is also given by:

$$\varepsilon_x = \frac{K_I}{8G_s}\sqrt{\frac{2}{r\pi}}\cos\left(\frac{\theta}{2}\right)\left[\kappa + 1 + 8\cos^4\left(\frac{\theta}{2}\right) - 10\cos^2\left(\frac{\theta}{2}\right)\right], \tag{E.12}$$

$$\varepsilon_y = -\frac{K_I}{8G_s}\sqrt{\frac{2}{r\pi}}\cos\left(\frac{\theta}{2}\right)\left[3 - \kappa + 8\cos^4\left(\frac{\theta}{2}\right) - 10\cos^2\left(\frac{\theta}{2}\right)\right], \tag{E.13}$$

$$\tau_{xy} = \frac{K_I}{4G_s}\sqrt{\frac{2}{r\pi}}\cos^2\left(\frac{\theta}{2}\right)\sin\left(\frac{\theta}{2}\right)\left[+4\cos^2\left(\frac{\theta}{2}\right) - 3\right]. \tag{E.14}$$

For this example ABCD is a square of 2mm x 2mm, $K_I = 1$, $E = 10e3\text{N/mm}^2$ and $\nu = 0.3$. The finite element mesh used for this example is shown in Fig. E.2.

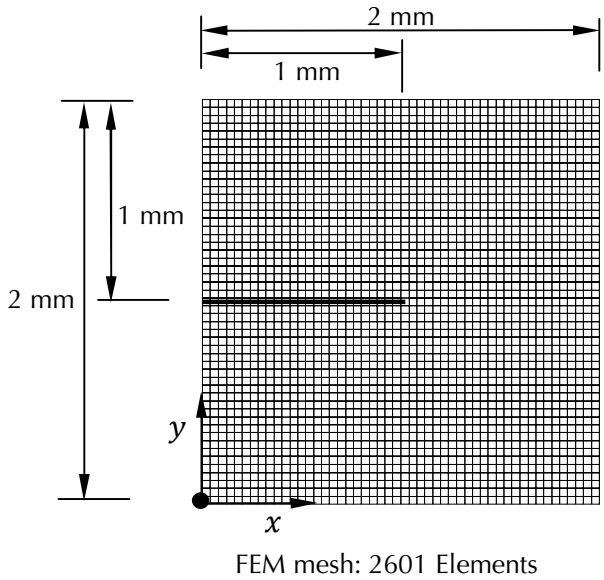


Fig. E.2. Finite element mesh used to model infinite plate with crack.

The strain fields obtained from the exact solution along with those obtained for the standard and enhanced global recovery procedure are shown in Fig. E.3.

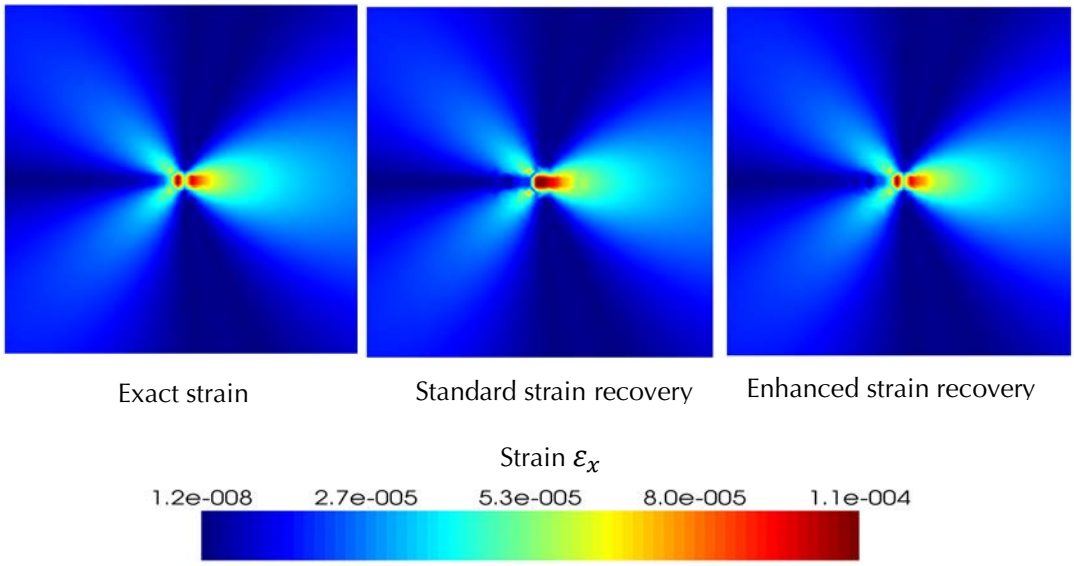


Fig. E.3. Strain fields, ε_x, obtained for infinite plate with crack.

Visually the strain fields appear very similar, with very small variations in the vicinity of the fracture tip. Plots of the strain along vertical sections through the domain are provided in Fig. E.4 and provide a comparison of the strain fields obtained from the standard and enhanced recovery procedures. The section at x=0.5, y, shows that while both recovery techniques are able to adequately determine the strain field away from the fracture, the standard recovery scheme overestimates the strain at the location of the fracture. This overestimation in the strain is also shown at the section taken at x=1, y, which passes through the fracture tip. The plots show that the incorporation of the enhanced strain field improves the approximation of the strain field. At section x=1.5, y, the fracture is not encountered and therefore both the standard and enhanced recovery procedure adequately approximate the strain.

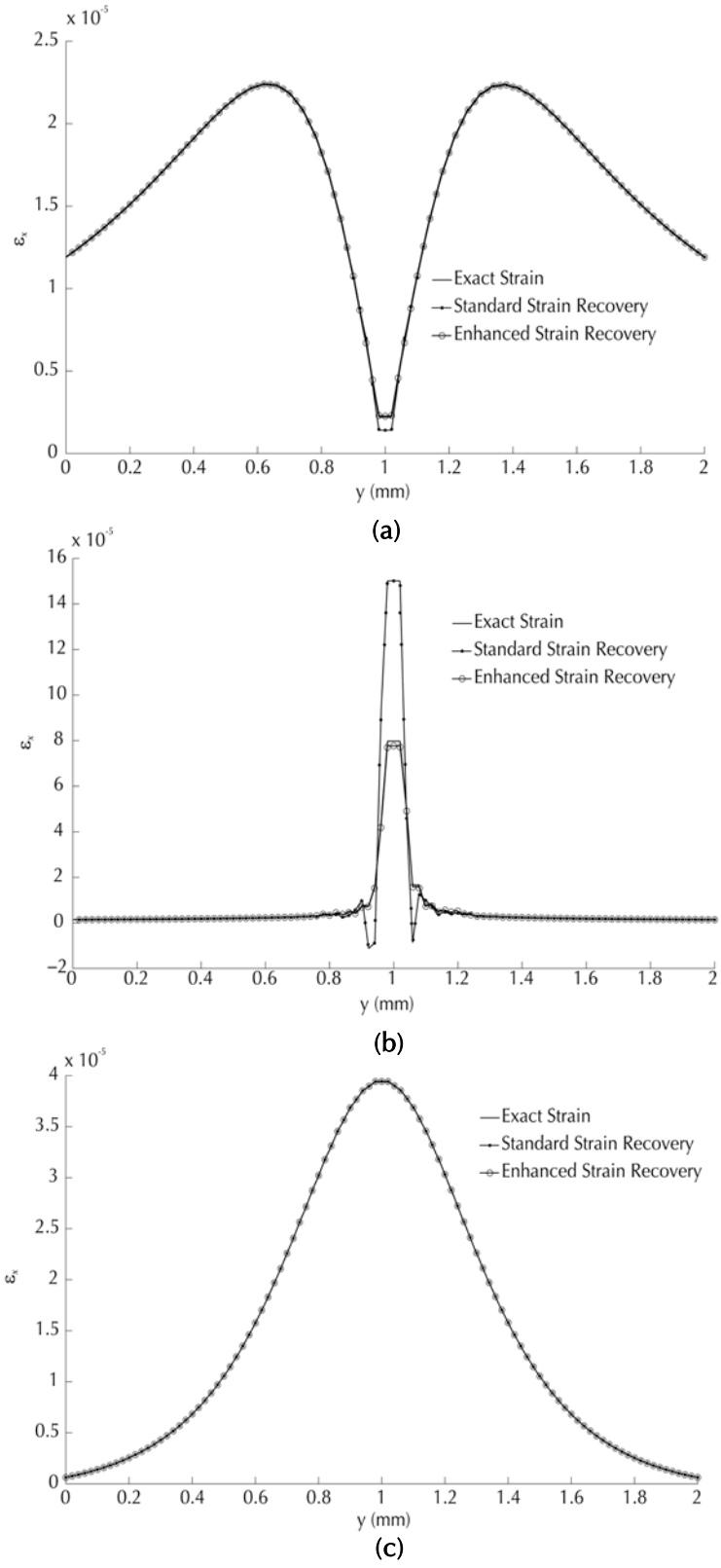


Fig. E.4. Comparison of exact strain to recovered strain, ε_x, at vertical sections taken at (a) x=0.5, y, (b) x=1.0, y and (c) x=1.5, y, through the domain.

References

Al-Khoury, R. & Sluys, L. J. 2006. A computational model for fracturing porous media. *International Journal for Numerical Methods in Engineering*, 423-444 DOI: 10.1002/nme.1886.

Anderson, T. L. 2005. *Fracture mechanics : Fundamentals and applications,* Boca Raton, FL, Taylor & Francis.

Atkinson, C. & Craster, R. V. 1991. Plane strain fracture in poroelastic media. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences,* 434**,** 605-633.

Babuška, I. 1971. Error-bounds for finite element method. *Numerische Mathematik,* 16**,** 322-333.

Babuška, I. 1973. The finite element method with lagrangian multipliers. *Numerische Mathematik,* 20**,** 179-192.

Bai, M., Meng, F., Elsworth, D., Zaman, M. & Roegiers, J. C. 1997. Numerical modeling of stress-dependent permeability. *International Journal of Rock Mechanics and Mining Sciences,* 34**,** 2.e1-2.e14 DOI: 10.1016/s1365-1609(97)00056-7.

Barenblatt, G. I., Zheltov, I. P. & Kochina, I. N. 1960. Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata]. *Journal of Applied Mathematics and Mechanics,* 24**,** 1286-1303.

Bear, J. 1988. *Dynamics of fluids in porous media,* New York, Dover.

Bechet, E., Minnebol, H., Moes, N. & Burgardt, B. 2005. Improved implementation and robustness study of the x-fem for stress analysis around cracks. *International Journal for Numerical Methods in Engineering,* 64**,** 1033-1056 DOI: 10.1002/nme.1386.

Becker, R. & Hansbo, P. 1999. A finite element method for domain decomposition with non-matching grids. *Technical Report No. 3613.* INRIA, Sophia Antipolis.

Belytschko, T. & Black, T. 1999. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering,* 45**,** 601-620.

Belytschko, T. & Fleming, M. 1999. Smoothing, enrichment and contact in the element-free galerkin method. *Computers & Structures,* 71**,** 173-195.

Belytschko, T., Gracie, R. & Ventura, G. 2009. A review of extended/generalised finite element methods for mechanics. *Modelling and Simulation in Materials Science and Engineering,* 17 DOI: 10.1088/0965-0393/17/4/043001.

Belytschko, T., Krongauz, Y., Organ, D., Fleming, M. & Krysl, P. 1996. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering,* 139**,** 3-47.

Belytschko, T., Lu, Y. Y. & Gu, L. 1994. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering,* 37**,** 229-256.

Belytschko, T., Moës, N., Usui, S. & Parimi, C. 2001. Arbitrary discontinuities in finite elements. *International Journal for Numerical Methods in Engineering,* 50**,** 993-1013 DOI: 10.1002/1097-0207(20010210)50:4<993::aid-nme164>3.0.co;2-m.

Berkowitz, B., Bear, J. & Braester, C. 1988. Continuum models for contaminant transport in fractured porous formations. *Water Resour. Res.,* 24**,** 1225-1236 DOI: 10.1029/WR024i008p01225.

Biot, M. A. 1941. General theory of three-dimensional consolidation. *Journal of Applied Physics,* 12**,** 155-164.

Biot, M. A. 1955. Theory of elasticity and consolidation for a porous anisotropic solid. *Journal of Applied Physics,* 26**,** 182-185 DOI: 10.1063/1.1721956.

Bockel-Rebelle, M. O., Vesseron, M. G., Jeanjean, F., Abdulla Khoori, S. & Rebelle, M. J. M. 2006. Polygonal faults/fractures pattern within abu dhabi onshore lower cretaceous platform carbonates. *GEO 2006 Middle East Conference and Exhibtion.* Manama, Bahrain.

Bordas, S., Rabczuk, T. & Zi, G. 2008. Three-dimensional crack initiation, propagation, branching and junction in non-linear materials by an extended meshfree method without asymptotic enrichment. *Engineering Fracture Mechanics,* 75**,** 943-960 DOI: 10.1016/j.engfracmech.2007.05.010.

Botros, F. E., Hassan, A. E., Reeves, D. M. & Pohll, G. 2008. On mapping fracture networks onto continuum. *Water Resour. Res.,* 44 DOI: 10.1029/2007wr006092.

Breitkopf, P., Rassineux, A., Touzot, G. & Villon, P. 2000. Explicit form and efficient computation of mls shape functions and their derivatives. *International Journal for Numerical Methods in Engineering,* 48**,** 451-466.

Brezzi, F. 1974. Existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Revue Francaise D Automatique Informatique Recherche Operationnelle,* 8**,** 129-151.

Budyn, E. 2004. *Multiple crack growth by the extended finite element method.* PhD., Northwestern University.

Budyn, E., Zi, G., Moës, N. & Belytschko, T. 2004. A method for multiple crack growth in brittle materials without remeshing. *International Journal for Numerical Methods in Engineering,* 61**,** 1741-1770.

Cartwright, J. A. 1994. Episodic basin-wide fluid expulsion from geopressured shale sequences in the north sea basin. *Geology,* 22**,** 447-450.

Cartwright, J. A. & Dewhurst, D. N. 1998. Layer-bound compaction faults in fine-grained sediments. *Bulletin of the Geological Society of America,* 110**,** 1242-1257.

Cartwright, J. A. & Lonergan, L. 1996. Volumetric contraction during the compaction of mudrocks: A mechanism for the development of regional-scale polygonal fault systems. *Basin Research,* 8**,** 183-193.

Chan, S. K., Tuba, I. S. & Wilson, W. K. 1970. On the finite element method in linear fracture mechanics. *Engineering Fracture Mechanics,* 2**,** 1-17 DOI: 10.1016/0013-7944(70)90026-3.

Chessa, J., Smolinski, P. & Belytschko, T. 2002. The extended finite element method (xfem) for solidification problems. *International Journal for Numerical Methods in Engineering,* 53**,** 1959-1977.

Clausen, J. A., Gabrielsen, R. H., Reksnes, P. A. & Nysæther, E. 1999. Development of intraformational (oligocene-miocene) faults in the northern north sea: Influence of remote stresses and doming of fennoscandia. *Journal of Structural Geology,* 21**,** 1457-1475 DOI: 10.1016/s0191-8141(99)00083-8.

Coussy, O. 2004. *Poromechanics,* Chichester, England ; Hoboken, NJ, Wiley.

Cryer, C. W. (ed.) 1963. *A comparison of the three-dimensional consolidation theories*: Oxford University Press.

Cundall, P. A. & Strack, O. D. L. 1979. A discrete numerical model for granular assemblies. *Geotechnique,* 29**,** 47-64.

Danielson, K. T., Hao, S., Liu, W. K., Aziz Uras, R. & Li, S. 2000. Parallel computation of meshless methods for explicit dynamic analysis. *International Journal for Numerical Methods in Engineering,* 47**,** 1323-1341.

Daux, C., Moes, N., Dolbow, J., Sukumar, N. & Belytschko, T. 2000. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering,* 48**,** 1741-1760.

Davis, R. O. & Selvadurai, A. P. S. 2002. *Plasticity and geomechanics*, Cambridge University Press.

De Borst, R., Réthoré, J. & Abellan, M.-A. 2006. A numerical approach for arbitrary cracks in a fluid-saturated medium. *Archive of Applied Mechanics,* 75**,** 595-606 DOI: 10.1007/s00419-006-0023-y.

Dewhurst, D. N., Cartwright, J. A. & Lonergan, L. 1999. Three-dimensional consolidation of fine-grained sediments. *Canadian Geotechnical Journal,* 36**,** 355-362.

Diodato, D. M. 1994. A compendium of fracture flow models. Work sponsored by U.S. Department of Defense, United States Army, Europe, Combat Maneuver Training Center, Hohenfels, Germany (Nov 1994).

Dolbow, J. 1999. *An extended finite element method with discontinuous enrichment for applied mechanics.* Doctoral PhD thesis, Northwestern University.

Dolbow, J., Moës, N. & Belytschko, T. 2000. Modeling fracture in mindlin-reissner plates with the extended finite element method. *International Journal of Solids and Structures,* 37**,** 7161-7183 DOI: 10.1016/s0020-7683(00)00194-3.

Duff, I. S. 2004. Ma57---a code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Softw.,* 30**,** 118-144 DOI: http://doi.acm.org/10.1145/992200.992202.

Duflot, M. 2007. A study of the representation of cracks with level sets. *International Journal for Numerical Methods in Engineering,* 70**,** 1261-1302.

Duflot, M. & Bordas, S. 2008. A posteriori error estimation for extended finite elements by an extended global recovery. *International Journal for Numerical Methods in Engineering,* 76**,** 1123-1138 DOI: 10.1002/nme.2332.

Erdogan, F. & Sih, G. C. 1963. On the crack extension in plates under plane loading and transverse shear. *Journal of Basic Engnieering, ASME,* 85**,** 519-527.

Fehl, B. D. & Truman, K. Z. 1999. An evaluation of fracture mechanics quarter-point displacement techniques used for computing stress intensity factors. *Engineering Structures,* 21**,** 406-415 DOI: 10.1016/s0141-0296(97)00221-6.

Fish, J. & Belytschko, T. 2007. *A first course in finite elements*, John Wiley & Sons.

Fleming, M., Chu, Y. A., Moran, B. & Belytschko, T. 1997. Enriched element-free galerkin methods for crack tip fields. *International Journal for Numerical Methods in Engineering,* 40**,** 1483-1504.

Fung, L. S. K. 1991. Simulation of block-to-block processes in naturally fractured reservoirs. *SPE Reservoir Engineering,* 6**,** 477-484 DOI: 10.2118/20019-pa.

Geiger-Boschung, S., Huangfu, Q., Reid, F., Matthai, S. K., Coumou, D., Belayneh, M., Fricke, C. & Schmid, K. S. 2009. Massively parallel sector scale discrete fracture and matrix simulations. *SPE Reservoir Simulation Symposium.* The Woodlands, Texas, USA, SPE 118924-MS.: Society of Petroleum Engineers.

Ghafouri, H. R. & Lewis, R. W. 1996. A finite element double porosity model for heterogeneous deformable porous media. *International Journal for Numerical And Analytical Methods in Geomechanics,* 20**,** 831-844.

Gibson, R. E., Knight, K. & Taylor, P. W. 1963. A critical experiment to examine theories of three-dimensional consolidation. *In:* European Conference on Soil Mechanics and Foundation Engineering. 69-76.

Golberg, M. 1996. Recent developments in the numerical evaluation of particular solutions in the boundary element method. *Applied Mathematics and Computation,* 75**,** 91-101 DOI: 10.1016/0096-3003(95)00123-9.

Goulty, N. R. 2001. Mechanics of layer-bound polygonal faulting in fine-grained sediments. *Journal of the Geological Society, London,* Vol. 159**,** 239-246.

Gravouil, A., Moës, N. & Belytschko, T. 2002. Non-planar 3d crack growth by the extended finite element and level sets—part ii: Level set update. *International Journal for Numerical Methods in Engineering,* 53**,** 2569-2586 DOI: 10.1002/nme.430.

Gringarten, A. C., Henry J. Ramey, J. & Raghavan, R. 1974. Unsteady-state pressure distributions created by a well with a single infinite-conductivity vertical fracture. *Society of Petroleum Engineers Journal,* 14**,** 347-360 DOI: 10.2118/4051-pa.

Henriet, J. P., De Batist, M. & Verschuren, M. 1991. Early fracturing of palaeogene clay, southermost north sea: Relevance to mechanisms of primary

hydrocarbon migration. *Special Publication of the European Association of Petroleum Geoscientists,* 1**,** 217-227.

Hollenbeck, K. J. 1998. *Invlap.M: A matlab function for numerical inversion of laplace transforms by the de hoog algorithm* [Online]. [Accessed December 01 2009].

Huang, R., Sukumar, N. & Prévost, J. H. 2003. Modeling quasi-static crack growth with the extended finite element method part ii: Numerical applications. *International Journal of Solids and Structures,* 40**,** 7539-7552.

Hughes, T. J. R. 2000. *The finite element method : Linear static and dynamic finite element analysis,* Mineola, NY, Dover Publications.

Hughes, T. J. R., Taylor, R. L., Sackman, J. L., Curnier, A. & Kanoknukulchai, W. 1976. A finite element method for a class of contact-impact problems. *Computer Methods in Applied Mechanics and Engineering,* 8**,** 249-276 DOI: 10.1016/0045-7825(76)90018-9.

Huyakorn, P. S., Lester, B. H. & Faust, C. R. 1983. Finite element techniques for modeling groundwater flow in fractured aquifers. *Water Resour. Res.,* 19 DOI: 10.1029/WR019i004p01019.

Ingraffea, A. R. & Wawrzynek, P. A. 2003. Finite element methods for linear elastic fracture mechanics. *In:* Milne, I., Ritchie, R. O. & Karihaloo, B. (eds.) *Comprehensive structural integrity.* Oxford: Pergamon.

Irwin, G. R. 1958. Fracture. *Handbuch der Physik,* IV.

Istok, J. D. 1989. *Groundwater modeling by the finite element method,* Washington, DC, American Geophysical Union.

Janssen, M., Zuidema, J. & Wanhill, R. J. H. 2004. *Fracture mechanics,* Abingdon, Oxfordshire, Spon Press.

Karimi-Fard, M., Durlofsky, L. J. & Aziz, K. 2004. An efficient discrete fracture model applicable for general-purpose reservoir simulators. *SPE Journal,* 9**,** 227-236. SPE 88812-PA DOI: 10.2118/88812-pa.

Kim, J. G. & Deo, M. D. 2000. Finite element, discrete fracture model for multiphase flow in porous media. *AIChE Journal,* 46**,** 1120-1130.

Knupp, P. M. & Salari, K. 2003. *Verification of computer codes in computational science and engineering,* Boca Raton, Fla., Chapman & Hall/CRC.

Laborde, P., Pommier, J., Renard, Y. & Salaun, M. 2005. High-order extended finite element method for cracked domains. *International Journal for Numerical Methods in Engineering,* 64**,** 354-381 DOI: 10.1002/nme.1370.

Lancaster, P. & Salkauskas, K. 1981. Surfaces generated by moving least squares methods. *Mathematics of Computation,* 37**,** 141-158.

Laursen, T. A. 2002. *Computational contact and impact mechanics : Fundamentals of modeling interfacial phenomena in nonlinear finite element analysis,* Berlin, Springer.

Lee, J., Choi, S.-U. & Cho, W. 1999. A comparative study of dual-porosity model and discrete fracture network model. *KSCE Journal of Civil Engineering,* 3**,** 171-180.

Lee, S. H., Lough, M. F. & Jensen, C. L. 2001. Hierarchical modeling of flow in naturally fractured formations with multiple length scales. *Water Resour. Res.,* 37 DOI: 10.1029/2000wr900340.

Lewis, R. W. & Shrefler, B. A. 1998. *The finite element method in the static and dynamic deformation and consolidation of porous media,* Chichester ; New York, John Wiley.

Liu, G. R. 2002. *Mesh-free methods: Moving beyond the finite element methods* CRC Press Inc.

Liu, G. R. & Gu, Y. T. 2001. A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering,* 50**,** 937-951.

Liu, G. R. & Gu, Y. T. 2005. *An introduction to meshfree methods and their programming,* Springer.

Liu, G. R., Zhang, G. Y., Wang, Y. Y., Zhong, Z. H., Li, G. Y. & Han, X. 2007. A nodal integration technique for meshfree radial point interpolation method (ni-rpim). *International Journal of Solids and Structures,* 44**,** 3840-3860 DOI: 10.1016/j.ijsolstr.2006.10.025.

Lu, Y. Y., Belytschko, T. & Gu, L. 1994. A new implementation of the element-free galerkin method. *Computational Methods in Applied Mechanics and Engineering,* 113**,** 397-414.

Mandel, J. 1953. Consolidation des sols (étude mathématique). *Geotechnique,* 3**,** 287–299 DOI: 10.1680/geot.1953.3.7.287.

Matthäi, S. K., Mezentsev, A. A. & Belayneh, M. 2007. Finite element - node-centered finite-volume two-phase-flow experiments with fractured rock represented by unstructured hybrid-element meshes. *Spe Reservoir Evaluation & Engineering,* 10**,** pp. 740-756. SPE 93341-PA DOI: DOI: 10.2118/93341-pa.

Mckenna, S. A. & Reeves, P. C. 2006. Fractured continuum approach to stochastic permeability modeling. *In:* Yarus, J. M. & Chambers, R. L. (eds.) *Stochastic modeling and geostatistics: Principles, methods and case studies, volume ii.* AAPG Computer Applications in Geology 5 ed.

Melenk, J. M. & Babuska, I. 1996. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering,* 139**,** 289-314.

Minkoff, S. E., Stone, C. M., Bryant, S., Peszynska, M. & Wheeler, M. F. 2003. Coupled fluid flow and geomechanical deformation modeling. *Journal of Petroleum Science and Engineering,* 38**,** 37-56 DOI: 10.1016/S0920-4105(03)00021-4.

Moen-Maurel, L., Jeanjean, F., Francois, G. & Sabah, A. K. 2007. *Polygonal faulting/fracturing: Characterization (seismic, well data) from a carbonate field, abu dhabi* [Online]. Available: http://www.searchanddiscovery.net/documents

/2007/07018annual_abs_lngbch/abstracts/lbMoen.htm. [Accessed November 1 2007].

Moës, N., Dolbow, J. & Belytschko, T. 1999. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering,* 46**,** 131-150.

Moës, N., Gravouil, A. & Belytschko, T. 2002. Non-planar 3d crack growth by the extended finite element and level sets - part ι: Mechanical model. *International Journal for Numerical Methods in Engineering,* 53**,** 2549-2568.

Nagashima, T., Omoto, Y. & Tani, S. 2003. Stress intensity factor analysis of interface cracks using x-fem. *International Journal for Numerical Methods in Engineering,* 56**,** 1151-1173 DOI: 10.1002/nme.604.

Nguyen, V. P., Rabczuk, T., Bordas, S. & Duflot, M. 2008. Meshless methods: A review and computer implementation aspects. *Mathematics and Computers in Simulation,* 79**,** 763-813 DOI: 10.1016/j.matcom.2008.01.003.

Nitsche, J. 1970. Über ein variationsprinzip für die lösung von dirichlet problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen in der Mathematik an der Universität Hamburg,* 36**,** 9-15.

Oliver, J. 1995. Continuum modelling of strong discontinuities in solid mechanics using damage models. *Computational Mechanics,* 17**,** 49-61.

Onate, E., Idelsohn, S., Zienkiewicz, O. C. & Taylor, R. L. 1996. A finite point method in computational mechanic—applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering,* 39**,** 3839-3866.

Osher, S. & Sethian, J. A. 1988. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics,* 79**,** 12-49.

Paluszny, A. & Matthai, S. K. 2008. Numerical simulation of geomechanic fracture networks: Application to measuring effective permeability variability. *The 42nd U.S. Rock Mechanics Symposium (USRMS).* San Francisco, CA: American Rock Mechanics Association.

Papadopoulos, P. & Solberg, J. M. 1998. A lagrange multiplier method for the finite element solution of frictionless contact problems. *Mathematical and Computer Modelling,* 28**,** 373-384 DOI: 10.1016/s0895-7177(98)00128-9.

Persson, P.-O. & Strang, G. 2004. A simple mesh generator in matlab. *SIAM Review,* 46**,** 329-345 DOI: 10.1137/s0036144503429121.

Potts, D. M. & Zdravkovic, L. 1999. *Finite element analysis in geotechnical engineering : Theory,* London Reston, VA, Telford ; Distributed by ASCE Press.

Prabel, B., Combescure, A., Gravouil, A. & Marie, S. 2007. Level set x-fem non-matching meshes: Application to dynamic crack propagation in elastic-plastic media. *International Journal for Numerical Methods in Engineering,* 69**,** 1553-1569.

Rabczuk, T. & Belytschko, T. 2004. Cracking particles: A simplified meshfree method for arbitrary evolving cracks. *International Journal for Numerical Methods in Engineering,* 61**,** 2316-2343 DOI: 10.1002/nme.1151.

Rashid, M. M. 1998. The arbitrary local mesh refinement method: An alternative to remeshing for crack propagation analysis. *Computer Methods in Applied Mechanics and Engineering,* 154**,** 133-150.

Renshaw, C. E. & Pollard, D. D. 1994. Numerical simulation of fracture set formation: A fracture mechanics model consistent with experimental observations. *J. Geophys. Res.,* 99**,** 9359-9372 DOI: 10.1029/94jb00139.

Rice, J. R. 1968. A path independent integral and the approximate analysis of stress concentration by notches and cracks. *Journal of Applied Mechanics,* 35**,** 379-386.

Rycroft, C. H., Grest, G. S., Landry, J. W. & Bazant, M. Z. 2006. Analysis of granular flow in a pebble-bed nuclear reactor. *Physical Review E,* 74**,** 021306.

Samardzioska, T. & Popov, V. 2005. Numerical comparison of the equivalent continuum, non-homogeneous and dual porosity models for flow and transport in fractured porous media. *Advances in Water Resources,* 28**,** 235-255 DOI: 10.1016/j.advwatres.2004.11.002.

Smith, I. M. & Griffiths, D. V. 2004. *Programming the finite element method,* Hoboken, NJ, Wiley.

Stazi, F. L., Budyn, E., Chessa, J. & Belytschko, T. 2003. An extended finite element method with higher-order elements for curved cracks. *Computational Mechanics,* 31**,** 38-48 DOI: 10.1007/s00466-002-0391-2.

Stolarska, M., Chopp, D. L., Moës, N. & Belytschko, T. 2001. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering,* 51**,** 943-960.

Sukumar, N., Chopp, D. L., Moës, N. & Belytschko, T. 2001. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering,* 190**,** 6183-6200.

Sukumar, N., Chopp, D. L. & Moran, B. 2003. Extended finite element method and fast marching method for three-dimensional fatigue crack propagation. *Engineering Fracture Mechanics,* 70**,** 29-48 DOI: 10.1016/s0013-7944(02)00032-2.

Sukumar, N., Moës, N., Moran, B. & Belytschko, T. 2000. Extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering,* 48**,** 1549-1570.

Sukumar, N. & Prévost, J. H. 2003. Modeling quasi-static crack growth with the extended finite element method part i: Computer implementation. *International Journal of Solids and Structures,* 40**,** 7513-7537.

Svensson, U. 2001a. A continuum representation of fracture networks. Part i: Method and basic test cases. *Journal of Hydrology,* 250**,** 170-186 DOI: 10.1016/s0022-1694(01)00435-8.

Svensson, U. 2001b. A continuum representation of fracture networks. Part ii: Application to the äspö hard rock laboratory. *Journal of Hydrology,* 250**,** 187-205 DOI: 10.1016/s0022-1694(01)00436-x.

Terzaghi, K. 1925. *Earthwork mechanics on the basis of groundphysics (erdbaumechanik, in german),* Leipzig, Germany, Franz Deuticke.

Tran, N. H. & Ravoof, A. 2007. Coupled fluid flow through discrete fracture network: A novel approach. *International Journal of Mathematics and Computers in Simulation,* 1**,** 295-299.

Ventura, G., Budyn, E. & Belytschko, T. 2003. Vector level sets for description of propagating cracks in finite elements. *International Journal for Numerical Methods in Engineering,* 58**,** 1571-1592.

Ventura, G., Xu, J. X. & Belytschko, T. 2002. A vector level set method and new discontinuity approximations for crack growth by efg. *International Journal for Numerical Methods in Engineering,* 54**,** 923-944 DOI: 10.1002/nme.471.

Wagner, G. J. & Liu, W. K. 2000. Application of essential boundary conditions in mesh-free methods: A corrected collocation method. *International Journal for Numerical Methods in Engineering,* 47**,** 1367-1379.

Wang, J. G. & Liu, G. R. 2002. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering,* 54**,** 1623-1648.

Warren, J. E. & Root, P. J. 1963. The behavior of naturally fractured reservoirs. *SPE Journal,* 3**,** 245 - 255. SPE 426-PA DOI: 10.2118/426-pa.

Wriggers, P. 2006. *Computational contact mechanics,* Berlin ; New York, Springer.

Yau, J. F., Wang, S. S. & Corten, H. T. 1980. A mixed-mode crack analysis of isotropic solids using conservation laws of elasticity. *Journal of Applied Mechanics,* 47**,** 335-341 DOI: 10.1115/1.3153665.

Zhang, X. & Sanderson, D. J. 2002. *Numerical modelling and analysis of fluid flow and deformation of fractured rock masses,* Oxford, Elsevier Science Ltd.

Zhong, Z.-H. 1993. *Finite element procedures for contact-impact problems,* Oxford ; New York, Oxford University Press.

Zi, G., Rabczuk, T. & Wall, W. 2007. Extended meshfree methods without branch enrichment for cohesive cracks. *Computational Mechanics,* 40**,** 367-382.

Zienkiewicz, O. C. & Taylor, R. L. 2005. *The finite element method for solid and structural mechanics,* Amsterdam ; Boston, Elsevier Butterworth-Heinemann.

Zienkiewicz, O. C., Taylor, R. L. & Too, J. M. 1971. Reduced integration technique in general analysis of plates and shells. *International Journal for Numerical Methods in Engineering,* 3**,** 275-290 DOI: 10.1002/nme.1620030211.

Zienkiewicz, O. C., Taylor, R. L. & Zhu, J. Z. 2005. *The finite element method - its basis and fundementals*, Elsevier Butterworth-Heinemann.