Meshless methods: theory and application in 3D fracture modelling with level sets

XIAOYING ZHUANG

Meshless methods: theory and application in 3D fracture modelling with level sets

Xiaoying Zhuang

A Thesis Submitted to the Faculty of Science for the degree

of Doctor of Philosophy

Mechanics Group
School of Engineering and Computing Sciences
University of Durham
United Kingdom

October 2010

Meshless methods: theory and application in 3D fracture modelling with level sets

Xiaoying Zhuang

Abstract

Accurate analysis of fracture is of vital importance yet methods for effetive 3D calculations are currently unsatisfactory. In this thesis, novel numerical techniques are developed which solve many of these problems. This thesis consists two major parts: firstly an investigation into the theory of meshless methods and secondly an innovative numerical framework for 3D fracture modelling using the element-free Galerkin method and the level set method. The former contributes to some fundamental issues related to accuracy and error control in meshless methods needing to be addressed for fracture modelling developed later namely, the modified weak form for imposition of essential boundary conditions, the use of orthogonal basis functions to obtain shape functions and error control in adaptive analysis. In the latter part, a simple and efficient numerical framework is developed to overcome the difficulties in current 3D fracture modelling. Modelling cracks in 3D remains a challenging topic in computational solid mechanics since the geometry of the crack surfaces can be difficult to describe unlike the case in 2D where cracks can be represented as combinations of lines or curves. Secondly, crack evolution requires numerical methods that can accommodate the moving geometry and a geometry description that maintains accuracy in successive computational steps. To overcome these problems, the level set method, a powerful numerical method for describing and tracking arbitrary motion of interfaces, is used to describe and capture the crack geometry and forms a local curvilinear coordinate system around the crack front. The geometry information is used in the stress analysis taken by the element-Free Galerkin method as well as in the computation of fracture parameters needed for crack propagation. Examples are tested and studied throughout the thesis addressing each of the above described issues.

Declaration

The work in this thesis is based on research carried out in the Mechanics Group in the School of Engineering & Computing Sciences, Durham University, UK. No part of this report has been submitted elsewhere for any other degree or qualification and it all my own work unless referenced to the contrary in the text.

Acknowledgements

This thesis is based on the research project supported by a Dorothy Hodgkin Postgraduate Award from the UK Engineering and Physical Sciences Research Council [1]. The author gratefully acknowledge their support.

Thanks to my supervisor Dr Charles Augarde. He is THE best teacher I have ever found. He is always supportive, positive and gives helpful advices during my PhD. I gratefully thank him for his sustained support and care to me over the past three years to make this thesis possible. What I have learned from him is beyond the research skills or management, but the passion, attitude and perseverance in doing research. Whenever things did not go smooth as expected, he guided me through, encouraged me and showed great patience and understanding that has become my momentum to keep on going. He is also the person who makes my PhD opportuniy at Durham possible. I first met him in March of 2005 at the China-UK Seminar on Geotechnical Engineering held at Tongji University, Shanghai, China. I did a presentation in the seminar as MSc student from Tongji on slope stability analysis using a meshless method. The presentation interested Charles and we had a discussion during the break of the seminar. One year after the meeting, I decided to do my PhD with him. However, funding for PhD is always difficult in the UK especially for an overseas student. Charles spared no efforts in helping me apply for the scholarship with the earnest support from Professor Roger Crouch who later became my second supervisor. I honestly think the moment I was informed of the grant of the scholarship was one the most unforgettable times in my life. Here I would like to convey my appreciation to them both.

[1] Reference number NH/MAC

Thanks to the respectable forerunners, Professor Jon Trevelyan in my group in Durham and Professor Stéphane Bordas from Cardiff University, for their advices and lending hands to my research. Thanks to my colleagues who make me feel in a happy and sweet family. Thanks to our mathematician Dr Claire Heaney. She gave her expertise when I was overwhelmed by the dyadic production in 3D. Dr (was still "Mr" then) Robert Simpson provided me many suggestions and references on fracture mechanics. He also encouraged me during my thesis writing. Talking to him about coding is always exciting. Thanks to Mr Will Coombs and Mr Gareth Bird for their tips on Matlab and LaTeX. Thanks to Mr Zahur Ullah for sharing references and interesting discussions on meshless methods.

Taking time for hobbies outside research helps to refresh oneself both mentally and physically. Listening to classic music at leisure time and evening jogging becomes part of my balanced life style. Thanks to Langlang (a well known Chinese pianist) for his amazing music that gives me lots of inspiration and enthusiasm. Though he does not know me and is also unlikely to read this thesis, I would like to thank him and hope he will continuously bring his new works to us in the future.

Last but not the least, I gratefully acknowledge my family. Though I am miles and miles away from them but I can always feel the love, encouragement and care from them deep inside. I owe them too much in the past three years for too little time being spent with them. This thesis is by all means devoted to them because they are the reasons I am here and make what I am doing meaningful. Living far away from home is not easy however such experience helps me to grow and learn to be independent. Moreover, I now become a good chef (self-esteemed) not only good at cooking Chinese dishes but also making pastries!

By this moment I submit this thesis, I thought of all the people who have given helps to me. I am not allowed to list their names all here but they are appreciated. I would like to close the acknowledgment by quoting a saying of Oliver Wendell Holmes that has for countless times given me strength and melt the sorrow away

"*What lies behind us and what lies before us, are tiny matters, compared with what lies within us.*"

Publications and presentations

Aspects of the work presented in this thesis have been published, submitted or presented in the following journal papers and conferences:

1. Xiaoying Zhuang and Charles Augarde. Aspects of the use of orthogonal basis functions in the element free Galerkin method. *International Journal for Numerical Methods in Engineering*, 81: 366-380, 2010.

2. Xiaoying Zhuang, Charles Augarde and Stéphane Bordas. Accurate fracture modelling using meshless methods and level sets: formulation and 2D modelling. *International Journal for Numerical Methods in Engineering*, 2010, accepted for publication.

3. Yongchang Cai, Xiaoying Zhuang and Charels Augarde. A new partition of unity finite element free from linear dependence problem and processing the delta property. *Computer Methods in Applied Mechanics and Engineering*, 199: 1036-1043, 2010.

4. Xiaoying Zhuang, Claire Heaney and Charles Augarde. On error estimation in the element-free Galerkin method. Engineering Analysis with Boundary Elements, 2010 (under review).

5. Xiaoying Zhuang and Charles Augarde. The use of the level set coordinates to introduce displacement jumps for arbitrary 3D crack surface modelling. *World Congress on Computational Mechanics*, WCCM, Sydney, Australia, 2010.

6. Xiaoying Zhuang and Charles Augarde. A tying procedure for crack tip closure in fracture modelling using a meshless method and level sets, Proc. *18th annual*

conference of Association of Computational Mechanics in Engineering, ACME, Southampton, UK, 2010.

7. Xiaoying Zhuang and Charles Augarde. 2009. The effects of nodal arrangement on solution accuracy in meshless methods. *17th annual conference of Association of Computational Mechanics in Engineering*, ACME 2009, Nottingham, UK, 2009.

Nomenclature

E	Young's Modulus
$E*$	Modified Young's Modulus
ν	Poisson's ratio
κ	Kolosov constant
μ	Shear modulus
β	Penalty
W	Strain energy density
u_i	Displacement vector component
n_j	Normal vector component
δ_{ij}	Kronecker delta function
$\boldsymbol{u}$	Displacement vector
$\hat{\boldsymbol{t}}$	Auxiliary traction vector
σ_{ij}	Stress tensor (index notation)
$\boldsymbol{\sigma}$	Stress tensor (matrix vector notation)
ε_{ij}	Strain tensor (index notation)
$\boldsymbol{\varepsilon}$	Strain tensor (matrix vector notation)
b_j	Body force vector (index notation)

$\boldsymbol{b}$	Body force vector (matrix vector notation)
P_{ij}	Eshelby momentum tensor
$\mathbf{n}$	Normal vector
$\bar{u}_i$	Prescribed displacement along essential boundary
$\hat{\boldsymbol{u}}$	Auxiliary displacement vector
$\bar{t}_i$	Prescribed traction along traction boundary
$\boldsymbol{a}(\mathbf{x})$	Coefficients of basis function
$\boldsymbol{p}(\mathbf{x})$	Basis function
$H(\mathbf{x})$	Heaviside step function
$N(\mathbf{x})$	Shape function
$u^h(\mathbf{x})$	Approximated field
$w_I(\mathbf{x})$	Weight function
Γ	Problem domain boundary
γ	Surface energy density
Γ_t	Traction boundary
Γ_u	Essential boundary
Γ_{cr}	Crack front
Λ	Integration path
$\mathbf{e}_1$	Unit base vector in x_1 direction
$\mathbf{e}_2$	Unit base vector in x_2 direction
$\mathbf{e}_3$	Unit base vector in x_3 direction
Ω	Problem domain

Π	Potential energy
θ	Angle with respect to a local coordinate of a source point
a	Crack length
d_I	Distance from node I to a point of interest
d_{mI}	Support size of node I
G	Energy release rate
G_{III}	Energy release rate in mode III crack
I	Interaction domain integral
J_k	J_k integral
r	Distance to a source point
U_E	Elastic strain energy
W	Strain energy
W_{cr}	Surface energy
K_{I}	Mode I stress intensity factor
K_{II}	Mode II stress intensity factor
K_{III}	Mode III stress intensity factor
t	Unit time step
τ	Pseudo-time
ϕ	Normal level set to crack surface
ψ	Tangent level set to crack surface
V_ϕ	Velocity of normal level set
V_ψ	Velocity of tangent level set

Acronyms

BEM	Boundary element method
CFD	Computational fluid dynamics
DEM	Diffuse element method
EFGM	Element-free Galerkin method
FDM	Finite difference method
FEM	Finite element method
LEFM	Linear elastic fracture mechanics
LSC	Level set coordinates
LSM	Level set method
MLPG	Meshless local Petrov-Galerkin
MLS	Moving least squares
PU	Partition of unity
RKPM	Reproducing kernel particle method
SCF	Stress concentration factor
SIF	Stress intensity factor
SPH	Smoothed particle hydrodynamics
WRM	Weighted residual method
XFEM	Extended finite element method
TVD RK	Total variation diminishing Runge-Kutta
WENO	Weighted essential non-oscillatory

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Most physical phenomena can be described by one or a set of ordinary or partial differential equations (PDEs), or integral equations, which are the mathematical description of the underlying mechanical principles. Due to mathematical difficulties, analytical solutions to these equations exist only for limited cases of simple geometry and boundary conditions; numerical solutions are required for more general situations. The core issue in computational mechanics is the development of numerical methods for finding approximate solutions to these equations. Researchers in this field follow four stages to develop or to use a numerical method to solve a problem:

- Find a mathematical model describing the underlying mechanical principles of the physical phenomenon.
- Discretisation: converting the mathematical model into system equations.
- Code a computer program to solve the discretised equations.
- Verifying and validating solutions with experimental data or reference answers.

These four steps form a complete process from defining a mechanical problem to solving it. One step after another is influenced by the previous, e.g. the algorithms in a computer code have to be developed in the context of the method used to discretise the model. In the of field computational mechanics, efforts are mostly

devoted to the latter three parts while the mathematical model is of more concern to researchers in theoretical mechanics.

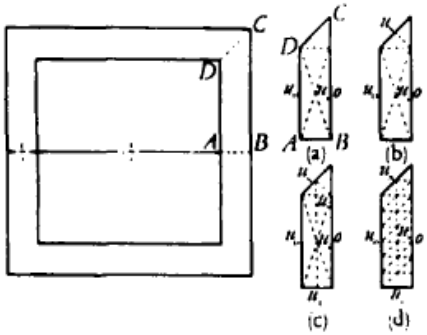


Figure 1.1: Triangular net used in Courant's 1943 paper [1] for solving torsion problem.

In the past century a wide variety of numerical methods have been developed to solve PDEs governing mechanical behaviour in a discretised way. The most well established methods are the finite difference method (FDM), the finite volume method, the finite element method (FEM), the boundary element method (BEM), the discrete element method (DEM) and meshless methods. Among these, the FEM is currently the most widely used method in practice. The idea of "finite elements" can be traced to 1943 when Courant [1] used local interpolation over a "triangular net", as shown in Figure 1.1 to solve a torsion problem. However the invention of the FEM as we know it today arose from the need to analyze nonrectangular structures as parts of aircraft. Before the FEM was developed, structural analysis was limited to elements connected with only two nodes by Hrennikoff's method. In the 1950s, a project in the Boeing aircraft company required the analysis of nonrectangular structures being part of swept-back box wings [8]. At that time structural analysis was restricted to rectangular areas using elements connected to only two points in space [8]. Clough joined the project and solved this problem in 1954. He then published his work with Turner in 1956 which is now regarded as the landmark paper [9] for the FEM in structural analysis. The name "finite element method" was coined by Clough at UC Berkeley during the 1960s. It has been over half a century since then and the notions of the stiffness matrix and element assembly are now routinely taught to engineering undergraduates. Today the FEM has become the

most ubiquitous analysis tool in analyzing engineering problems, e.g. solids, fluids, electronics, magnetic or multiphysical problems in fields such as civil engineering, aeronautical engineering, biomechanical engineering and automotive industries [10]. Despite the successes, the FEM is being challenged by a series of new problems with which it is incapable of or for which it finds difficulties in dealing with, such as crack propagation, material damage and failure, projectile and penetration and large deformation problems. The key issue is the existence of a mesh that conflicts with the real physical compatibility condition possessed by a continuum and which hinders the computation. As a consequence, remeshing is needed during successive analysis steps. Remeshing in 3D is not a time-bounded process and normally involves manual interaction even using the most sophisticated mesh generator [11]. The most daunting task is the mapping of the field solution from a previous mesh to a new mesh which often leads to the degradation of accuracy due to accumulated errors in mapping. To fix this problem, early efforts were made to develop automatic remeshing techniques [12], an area which is still of research interest today [13, 14]. On the other hand, researchers have also started to develop new methods which can get rid of the mesh and hence the entanglement of element topology. The pressing need to analyze more advanced and complicated problems in engineering practice and technology has propelled the development of new kinds of numerical methods. The goal of the work in this thesis is the development of methods to enable fast and accurate modelling of a particular set of challenging engineering problems involving fracture. This topic will be addressed in more detail later in this chapter.

1.2 Meshless methods

Before considering meshless methods in detail, a fundamental question is “what is a meshless method?” and “how do we distinguish it from a mesh-based method?”. Nowadays people perhaps do not tend to emphasize too much whether a given method is a mesh-based or a meshless method since nodal connectivity is needed anyway, if not in the preprocessing then in the realtime computation. However it is still of research interest to define a meshless method. Meshless methods (sometimes

called meshfree methods), as indicated by its name, requires no mesh to discretise a domain. Ever since the beginning of their development, meshless methods have been aimed towards problems of changing geometry where a mesh-based method might struggle or not converge due to mesh distortion, or due to mesh sensitivity such as in crack propagation, strain localization and large deformation problems. To define what a meshless method is, one can go to the literature. A numerical method can be regarded as meshless if it satisfies this requirement:

- There is no need for a predefined mesh in constructing the field approximation.

So, meshless methods form a class of numerical methods for solving PDEs in a discrete way where only nodal information is required. An ideal meshless method would require no mesh throughout the analysis. However most current meshless methods rarely meet this requirement perfectly. Some meshless methods need to use a mesh for integration purposes before or during analysis, while others do not require a mesh but suffer from unstable and degrading accuracies. Meshless methods developed so far fall into one of the following categories:

- The method requires a predefined background mesh for integration.
- The method does not require a predefined background mesh but requires realtime local cells for integration.
- The method does not a need mesh for integration at all.

In terms of solution stability and accuracy, methods in the first category generally fare better than the latter two with the second better than the third. So generally the less the "mesh" requirement, the lower the solution accuracy.

1.2.1 Development and classification

The origin of meshless methods dates from the 1970s with the smoothed partial hydrodynamics (SPH) method proposed by Lucy [15] and by Gingold and Monaghan [16] respectively at almost the same time. The SPH is a meshless particle method based on a Lagrangian formulation where a set of moving particles represent a

continuum where each particle represents a predefined mass or volume. It was a breakthrough for calculations in fluid mechanics based on a grid or mesh. Monaghan [17] conducted a detailed study of the SPH and summarised it as a "kernel estimate integral" for the governing PDEs. The SPH method can be used even the problem domain undergoes extremely large deformation or separation. It has been applied to simulate shock problems [18], computational fluid hydrodynamics [19], astrophysical problems e.g. the formation of protostars and galaxies [15, 20], underwater explosion [21], impact problems [22] and metal forming [23]. Despite its successful application, the SPH method suffers from a number of problems such as as a tensile instability problem, inconsistency of approximation, difficulty in imposing essential boundary conditions and in choosing artificial viscosity. Modifications to the SPH method to overcome these shortcomings have been proposed in more recent works [22, 24, 25]. Detailed reviews of the history and recent development of the SPH method are presented in [26].

It has been over thirty years since the inception of the SPH, however it was not until the early 1990s that meshless methods started to hold wide attention from researchers in computational mechanics and began to make significant advances in solving engineering problems. In 1981 Lancaster and Salkauskas [27] proposed a moving least square (MLS) approximation to construct a smooth curve or surface fitting to a set of scattered nodal data. Before the 1990s, the MLS approximation was limited to topology optimization problems. Ten years later Nayroles et al. [28, 29] developed the MLS approximation into a field approximation method and named it as the diffuse element method (DEM). Based on this idea, in 1994 Belytschko et al. [3] published a landmark paper where the MLS approximation was incorporated into a Galerkin weak form for solving solid mechanics problem. They named this method the element-free Galerkin method (EFGM) and it is this method that is currently the most widely used meshless method. A wide class of other meshless methods have been proposed in the past intervening 15 years and these will be reviewed below.

One can classify meshless methods in various senses, namely according to the mesh dependence (whether a mesh is needed for integration or not), or by type of approximation. Here we introduce meshless methods chronologically and classify

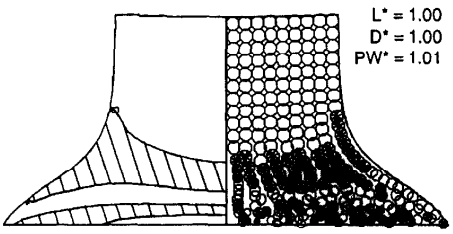


Figure 1.2: Impact simulation of a cylinder impacting fluid by the SPH (taken from [2]).

their type of approximation. Liu et al. [30] developed the reproducing kernel particle method (RKPM) as a claimed improvement to the SPH method and EFGM and introduced the use of wavelet functions. Oñate et al. [31] introduced the finite point method where the shape functions constructed by the MLS approximation are directly fitted into the governing equations. There is no need for a mesh either for field approximation or integration. Melenk and Babuška [32] proposed the concept of partition of unity (PU) for constructing the approximation under which the FE interpolation and meshless approximation can be generalized as special cases of PU [33]. Based on local Petrov-Galerkin weak forms and MLS approximation, a family of meshless local Petrov-Galerkin methods (MLPG) [34–37] has also been proposed. The natural element method (NEM) [38], the natural neighbour method (NNM) [39] and the meshless finite element method (MFEM) [11] are based on Sibsonion interpolations and were introduced by Sukumar. The point interpolation method (PIM) [40–43] is based on point local interpolation similar to the FEM and an advantage is that the essential boundary conditions can be directly applied. Duarte and Oden [44] [45] and Oden et al. [46] proposed the *h-p* cloud FEM based on PU and MLS approximation. Meshless methods described above have shown their advantages over the FEM in the following senses:

- Only nodal information is required, therefore there is less time and effort in meshing which is a major overhead in the FEM.

- They can easily adapt to changes in geometry of the continuum since nodal connectivity is part of the computation. Therefore they are particularly suitable to deal with problems of moving boundaries.

- They provide a field approximation which is smooth and of high order differentiability.
- The stress field obtained is smooth and does not require an additional smoothing technique throughout the domain unlike the FEM in general.

These advantages enable meshless methods to move beyond the applications covered by the conventional FEM in dealing with moving boundaries and changing geometries. They have shown feasibility and advantages in dealing with problems such as

- Impact computation, e.g penetration and target fragmentation [2, 18, 22].
- Moving boundary problems, e.g. free surface flow [47], phase change in heat transfer and crystal growth, crack propagation [48, 49].
- Strain localization and dynamic shear band propagation problems [50–54].
- Large deformation problems such analysis of shells, plates and thin structures [55–59] and metal forming [23, 60, 61].

1.2.2 Remaining issues

Despite these successful applications, some issues remain to be addressed before a meshless method can be widely and robustly applied in substantial applications. The major issues are

- Computational cost and complexity in obtaining the shape functions.
- The imposition of boundary conditions and compatibility conditions.
- Integration issues, in terms of convergence and efficiency.
- Efficient algorithms in finding realtime nodal connectivity for parallel computation.
- Convergence and error control in adaptive analysis.

In terms of the computational cost in obtaining shape functions, meshless methods are generally more expensive than the FEM. So it is commonly believed that meshless methods are not as efficient as the FEM. However Trobec et al. [62] compared the total computational complexity (including preprocessing) of the MLPG method (a meshless method using a local weak form instead of the global weak form in the EFGM) with the FEM and showed that, with properly designed parallel algorithms, the MLPG method has a good scaling property in analyzing large problems and can be comparable to the FEM. Idelsohn and Oñate [63] pointed out that the key issue in choosing a meshless or mesh based method is how one can design an efficient algorithm for nodal connectivity that can keep the analysis running successfully and efficiently. They also showed that a good algorithm, not restricted to a mesh or non-mesh based method, should be almost the same order of computational time in term of nodes, i.e. $O(n\log(n))$ where n is the total number of nodes. Their meshless finite element method lies between the FEM and a meshless method.

In meshless methods, imposition of essential boundary conditions is not as straightforward as in the FEM. This is due to the lack of Kronecker delta property of the shape functions in meshless methods, i.e. the field approximation at a node does not take the nodal value. So a prescribed boundary value cannot be directly assigned to nodes as with the FEM. This problem has been much investigated ever since the invention of meshless methods and various methods have been proposed which will be described in Chapter 2. As will be demonstrated, now it does not seem to be an outstanding issue and in most cases different approaches perform fairly well. However some studies [64] have revealed solutions by the meshless methods become incompatible close to where boundary conditions are applied, which are not present in the FEM. This issue shows the different approximations in a global sense between the FEM and meshless methods [43, 65]. The latter is more affected by parameters in constructing shape functions and awaits more rigorous mathematical work.

Integration is perhaps the mostly significant issue that practically reduces the usefulness of meshless methods based on global or local weak forms (to be explained in more detail later). Firstly an integration scheme affects the convergence behaviour of results, and secondly a large number of integration points are normally needed in

meshless methods which significantly increases the computational cost. Attempts have been made to develop efficient, simple and accurate integration techniques. however, the status quo is that a simple integration scheme is not accurate or stable enough while integration schemes that show a good convergence rate are not efficient. A more detailed discussion will be found in this thesis in Chapters 2 to 4.

Adaptive analysis and error estimation is not as easy as first anticipated with meshless methods. Duarte and Oden [44, 45] demonstrated the convergence of the *h-p* could method mathematically however this is still a mesh-based method and their conclusions do not apply to methods like the EFGM or MLPG. The scene on this topic so far is hazy despite the various *a posteriori* error estimates and adaptive analysis proposed, such as the polynomial basis recovery method in [66, 67] and reduced nodal support for stress recovery by Chung and Belytschko [68], adaptive analysis schemes [41, 57, 69–71]. The *a posteriori* error estimates developed so far for meshless methods follow a similar approach to the well known ZZ patch recovery based error estimates [72, 73]. However the stress recovery methods are normally arbitrarily chosen and lack mathematical proofs. These recovery methods are effective only for some problems but not for general cases. For example, the method in [68] has been found by this author of the thesis not to be applicable to fracture modelling where singularity dominates around the crack tip.

There have been efforts devoted to developing hybrid methods to repair the shortcomings of meshless methods by taking advantage aspects of the FEM. In these hybrid methods, a problem domain is divided into several parts; some parts use a meshless approximation and for others the FE interpolation is adopted. Krongauz and Belytschko [74] proposed using the FE interpolation with the EFGM near parts of a domain where essential boundaries are applied. Gu and Liu [75] coupled the MLPG method with the FEM and BEM where the former was used for the part of the domain undergoing stress concentration or plastic deformation while FEM or BEM were employed to impose essential boundary conditions. Some other coupled methods are also proposed such as the EFGM with the FEM [76] and the MLPG with the scaled boundary method for analyzing problems of semi-infinite domains [77].

1.3 Fracture mechanics

A fracture is the (local) separation of an object or material into two, or more, pieces under the action of stress. Common examples of fractures in daily life are the breaking of glasses or bones and the snap of Cheerios when they are soaked in milk. The former two failures are clearly seen as due to external stresses while the latter is due to stresses from nonuniform expansion. Fracture mechanics is the field of mechanics concerned with the study of the formation and effects of cracks in materials. Since it is the main application of the numerical work in the thesis, it is necessary to have an overall systematic understanding of the development and basic concepts of fracture mechanics.

1.3.1 Historical review

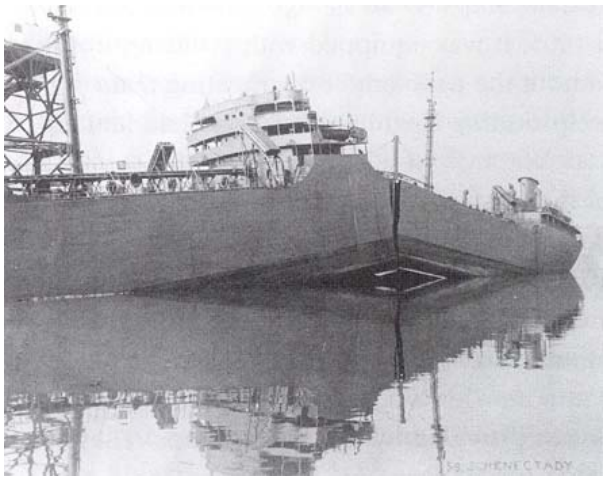


Figure 1.3: The Liberty Ship was completely broken into two parts due a fracture initiated at a weld. (`http://www-g.eng.cam.ac.uk/125/1925-1950/tipper3`)

The earliest mention of fracture mechanics may be traced back to 1500s when Leonardo da Vinci conducted experiments testing the tensile strength of wires by varying their lengths. The results recorded in his notebook showed that longer wires break more easily than short wires, which is understood today due to the microscopic flaws introduced along the length. Unfortunately the results were misinterpreted as consequence from heterogeneous material [78]. In 1921, an English aeronautical en-

gineer, A.A. Griffith, studied the effect of specimen sizes on the strength of glass rods and developed a theory to explain this phenomenon which is now regarded as the beginning of modern fracture mechanics. However his work was largely ignored by the engineering community until the 1940s after a number of fatal incidents happened due to fractures. The beginning of World War II put a pressing need for the manufacture of structures of light weight and high strength such as vessels for carrying large cargoes of weapons. The development of new materials and welding techniques at that time enabled engineers to achieve this goal. A large number of ships, known as the Liberty Ships, were produced in the US using new welding techniques. This was a great success until the failure of one Liberty Ship in 1943. The vessel was completely broken into two parts when sailing as shown in Figure 1.3. Investigations into the causes of the incident afterwards revealed that the fractures initiated at the welds on the deck done by semi-skilled workers [79]. In the post-war period, fracture mechanics was further propelled by the military competition induced by the Cold War and theories and design codes were developed for advanced structural analysis in aerospace engineering. Irwin's [80] team investigated the work by Griffith [81] and Westergaard [82] and, based on their work, they proposed a method to describe the singular stress field around a crack using a single parameter which is known today as the "stress intensity factor" (SIF). Williams [83] independently derived an analytical solution for the singular stress field around a crack tip essentially identical to Irwin's method. Wells [84] applied fracture mechanics to explain the fuselage failures in a number of Comet jet aircrafts caused by crack propagation initiated at the corners of windows. During 1960 to 1970 new design codes for structures considering cracks were developed in the US largely supported by the military and electrical industries. While in the UK, fracture mechanics was applied to welded structures and during the 1970s it was further motivated by the oil exploration in the North Sea as well as the development of the nuclear power industry. Most of the modern theories of fracture mechanics were developed in the period 1960 to 1980, breakthroughs in the area are marked by work of Eshelby, Rice and Shih which will be discussed in Chapter 5.

1.3.2 Inglis's solution for stress concentration around a notch

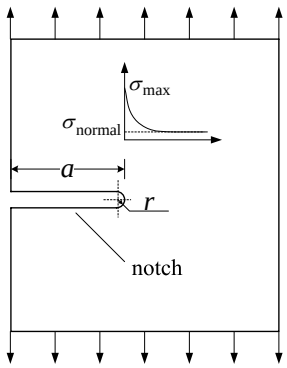


Figure 1.4: Stress concentration near a notch.

To understand the failure mechanism of material due to fracture, we can start with the stress concentration around a notch end as shown in Figure 1.4. A notch is a geometric discontinuity with definite depth and end radius. Bolt holes and screw threads are examples of notches in engineering practice. The boundary between a crack and a notch is often blurred by engineers, however they are different. If the notch end is smooth such as the circular shape shown in Figure 1.4, the stress is high around the end but the stress is not singular as tat of a crack tip. For a notch end with sharp corners such as a square shaped end, the stresses become singular but the order of singularity is lower than that of a crack tip. In the former, stress concentration around a notch can be quantified by a parameter, the stress concentration factor (SCF), noted as K_c which is a dimensionless value indicating the ratio of maximum stress at the end of notch over the normal stress applied. Inglis [85] studied the stress field near an elliptic hole and derived the analytical solution in linear elasticity for the SCF as

$$K_c = \frac{\sigma_{\max}}{\sigma_{\text{normal}}} = 1 + 2\sqrt{\frac{a}{r}} \tag{1.1}$$

where a and r are the notch depth and radius of notch end as shown in Figure 1.4. If $a = r$ then the notch becomes circular and $K_c = 3$ which is the well known solution for the SCF on a circular hole in an infinite plate. It can be seen that the SCF tends to increase when r decreases. The limit case is when r approaches zero then the

notch is reduced to a crack and

$$\sigma_{\max} = \lim_{r \to 0} \sigma_{\text{normal}} \left(1 + 2\sqrt{\frac{a}{r}}\right) \approx \infty \tag{1.2}$$

which shows $\sigma_{\max}$ becomes infinite with the value not dependent on the size of crack a or the applied stress σ. The SCF for notches is inadequate to qualitatively describe the severity of stress near a crack tip. However this solution is still of great importance in the development of the theory of fracture mechanics as will shown in the following section. In engineering applications, the mechanical behavior of a material with a notch provides important information on the fracture properties of the material. The most widely used tests on notches are known as the Charpy v-notch test, which was invented by Georges Augustin Albert Charpy (1865-1945). A Charpy test is a standardized high strain-rate test to determine the amount of energy absorbed by a material during fracture. Today it continues to be used as an economical quality control method to determine the notch sensitivity and impact toughness of a material.

1.3.3 Griffith's theory

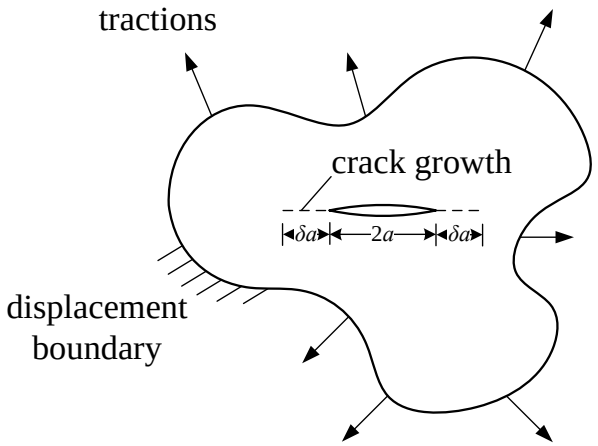


Figure 1.5: A thin plate with an initial crack which grows under constant loading conditions.

The first study on the effects of the flaw size over the strength of material was by Griffith [81] in 1921. He tested the tensile strength of rods made of "comparatively hard English glasses" and observed that the average tensile strength gets higher

with decreasing diameter of cross-section. He suggested the low strength of a thick rod was due to the microscopic flaws introduced with an increase of the size of geometry. To prove his guess, he produced surface scratches of varying depth on glasses as flaws, much larger than microscopic flaws found in normal glass, to test the maximum surface tension under various loading conditions. It was observed that the product of the square root of the flaw length a and the strength at fracture, noted as σ_f, was nearly constant, i.e.

$$\sigma_f \sqrt{a} = C \ , \tag{1.3}$$

i.e. traction σ_f decreases with an increase of a. To find the physical meaning of C, Griffith developed brittle crack failure theory by studying a thin plate of unit thickness containing an initial crack under constant loading condition as shown in Figure 1.5. The theory assumed the reduction of potential energy is dissipated as the energy to create new crack surfaces, i.e.

$$-\frac{\partial \Pi}{\partial a} = \frac{\partial W_{\text{cr}}}{\partial a} \tag{1.4}$$

where Π is the total potential energy given by

$$\Pi = -U_E + W \ , \tag{1.5}$$

U_E is the elastic strain energy and W is the external work. Under constant loading condition $W = 2U_E$ and therefore $\Pi = U_E$ and Equation (1.4) becomes

$$\frac{\partial U_E}{\partial a} = \frac{\partial W_{\text{cr}}}{\partial a} \tag{1.6}$$

The elastic strain energy U_E was derived using Inglis [85]'s solution for an elliptic cavity of zero radius in an infinite plate

$$U_E = \frac{\pi \sigma_f^2 a}{E} \tag{1.7}$$

where E is the Young's modulus of the material. The surface energy W_{cr}as a function of crack length is given by

$$W_{\text{cr}} = 2 \times 2a\gamma = 4a\gamma \tag{1.8}$$

where $2a$ is the size of crack length and γ is defined as the density of surface energy, i.e. the energy needed to form new surfaces per unit length and the factor 2 indicates two crack surfaces are formed, an upper and a lower. Substituting Equation (1.7) and (1.8) into (1.4) gives

$$\sigma_f = \sqrt{\frac{2E\gamma}{\pi a}} \tag{1.9}$$

Now introducing the total energy as the sum of potential energy and surface energy

$$U_{\text{tot}} = \Pi + W_{\text{cr}} = U_E + W_{\text{cr}} \ . \tag{1.10}$$

Plotting U_E, W_{cr} and U_{tot} in Figure 1.6, it can be seen that at a critical crack

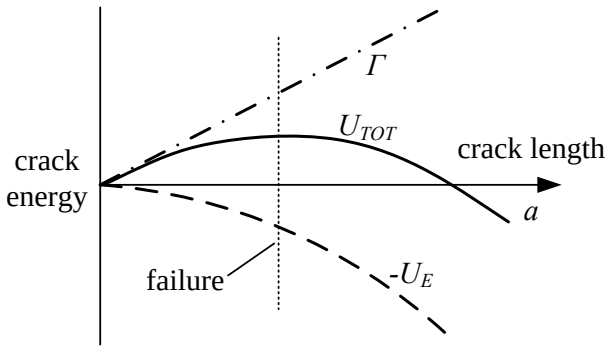


Figure 1.6: Energy changes of Griffith's problem

length, U_{tot} attains a peak value, i.e. the stationary point $\delta U_{\text{tot}} = 0$, beyond which the total energy decreases by increasing the crack length. Substituting Equation (1.9) into (1.3) gives

$$C = \sqrt{\frac{2E\gamma}{\pi}} \ . \tag{1.11}$$

Equation (1.6) shows the critical state when crack propagation is triggered and the crack starts to become unstable. When the *r.h.s* is greater than the *l.h.s*, crack propagation continues and when the *r.h.s* is less than the *l.h.s*, the crack is stable and no crack growth occurs. Introducing a parameter G to note the change of potential energy

$$G = -\frac{\partial \Pi}{\partial a} = \frac{\pi \sigma_f^2 a}{E} \ , \tag{1.12}$$

it can be seen that G can be regarded as a crack driving force per unit crack length and the state of crack growth can be identified via the relation between G and surface energy density

$$\begin{aligned} G &\geq 2\gamma \quad \text{crack propagation takes place} \\ G &< 2\gamma \quad \text{no crack propagation} \end{aligned} \tag{1.13}$$

Griffith's theory showed excellent agreement with predicted fracture stresses from experimental results for glasses. However, there are some limitations of the theory. Firstly, it is based on the assumption of an pre-existing crack so cannot be used to explain crack initiation. Secondly it does not provide prediction of the crack path during propagation. These issues have been recently revisited by Francfort and Marigo [86, 87] and some variational models of crack evolution have been proposed which can deal with the crack initiation and crack path continuity problems.

1.3.4 Irwin's modification

Attempting to apply Griffith's theory for metal fracture problems was not successful. Irwin [80] found that cracks formed in metals are not brittle but the area ahead of the crack tip undergoes plastic deformation before a cracked surface is formed. Thus the change of potential energy is dissipated as plastic energy and surface energy, i.e.

$$-\frac{\partial \Pi}{\partial a} = \frac{\partial W_{\text{cr}}}{\partial a} + \frac{\partial U_P}{\partial a} \tag{1.14}$$

where U_P is the plastic energy. The above expression is known as Irwin's law and can be rewritten as

$$G = 2\gamma + G_p \tag{1.15}$$

where $G_p = \dfrac{\partial U_P}{\partial a}$. The critical stress where a crack grows is modified to

$$\sigma_f = \sqrt{\frac{EG}{\pi a}} \ . \tag{1.16}$$

For brittle materials $G_p = 0$, and Equation (1.16) is reduced to (1.9).

Another contribution of Irwin is the introduction of the SIF as an expression of energy release rate. Based on the study of Westergaard's solution for crack tip field

[82], Irwin found the singular stress field can be described by a single parameter, namely the SIF and all fracture behaviour can be described by the linear combination of the three crack modes as shown in Figure 1.7, namely

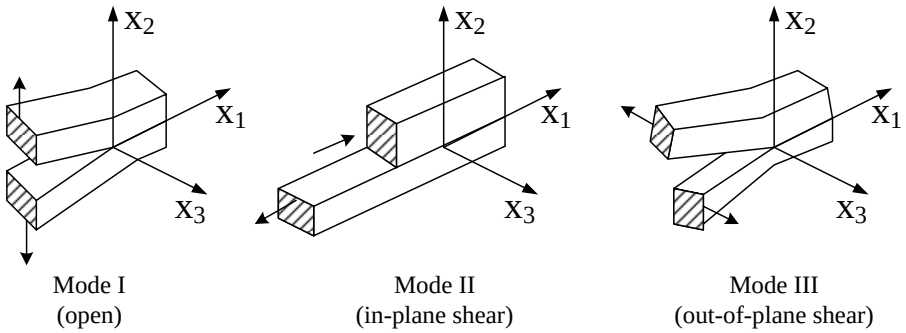


Figure 1.7: The three fracture modes

- mode I crack (opening mode) which is driven by tensile stress normal to the crack surface,
- mode II crack (sliding mode) which is driven by shear stress in the plane of crack surface and perpendicular to the crack font,
- mode III crack (tearing mode) which is driven by shear stress acting out of the plane of the crack surface.

Crack propagation found in materials can be decomposed into the three modes. Irwin also found the relation between the energy dissipation and the SIF as

$$G_{\mathrm{I}} = \begin{cases} \dfrac{K_{\mathrm{I}}^2}{E} & \text{plane stress} \\ \dfrac{K_{\mathrm{I}}^2(1-\nu^2)}{E} & \text{plane strain} \end{cases} \tag{1.17}$$

where K_{I} is the SIF for mode I crack. Similar relations can be found for mode II and III cracks and the total energy release rate is the sum of the three modes as

$$G = G_{\mathrm{I}} + G_{\mathrm{II}} + G_{\mathrm{III}} \ . \tag{1.18}$$

1.4 Numerical methods for fracture modelling

Analytical analysis for crack tip fields, such as Williams's and Westergaard's solutions, are limited to simple geometries and boundary conditions. Numerical methods for fracture modelling are the only way to find solutions to more sophisticated problems found in the real world.

1.4.1 Conventional methods

Conventional methods for fracture modelling are the boundary collocation method (BCM), the FEM and the BEM. Today they or their advanced forms, such as the dual BEM, remain the most powerful tools in fracture modelling. In the early application of the FEM in fracture modelling, trials were made to refine the elements around a crack tip to improve the solution accuracy [88]. Later on, the quarter-point method [89, 90] is proposed where the mid-node of an isoparametric element is moved close to a crack tip so that the singular stress field can be exactly captured. The method was shown accurate enough for calculating the SIFs, however the method requires remeshing around a crack tip and repositioning of mid-nodes during crack propagation. The implementation can be quite complicated for arbitrary crack surface in 3D. The BEM is a numerical computational method for solving linear partial differential equations formulated in boundary integral form. Today it remains one of the most powerful tools for simulating fracturing processes in solids. The BEM fits the given boundary conditions into the integral equation instead of solving values throughout the space defined by a PDE. So it has the advantage of smaller computer memory and a block-like matrix structure using the multi-region technique and provides much potential for solving large-scale problems with fewer degrees of freedom compared with the FEM. Recently an enrichment-based PU concept has been introduced into the BEM for accurate computation of SIF [91].

1.4.2 XFEM and meshfree and other PU based methods

Belytschko and Black [92] introduced *a priori* knowledge of analytical solution of a problem field into the FE approximation and developed this into the extended

finite element method (XFEM) [93] based on the concept of PU. This publication launched a new research surge in computational mechanics developing and applying the XFEM for solving various engineering problems, as described in a recent review by Belytschko et al. [94]. Fracture modelling is perhaps the most mature application of the XFEM and recent research interests have moved to the size effects of fractures as well as multiscale modelling. Rannou et al. [95] developed a level multigrid XFEM for 3D crack modelling including multilevel adaptive refinement of the element around crack surface.

The first application of the EFGM in fracture modelling is presented in the original paper on the EFGM [3] and later in a paper on crack propagation [96]. The first attempt to apply the EFGM for 3D fracture modelling was done by Sukumar et al. [97]. A coupled FEM-EFGM approach for 3D fracture modelling was proposed in [98] where the FEM is used in most of the domain and a meshless model is used only around the crack. Krysl and Belytschko [99] extended the EFGM for 3D dynamic fracture using the visibility criterion and adaptive analysis based on stress gradients. Recently, a particle based extended EFGM (XEFGM) [4, 100–105] has been formulated where crack growth is represented discretely by activation of crack surfaces at individual particles for cohesive crack modelling. However this is somewhat far from we are concerned with here. Extensive use is made later in this thesis of the LSM as a component of the final fracture modelling procedure. Literature on applying the EFGM for 3D fracture modelling in terms of fracture parameters are limited to [97, 99, 106–108].

1.5 Level set methods

The level set method (sometimes abbreviated as LSM) is a recently developed numerical method to capture the motion of interfaces and shapes in an implicit way. The name “level sets” refers to the sets which collect points having the same certain level, i.e. value of signed distance. It provides an implicit way of describing the geometry of a surface by measuring the shortest distance from any point inside the domain to the boundary of the surface. The LSM has proven efficient and powerful

in simulating problems such as burning flames, image enhancement, film etching and crystal growth [109]. In the following we will present some basic concepts of the LSM, i.e. implicit interface and signed distance functions.

1.5.1 Implicit interface and signed distance functions

In an explicit interface representation method, a number of points belonging to the interface are sampled and recorded. Alternatively, implicit interface representation defines the interface as isocontours of a function. For example, consider an interface shown as a closed solid curve in Figure 1.8. A domain closed by the interface is noted as Ω and its boundary (noted as $\partial\Omega$) is the interface itself. In an implicit method, a set of isocontours, shown as the dashed curve, are used to describe the interface. Contours inside Ω take negative values while those outside are positive. Each isocontour collects all the points having the same distance value to the interface by

$$\phi(\mathbf{x}) = \text{sign}\left(\vec{N}(\mathbf{x}_C) \cdot (\mathbf{x} - \mathbf{x}_C)\right) \|\mathbf{x} - \mathbf{x}_C\| \tag{1.19}$$

where $\mathbf{x}$ contains the coordinates of an arbitrary point, $\mathbf{x}_C$ is the closest point on the interface to $\mathbf{x}$, $\phi(\mathbf{x})$ is termed the the implicit signed distance function and $\vec{N}(\mathbf{x}_C)$ is the unit outer normal at $\mathbf{x}_C$ calculated by [110–112]

$$\vec{N}(\mathbf{x}_C) = \frac{\nabla\phi}{\|\nabla\phi\|} \tag{1.20}$$

where $\|\cdot\|$ is the L_2 norm of a vector and is defined as follows. Suppose a vector $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ where n is the length of the vector and the L_2 norm of the vector is calculated by

$$\|\mathbf{X}\| := \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}\ . \tag{1.21}$$

The L_2 norm for the first occurrence is explained here and any of its use latter in this thesis will refer to the above definition. The zero contour $\phi(\mathbf{x}) = 0$ represents the interface. Normally, a set of data points are used to discretise the implicit functions ϕ. It is usually convenient to generate the points from the grid lines based on Cartesian coordinates, i.e. shown as the square mesh in Figure 1.8.

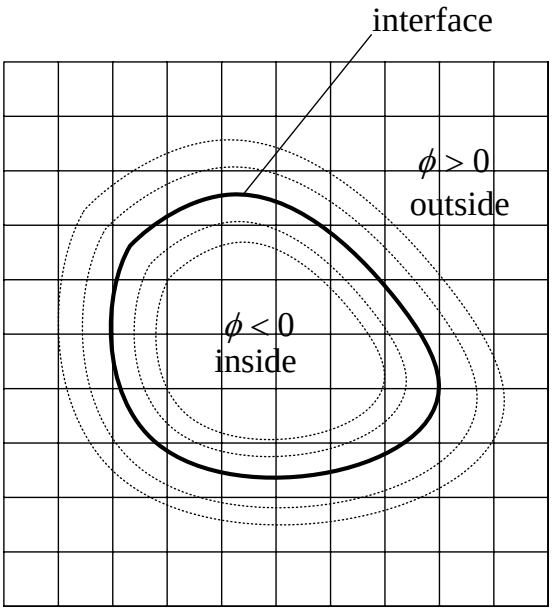


Figure 1.8: Signed distance function to describe an implicit surface.

1.5.2 Capturing motion interface

The advantage of the LSM is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects (this is called the *Lagrangian approach* [109]). Explicit methods such as the marker particle method, shown in Figure 1.9 suffer from problems of unstable results and the loss of information as the interface moves. For example, consider an interface which moves along its outer normal direction as shown in Figure 1.9. The marker particle method arranges scattered points along the interface shown as the black dots in the figure. As the interface grows it has been proved by Sethian [109] that by using the LSM the motion of the interface can be tackled satisfying the entropy condition, i.e. the geometric information is not lost as the interface moves. In the LSM the underlying equation to capture the motion is a Hamilton-Jacobi equation in the form of

$$\begin{aligned} |\nabla\phi(x)| &= F(x) \quad \text{in} \quad \Omega,\ F(x) \geq 0 \\ \phi &= g(x) \quad \text{on} \quad \Gamma \end{aligned} \tag{1.22}$$

where Ω is problem domain in $\mathbb{R}^2$ or $\mathbb{R}^3$, $F(x)$ on the *r.h.s* is known (normally *speed* at interface) which is determined by the underlying physical field that drives

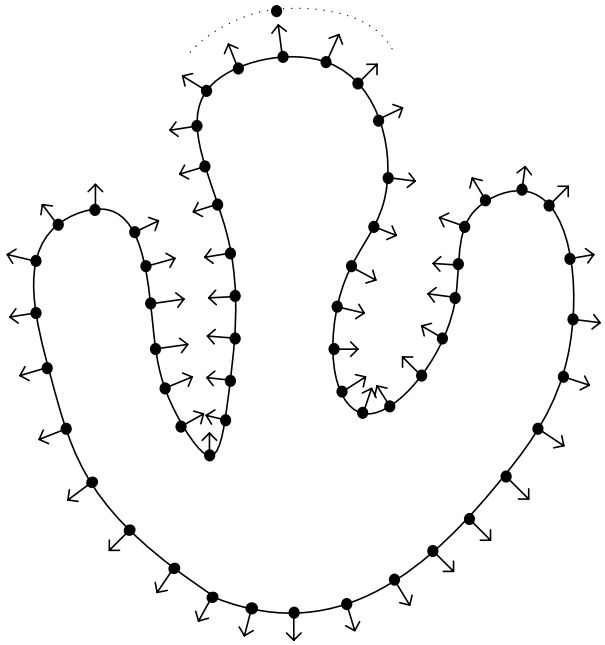


Figure 1.9: Marker particles used to discretise the front.

the movement of the interface. The function $g(x)$ corresponds to the boundary conditions for ϕ along a curve or surface Γ in Ω. The gradient of the interface function is calculated by

$$\nabla\phi(\mathbf{x}) = \left\{ \frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z} \right\}. \tag{1.23}$$

The gradient $\nabla\phi$ is orthogonal to the contours of ϕ and points in the direction of an increasing ϕ.

1.5.3 Level sets for 3D fracture Modelling

The accurate description of a crack's surface and front is important since the stress field around a crack front is strongly dependent on the front curvature and surface curvature. For example, an elliptic crack, which has a varying curvature along its front (the rim of the crack surface), has significant variations in the singular stress along the crack front. However conventional explicit ways (means parametric description of geometry) for describing the motion interface are incapable of dealing with the curvature accurately and suffer from problems in retaining the existing interface and show decreasing accuracy as the crack evolves. For example, in [113, 114]

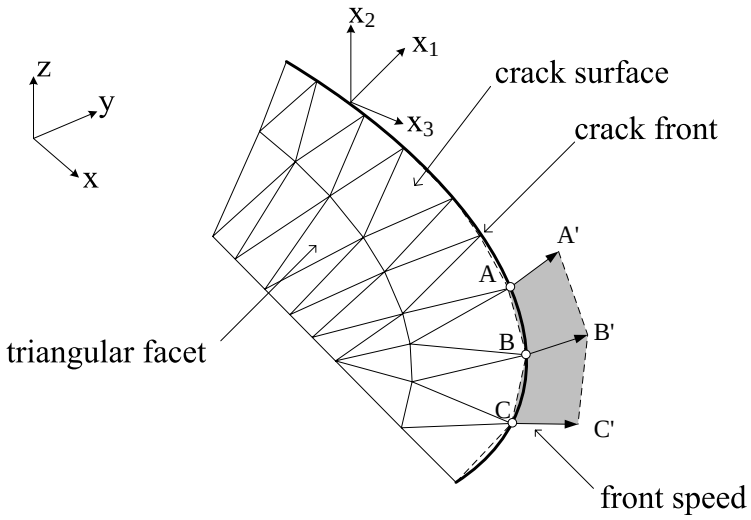


Figure 1.10: Piece-wise triangular facets used to describe a crack surface.

a crack surface in 3D is represented by piecewise triangular facets as shown in Figure 1.10, an explicit way of describing geometry like the marker particle method in Figure 1.9. Like many other explicit methods, new points will be added in this method as the crack front moves and thus the entropy condition cannot be guaranteed. Furthermore, to obtain the local coordinate systems using the coordinates of the points is cumbersome and lacks accuracy. The solutions to this problem is to use level sets to describe the crack geometry.

To use level sets in fracture modelling requires specific adaptation and implementation. The first use of the LSM for crack description is given by Stolarska et al. [115] in 2D with the XFEM. Two orthogonal level sets were used to describe a crack line (or combinations of lines) in 2D. A number of examples were tested and showed the feasibility of using level sets in describing and tracking crack propagation. The idea was later extended and developed in 3D crack modelling in a number of papers using the XFEM [107, 116, 117]. In this thesis, a numerical framework will be developed coupling a meshless method, the element-free Galerkin method (EFGM), and the level set method (LSM) for 3D crack modelling. In this framework, the EFGM analyzes the mechanical behaviour around a crack or cracks and the LSM accurately describes and captures the crack geometry as it propagates. This method of applying the level sets with the EFGM in 3D fracture modelling, is entirely new and to the author's knowledge has never been presented before.

1.6 Outline of the thesis

This thesis comprises two major parts where the first, from Chapter 2 to 4 concerns the theory and fundamentals of meshless methods. The second part from Chapter 5 to 8 is a contribution to the application of the EFGM in fracture modelling, especially in 3D. The contents of each Chapter are detailed in the following.

In Chapter 2 the formulation and implementation of the EFGM are described. Modified weak forms, including the penalty method, Lagrange multiplier method and the Nitsche-like method for imposing the essential boundary conditions are derived. Discretisation of the weak forms are derived and the relations between various weak forms are outlined. Implementation issues are discussed and new algorithms and a new data structure are introduced which improve computational efficiency.

Chapter 3 deals with the use of orthogonal basis function in the EFG shape functions from the MLS approximation. It is shown that the source of difficulties cited in the literature is not adequately dealt with by adopting the orthogonal basis and inaccuracies remain. An alternative approach is proposed which maintains the positive features of the use of orthogonalization while avoiding the inaccuracies of the original approach.

Chapter 4 investigates the control of discretisation error and discusses adaptivity in the EFGM highlighting the differences from the FEM. It is demonstrated that the (now) conventional procedures for error analysis and adaptivity used in the finite element method require careful application in the EFGM, otherwise competing sources of error work against each other. The control of error in the EFGM is shown to be a competition between the field and its derivatives. Numerical studies show that for the EFGM this error cannot be easily split into component parts to assess appropriate refinement or basis change strategies without conducting some sensitivity analyses. While numerical tests are performed only for the EFGM, the conclusions are applicable to other meshless methods based on the concept of nodal support.

Chapter 5 reviews the methods developed in the EFGM for fracture modelling. Present methods for fracture modelling are summarised into extrinsic enrichment

and intrinsic enrichment. The use of nodal support rather than elements leads to some major differences in enrichment procedures as compared to the XFEM. The displacement and stress results obtained using the enrichment in the EFGM are used to extract the fracture parameters. Various methods of arriving at the SIF as fracture parameters are discussed and compared in a systematic way.

Chapter 6 presents the use of level sets in fracture modelling with meshless methods. The geometry of an arbitrary crack surface in 3D can be described by two orthogonal level sets. They together build a local curvilinear coordinate system near crack front and facilitates the stress analysis and computation of fracture parameters. New formulations to introduce the jump term based on the level sets in the EFGM for cracks of arbitrary geometry in 2D and 3D are proposed. A number of problems with various geometries are tested showing the feasibility and accuracy of using level sets for fracture modelling.

Chapter 7 develops a crack tip tying procedure to remove the spurious crack extension found when using the visibility criterion alone. The method can be used in 2D and 3D and avoids the potential difficulties in use of the diffraction method met with multiple cracks and non-planar crack fronts and faces. The formulation is based on the use of level set coordinates, is simple to implement and can be used for multiple crack problems. The formulation is tested for 2D examples.

Chapter 8 brings together all aspects of the preceding work and proposes a new framework coupling the EFGM and LSM for 3D crack modelling. In this framework, the level sets grid used to represent the crack geometry is decoupled from the meshless nodal arrangement so that the problem of spurious shifting of the original crack surface is eliminated. The level sets algorithms and data storage can be vectorized and greatly improve the computational efficiency. The simplest visibility criterion is used to introduce the jump term for an arbitrary crack surface with the help of the tying procedure. Some issues concerning efficient implementation due to the increase of complexity of crack geometry and stress analysis in 3D are addressed. A PU integration scheme near the crack front based on level sets is proposed. A number of 3D examples including edge crack, penny-shaped crack and elliptic crack subjected to various boundary conditions are tested to study the performance of the

present framework.

The thesis finishes with a summary of present work and some perspectives on further research in Chapter 9. Future research trends in meshless methods and the computational mechanics as a whole are discussed.

Chapter 2

Modified weak forms and implementation

2.1 Introduction

This chapter concerns the formulations and implementation issues of the EFGM. Firstly, the basic formulation of the EFGM is described including the derivation of shape functions from the MLS approximation, weight function, choice of nodal support and integration scheme of the weak form. Secondly, due to the lack of delta property in the EFGM shape functions, the weak form in the FEM where the essential boundary conditions are directly applied is no longer valid. Additional constraints along essential boundaries have to be added to obtain compatible field solutions. Some well-used methods to modify the weak forms in meshless methods, namely the penalty method, the Lagrange multiplier method and the Nitsche-like method are derived and an outline of the interconnection between these methods provided. The discretisation of the weak forms are detailed and the differences highlighted. Finally, implementation issues are discussed and algorithms and data structures to improve the computational efficiency are proposed.

2.2 The EFGM shape functions and weight functions

2.2.1 Derivation of shape functions using the MLS approximation

For convenience of later derivation, the formulation of the EFGM shape functions based on the MLS approximation [27–29] is described in the following. Given a set of n data pairs $\mathbf{U} = \{u_I, \mathbf{x}_I\}, I = 1, 2, \cdots n$ to approximate an unknown field value $u(\mathbf{x})$, the MLS approximation can be constructed as

$$u^h(\mathbf{x}) = \sum_I^n \phi_I(\mathbf{x}) u_I = \mathbf{\Phi}(\mathbf{x})\mathbf{u} \tag{2.1}$$

where $u^h(\mathbf{x})$ denotes the approximate value of $u(\mathbf{x})$, n is the number of nodes in support at $\mathbf{x}$ and $\phi_I(\mathbf{x})$ is the shape function of node I at $\mathbf{x}$. $\mathbf{\Phi}(\mathbf{x})$ is a $1 \times n$ matrix collecting together the shape functions ϕ_I, and $\mathbf{u}$ is a vector containing the fictitious nodal values. Here "fictitious" indicates the approximation at a node, e.g. $u^h(\mathbf{x}_I)$, does not necessarily equal to nodal value u_I as in the FE method. If $u(\mathbf{x})$ is approximated as a polynomial then

$$u^h(\mathbf{x}) = \sum_j^m p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \tag{2.2}$$

where $m =$ is the number of monomials in the basis vector $\mathbf{p}(\mathbf{x})$, (e.g. $m = 3$ for a linear basis in 2D or a quadratic basis in 1D) and $\mathbf{a}(\mathbf{x})$ is a vector of coefficients. The basis vector $\mathbf{p}^T(\mathbf{x}) = [p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})]$ is built using Pascal's triangle in 2D and Pascal's pyramid in 3D, and for convenience it is called Pascal basis here. In the MLS approximation, the shape functions are obtained by minimizing a weighted residual J to determine the coefficients $\mathbf{a}(\mathbf{x})$ where

$$J(\mathbf{x}) = \sum_I^n w(\mathbf{x}) \left[\mathbf{p}^T(\mathbf{x}_I)\mathbf{a}(\mathbf{x}) - u_I\right]^2 \tag{2.3}$$

where $w_I(\mathbf{x}) \equiv w(\mathbf{x} - \mathbf{x}_I)$ is the weight function for point $\mathbf{x}$ (see next section for a discussion on the weight function). Equation (2.3) leads to the following matrix equation

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{u} \tag{2.4}$$

where the elements of matrix $\mathbf{A}(\mathbf{x})_{m\times m}$ are given by

$$A_{jk} = \sum_{I}^{n} w_I(\mathbf{x}) p_j(\mathbf{x}_I) p_k(\mathbf{x}_I) \quad j,k = 1,\ldots,m \tag{2.5}$$

and the elements of matrix $\mathbf{B}(\mathbf{x})_{m\times n}$ by

$$B_{jI} = w_I(\mathbf{x}) p_j(\mathbf{x}_I) \quad j = 1,\ldots,m,\ I = 1,\ldots,n. \tag{2.6}$$

The coefficients $\mathbf{a}(\mathbf{x})$ can be found from (2.4) by inverting $\mathbf{A}(\mathbf{x})$

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}(\mathbf{x})^{-1}\mathbf{B}(\mathbf{x})\mathbf{u},$$

so (2.2) becomes

$$u^h(\mathbf{x}) = \mathbf{p}(\mathbf{x})^T \mathbf{A}(\mathbf{x})^{-1}\mathbf{B}(\mathbf{x})\mathbf{u} \tag{2.7}$$

and the shape functions are found by comparison with (2.1) as

$$\mathbf{\Phi} = \mathbf{p}^T\mathbf{A}^{-1}\mathbf{B} \tag{2.8}$$

where the dependence on $\mathbf{x}$ for all terms has been removed for clarity. The derivatives of the shape functions can be found as

$$\mathbf{\Phi}_{,k} = \mathbf{p}_{,k}^T\mathbf{A}^{-1}\mathbf{B} + \mathbf{p}^T\left(\mathbf{A}_{,k}^{-1}\mathbf{B} + \mathbf{A}^{-1}\mathbf{B}_{,k}\right) \tag{2.9}$$

where k denotes the coordinate index and

$$\mathbf{A}_{,k}^{-1} = -\mathbf{A}^{-1}\mathbf{A}_{,k}\mathbf{A}^{-1}\ . \tag{2.10}$$

$\mathbf{A}$ and $\mathbf{B}$ can be written in matrix form as

$$\mathbf{A} = \mathbf{P}\mathbf{W}\mathbf{P}^T \tag{2.11a}$$

$$\mathbf{B} = \mathbf{P}\mathbf{W} \tag{2.11b}$$

where $\mathbf{P}$ is an $m \times n$ matrix defined by

$$\mathbf{P} = \Big[\mathbf{p}(\mathbf{x}_1),\ldots,\mathbf{p}(\mathbf{x}_n)\Big] \tag{2.12}$$

and $\mathbf{W}$ is an $n \times n$ diagonal matrix

$$\mathbf{W} = [\ diag(w_1(\mathbf{x}),\ldots,w_n(\mathbf{x}))\ ]_{n\times n}\,. \tag{2.13}$$

2.2.2 Weight function and choice of size of nodal support

The use of the weight function w (see Equation (2.3)) in constructing the shape functions is a most important feature of meshless methods compared with the FEM. In the FEM, elements connected to a node defines the area where a nodes exert influence over the approximation. Such influence in the EFGM is regulated by the weight function defined at a node. Here the weight function is a radial function which is directionally independent. There are some basic requirements for weight function as has been discussed in [118, 119] namely:

1. non-negativity,
2. monotonically decreasing with distance,
3. compactly supporting,
4. continuous and differentiable.

Various weight functions have been proposed for the EFGM including exponential, conical, spline and rational functions. However the choice of weight function is to some extent arbitrary since there is no rigorous mathematical proof available at the moment to judge which type of weight function is more favourable than others. Studies comparing the performance of weight functions on error convergence rate have been carried out in [3, 119] indicating one type better than the others for certain problems but further research on this topic is necessary. In this thesis three popular weight functions will be used: a rational function w^1, an exponential (actually a shifted Gaussian) w^2, and a conical function w^3. These will appear in test examples to be presented in Chapter 4.

$$w^1(d_I) = \begin{cases} \dfrac{d_{mI}^2}{{d_I}^2 + \ell^2 d_{mI}^2}\left(1 - \dfrac{{d_I}^2}{d_{mI}^2}\right)^{2k}, & d_I \le d_{mI} \\ 0, & d_I > d_{mI} \end{cases} \tag{2.14}$$

$$w^2(d_I) = \begin{cases} \dfrac{e^{-(d_I/c)^{2k}} - e^{-(d_{mI}/c)^{2k}}}{1 - e^{-(d_{mI}/c)^{2k}}}, & d_I \le d_{mI} \\ 0, & d_I > d_{mI} \end{cases} \tag{2.15}$$

$$w^3(d_I) = \begin{cases} 1 - \left(\dfrac{d_I}{d_{mI}}\right)^{2k}, & d_I \leq d_{mI} \\ 0, & d_I > d_{mI}. \end{cases} \tag{2.16}$$

Other types of spline weight functions are a cubic spline function w^4

$$w^4(d_I) = \begin{cases} \dfrac{2}{3} - 4\left(\dfrac{d_I}{d_{mI}}\right)^2 + 4\left(\dfrac{d_I}{d_{mI}}\right)^3, & d_I \leq \dfrac{1}{2}d_{mI} \\ \dfrac{3}{4} - 4\left(\dfrac{d_I}{d_{mI}}\right) + 4\left(\dfrac{d_I}{d_{mI}}\right)^2 - \dfrac{3}{4}\left(\dfrac{d_I}{d_{mI}}\right)^3, & \dfrac{1}{2}d_{mI} < d_I \leq d_{mI} \\ 0, & d_I > d_{mI} \end{cases}, \tag{2.17}$$

and a quartic spline function w^5

$$w^5(d_I) = \begin{cases} 1 - 6\left(\dfrac{d_I}{d_{mI}}\right)^2 + 8\left(\dfrac{d_I}{d_{mI}}\right)^3 - 3\left(\dfrac{d_I}{d_{mI}}\right)^4 & d_I \leq d_{mI} \\ 0, & d_I > d_{mI} \end{cases}. \tag{2.18}$$

In the above d_I denotes the distance between a point and node I, d_{mI} is the size of the nodal support, k and ℓ are integer parameters and c is a constant which controls the relative weights. A comparison of the first three weight functions is shown in Figure 2.1 showing the effect of varying the parameter k on function shape. In the examples presented later in this thesis, $k = 1$ is used and $\ell = 1$ for w^2. These weight functions are often high-order differentiable which contributes to the smooth and high order consistency field approximation obtained with MLS-based meshless methods. Weight function w_1 over a 2D domain is plotted in Figure 2.2. In Figure 2.3, the first and second order derivatives of w_1 over a 2D domain are plotted.

The size of the nodal support is critical, as highlighted in many papers, however advice varies. In [120], $d_{mI} = \alpha r_{mI}$, where r_{mI} is the distance from node I to its second nearest neighbour, or is the size of the background integration cell containing the node. It is suggested that α be reduced to 1.0 for nodes that are located where the field undergoes high gradient however this is rarely known *a priori* for real problems. Dolbow and Belytschko [121] suggest $d_{mI} = d_{max}c_I$ is used where c_I is a "large enough value to contain sufficient number of nodes in support" and d_{max} is a factor which takes an empirical value between 2.0 and 4.0. In [119] the nodal support size is an averaged nodal spacing value multiplied by a factor. To aid the

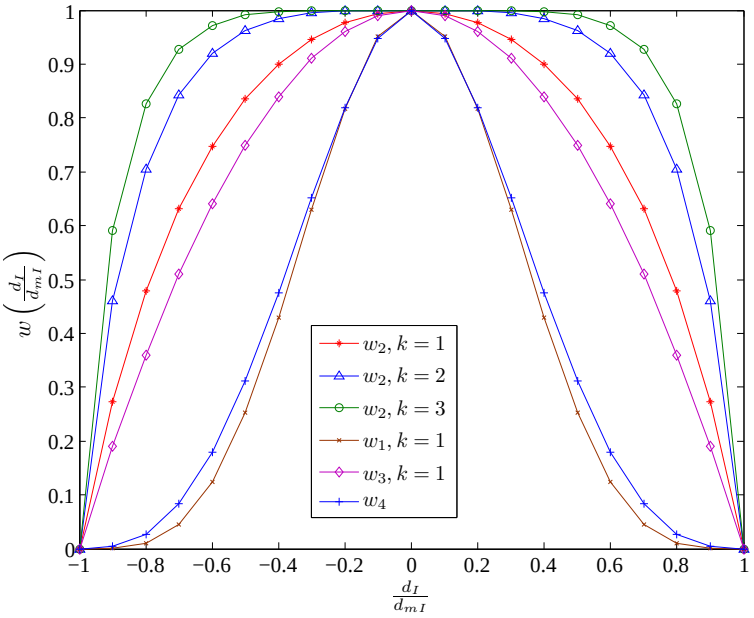


Figure 2.1: Plot of rational, conical, exponential and spline weight functions.

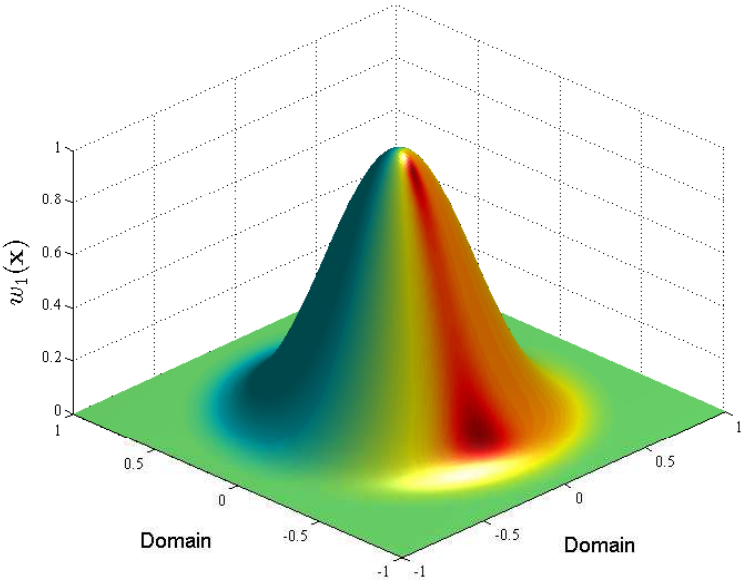
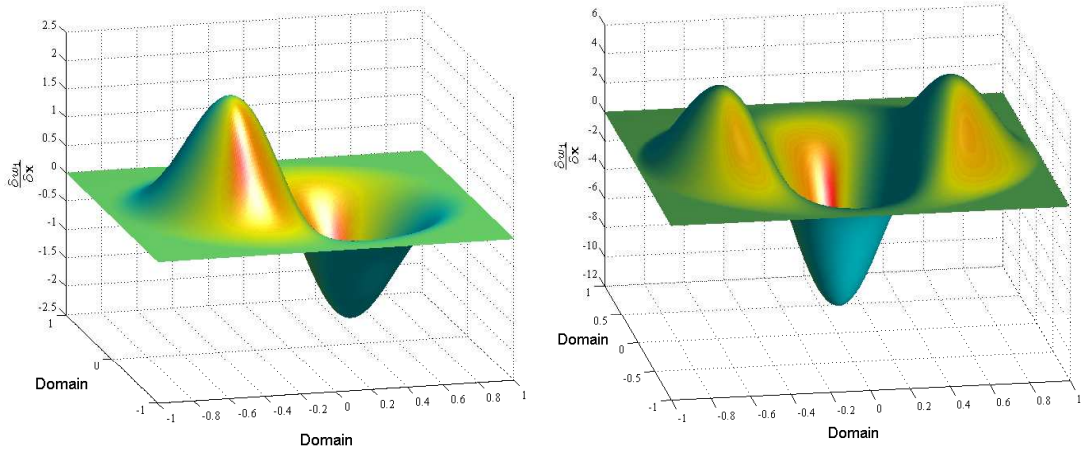


Figure 2.2: Plot of rational weight function w_1 over a 2D domain.



(a) first order derivatives (b) second order derivatives

Figure 2.3: Plot of first and second order derivatives of rational weight function w_1 over a 2D domain.

analysis which follows these rules can all be written in the following single form:

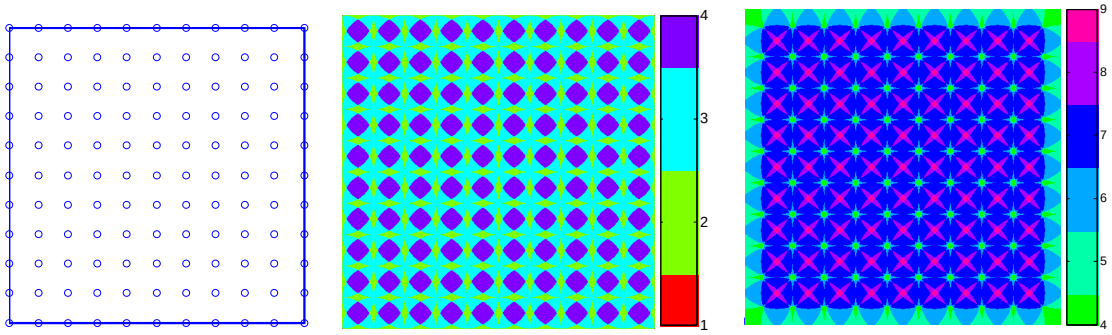
$$d_{mI} = \alpha c_I \tag{2.19}$$

where α is a dimensionless value for scaling the size of the nodal support and c_I can be set using different rules, such as the average node spacing near node I, the longest edge of the background integration cell containing I, the distance to the nearest node in support or the minimum value of a radius to include surrounding nodes to form a polygon around I. Whichever method is used used to set c_I, d_{mI} may be scaled up or down by adjusting α. The procedure to set nodal support size can be summarized as follows:

1. Use nodal topology information around node I to determine c_I.

2. Make an empirical choice for α from a certain range.

In order to ensure a solution, α must be chosen so that sufficient nodes are in support for the basis used throughout the domain. The coefficient α plays the role of a "magnifier" that changes the density of overlapping of supports. To illustrate variation in support, we consider an example of a square domain with uniformly distributed nodes (Figure 2.4(a)). The colour bars in Figure 2.4(b) and (c) indicate the number of nodes in support. Overlapping is greater for $\alpha = 1.5$ than for $\alpha = 1.0$.



(a) nodal arrangement (b) scale size $\alpha = 1.0$ (c) scale size $\alpha = 1.5$

Figure 2.4: Plot of the number of nodes in support over a square domain.

2.2.3 Properties of shape functions

In the FEM, the shape functions are constructed element-by-element and the computation is very straightforward. In the early years of development of the FEM, much effort was devoted to derive formulations for different types of elements. Examples are elements using low order interpolation, such as three noded triangular elements and four node quadrilateral elements in two-dimensions, and four noded tetrahedral elements and eight noded brick elements in three-dimensions for problems of constant strain. Elements using quadratic and cubic interpolation such as six noded triangular elements and twenty noded hexahedral elements are used for solving problems with high stress gradient. In the FEM the shape function of a node is unity at the node itself and vanishes at other nodes connected to it, which is known as the Kronecker-delta property. This property is desirable for the imposition of essential boundary conditions such as prescribed displacement and material boundaries. The shape functions in meshless methods by MLS share some common properties with the FEM, such as partition of unity and linear reproduction but not the delta property. The basic compulsory requirement of all kinds of shape functions is known as the *partition of unity* (PU) condition. A PU is a set of functions $\{N_I(\mathbf{x})\}$ that for every point in the domain under consideration

$$\sum_{I}^{n} N_I(\mathbf{x}) = 1 \ . \tag{2.20}$$

Shape function that satisfies PU can produce any rigid motion of the problem domain. There are other conditions that shape functions preferably should satisfy.

The first is the *linear field reproduction condition*, which is in 3D

$$\sum_{I}^{n} N_I(\mathbf{x}) x_I = x \tag{2.21}$$

$$\sum_{I}^{n} N_I(\mathbf{x}) y_I = y \tag{2.22}$$

$$\sum_{I}^{n} N_I(\mathbf{x}) z_I = z \tag{2.23}$$

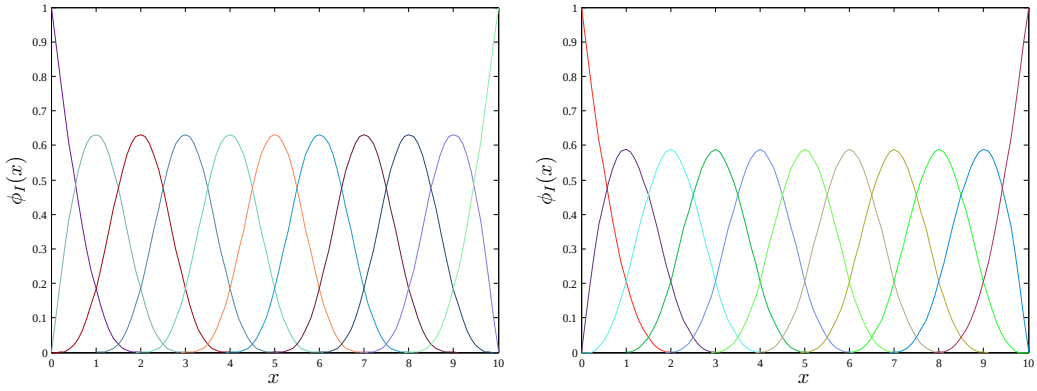
where x_I, y_I and z_I are coordinates of node I in support of $\mathbf{x}$. The above equations are required for the shape functions to pass the *patch test*, which is used to test finite elements. However, this condition is not compulsory as long as the solution converges. Many types of finite elements do not pass the patch test but are still widely used in commercial software. For shape functions constructed from MLS, we can derive further properties such as

$$\sum_{I}^{n} N_I(\mathbf{x})_{,i} \, x_I^j = \delta_{ij} \quad i, j = 1, 2, 3 \tag{2.24}$$

where x_I^j denotes the j th coordinate of $\mathbf{x}_I$. This is the consistent convergence property of shape functions by MLS procedure.

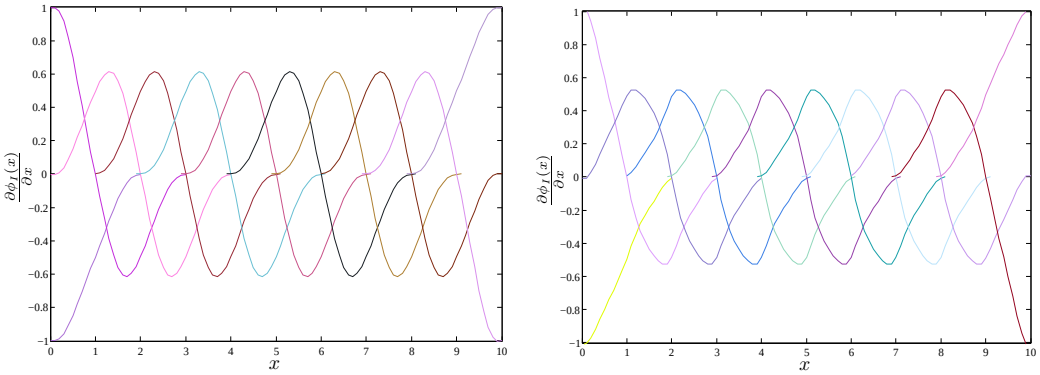
In Figure 2.5 shape functions are plotted over a 1D domain, where 11 nodes are located at $x = 0, 1, \cdots, 10$. The nodal support is set the same for each node at 2.1 and two types of weight function, namely w_1 and w_4, are compared. It can be seen that the nodal values of shape functions by w_1 tend to be larger than w_4. This phenomenon can be explained by the plot of weight functions in Figure 2.1 where changes of w_1 with respect to distance are more rapid than for w_4. The derivatives of the shape functions are plotted in Figure 2.6 and differences between using different weight functions are even clearer. The smoothness of derivatives are largely influenced by the size of nodal support and when this is reduced to 1.3, the derivatives of the shape functions are no longer smooth as shown in Figure 2.7.

As a further illustration of MLS-based shape functions, a plot of shape function values over a 2D domain is shown in Figure 2.8(b). Nine nodes are arranged in a grid over a 2×2 domain as shown in Figure 2.8(a) and the shape function is shown for the node at (1, 1). The derivatives of shape functions in with respect to the horizontal direction, e.g. x axis, are plotted in Figure 2.9. It can be seen that the shape function behaves in a similar way to that in 2D. The plot the derivative of



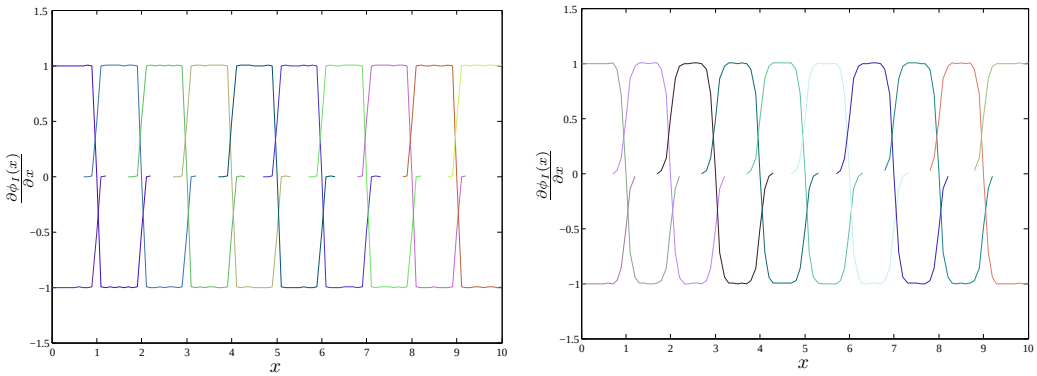
(a) Using w_1 weight function (b) Using w_4 weight function

Figure 2.5: Plot of nodal shape functions over a 1D domain.



(a) by w_1 weight function (b) by w_4 weight function

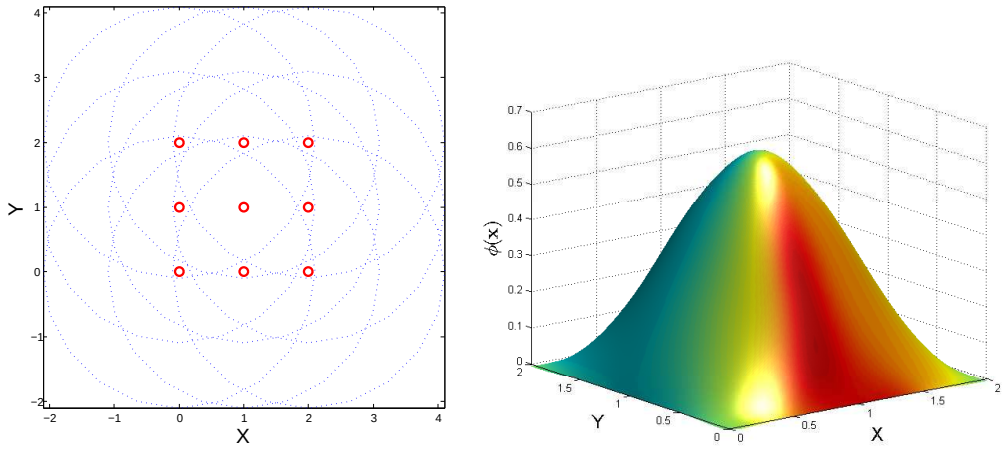
Figure 2.6: Plot of first order derivatives of shape functions over a 1D domain for $d_{mI} = 2.1$.



(a) Using w_1 weight function (b) Using w_4 weight function

Figure 2.7: Plot of first order derivatives of shape functions over a 1D domain for $d_{mI} = 1.3$.

shape function is a curved surface which decreases as a point moves in an increase direction of x. And this is also similar to the plot in 2D.



(a) nodal arrangement over 2D domain (b) shape functions at central node

Figure 2.8: Nodal arrangement used over a 2D domain and plot of shape functions over the domain for $d_{mI} = 2.1$ and using w_1.

2.3 Modified weak forms

In the FEM, essential boundary conditions can be directly applied to nodes since the shape functions in the FEM are interpolatory [122] and results obtained in this way satisfy compatibility conditions at essential boundaries. Meshless methods such as the EFGM use shape functions often constructed by the moving least squares approximation which does not satisfy the Kronecker delta property. Therefore the weak form used in a standard FEM where essential boundary conditions are directly applied to nodal values is no longer valid for a meshless method based on the MLS approximation. And the enforcement of essential boundary conditions requires additional treatment. Methods developed to fix this problem can be classified into the following approaches:

- Modification to the EFGM shape functions or global equations to allow the direct imposition of essential boundary conditions.
- Modification to the weak form to include a work term along the essential boundaries.

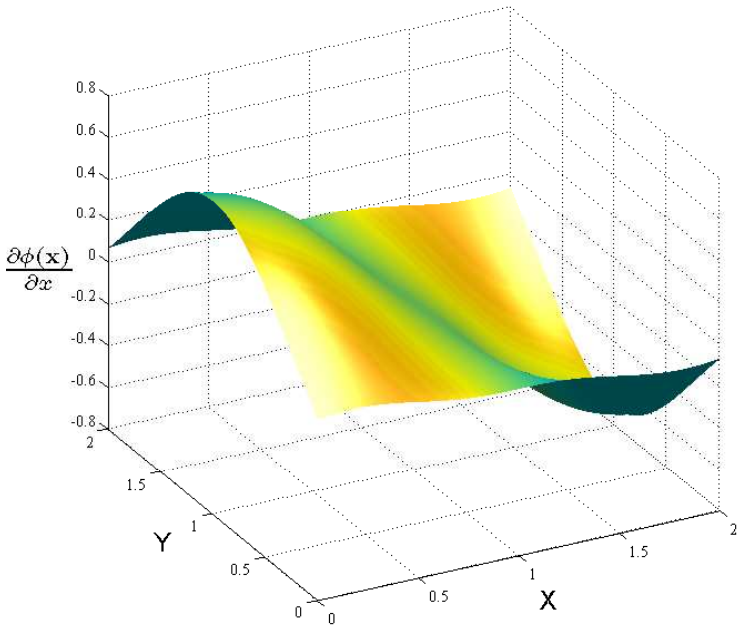


Figure 2.9: The derivatives of shape functions over a 2D domain for $d_{mI} = 2.1$ and using w_1.

- Coupling a meshless method with the FEM where the FE interpolation is used near the essential boundaries.

Examples of the first kind of approach are the use of a singular weight function [123], transformation methods [124] and maximum entropy shape functions. The transformation method requires special treatment to transform the global stiffness matrix, and is complicated to implement since the formulation needs to be changed for every problem. The maximum entropy (MAXENT) concept have been used to construct shape functions which possess the delta property [125]. Though the computational cost in obtaining the MAXENT shape functions is expensive, it can be coupled with the EFGM where boundary conditions are needed. A singular weight function is perhaps the simplest way to implement essential boundary conditions by this approach in that the only change to the standard EFGM is to use a singular weight function, such as

$$w(d_I) = \begin{cases} \dfrac{d_{mI}^2}{d_I^2 + \epsilon} \cos^2\left(\dfrac{\pi d_I}{2 d_{mI}}\right) & d_I \leq d_{mI} \\ 0, & d_I > d_{mI} \end{cases}, \tag{2.25}$$

where the definitions of d_{mI} and d_I are the same as in Equation (2.14) and ϵ is a small value parameter to avoid a zero value in the denominator (normally 10^{-10} is used [126]). Results obtained using this approach are however very sensitive to the choice of parameter ϵ and sometimes ill-conditioning occurs in determining the shape functions.

Coupling the FEM with the EFGM is proposed by Belytschko et al. [127] where FE interpolation is used near essential boundaries. This method requires a transitional region to ensure a smooth connection of approximation between the FE shape functions and the EFGM shape functions. Mesh is involved which is perhaps not an ideal solution to a meshless method. In this section, we focus on second kind of approach, namely modified weak forms, which has a simple general formulation and performs robustly. We will firstly state the governing equations of the strong form and then describe the corresponding weak forms in the meshless method.

2.3.1 Governing equations for elastostatics

Before the discussion on weak forms, it is essential to know the strong forms of the underlying PDEs in elastostatics. Though they are well known within the field of computational mechanics, for the convenience of derivation later, they are stated here. The problem being considered in this thesis is modelling stresses and displacements for small strain linear elastic homogeneous materials under static equilibrium condition. The displacement at a point in a Cartesian coordinate system (x, y, z) is described by the vector $\mathbf{u}$

$$\mathbf{u}^T = \begin{Bmatrix} u & v & w \end{Bmatrix} , \tag{2.26}$$

where the three components are orthogonal and each indicates the movement in a certain direction, e.g. u in x direction. $\mathbf{u}$ can also be noted in a tensorial form as

$$\mathbf{u}^T = \begin{Bmatrix} u_1 & u_2 & u_3 \end{Bmatrix} . \tag{2.27}$$

In this thesis, the vector notation in Equation 2.26 is used. To describe the force balance state at a point, e.g. P, an infinitesimal cube surrounding P is taken for analysis as shown in Figure 2.10. On each of the six surfaces of the cube,

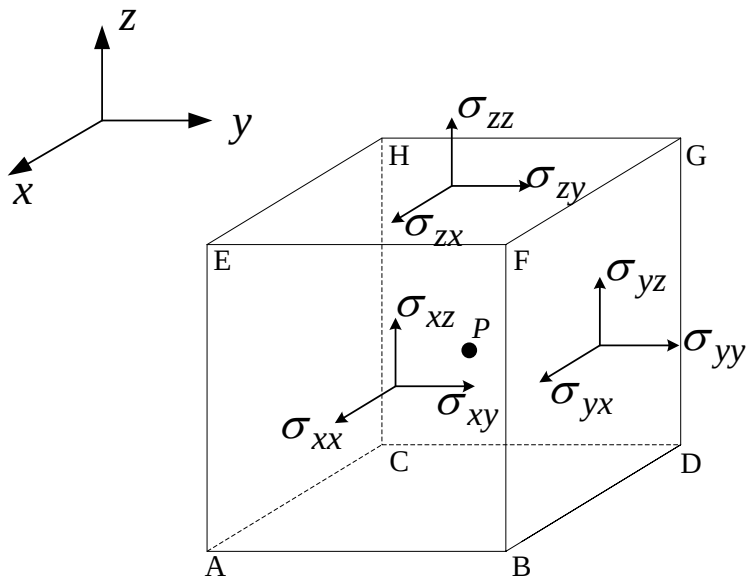


Figure 2.10: Definition of stress components on an infinitesimal cube surrounding a point P.

there exist three orthogonal stress components namely, one normal to the surface and the other two within the surface. For example, surface BDGF is subjected to normal stress σ_{yy} aligned with positive y axis and σ_{yx} and σ_{yz} in positive x and z directions respectively. Similarly surface EFGH is subjected to normal stress σ_{zz} and shear stresses σ_{zx} and σ_{zy}, and ABFE is subjected to normal stress σ_{xx} and shear stresses σ_{xy} and σ_{xz}. Opposite surfaces such as AEHC have the same stress components pointing in opposite direction to those on BDGF. Thus there are nine stress components to uniquely determine the elastic stress state at a point and they form a stress tensor σ_{ij} which can be written into a matrix form as

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} . \tag{2.28}$$

Note the first subscript of σ refers to the outer normal of the surface and the second refers to the direction of the stress. Due to the shear stress balance between neighbouring surfaces, e.g. $\sigma_{xy} = \sigma_{yx}$, the off-diagonal terms are always equal. Thus only six of the nine components are independent, and a stress vector is introduced only

with six components

$$\boldsymbol{\sigma}^T = \left\{ \sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \sigma_{xy} \quad \sigma_{yz} \quad \sigma_{xz} \right\} \ . \tag{2.29}$$

The above is known as the Voight notation of the elastic stress tensor which will be used in the derivations that follow. The deformation at a point is described by the infinitesimal strain tensor ε_{ij} where the rule between subscripts and orientations is similar to that of the the stress vector. The matrix form the strain tensor is

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{bmatrix} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{bmatrix} \tag{2.30}$$

where

$$\begin{aligned} \varepsilon_{xy} &= \frac{1}{2}\gamma_{xy} \\ \varepsilon_{yz} &= \frac{1}{2}\gamma_{yz} \\ \varepsilon_{zx} &= \frac{1}{2}\gamma_{zx} \ , \end{aligned} \tag{2.31}$$

and the terms with γ is normally termed as "engineering strain". Again the off-diagonal terms are also symmetric that $\varepsilon_{ij} = \varepsilon_{ji}$. The Voight notation of the strain tensor is written as

$$\boldsymbol{\varepsilon}^T = \left\{ \varepsilon_{xx} \quad \varepsilon_{yy} \quad \varepsilon_{zz} \quad \gamma_{xy} \quad \gamma_{yz} \quad \gamma_{xz} \right\} \tag{2.32}$$

which will be used in later derivations. The governing equations in elasticity are equilibrium, compatibility and constitutive equations. The **equations of equilibrium** for statics describe the force balance at a point in a continuum as

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} + b_x = 0 \\ \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} + b_y = 0 \\ \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + b_z = 0 \end{aligned} \tag{2.33}$$

where b_x, b_y, b_z are the components of body forces per unit volume in x, y and z directions. Equation (2.33) can also be written in a tensorial form as

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = 0 \quad i,\, j = 1,\, 2,\, 3. \tag{2.34}$$

The relations between strain and displacement are given by the **equations of compatibility** for infinitesimal strain as [128]

$$\begin{aligned} \varepsilon_{xx} &= \frac{\partial u}{\partial x}, \quad \gamma_{yz} = \frac{\partial w}{\partial y} + \frac{\partial v}{\partial z} \\ \varepsilon_{yy} &= \frac{\partial v}{\partial y}, \quad \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \\ \varepsilon_{zz} &= \frac{\partial w}{\partial z}, \quad \gamma_{xy} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \end{aligned} \tag{2.35}$$

and the corresponding tensorial notation is expressed as

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.36}$$

$$\gamma_{ij} = 2\varepsilon_{ij} \quad (i \neq j) \tag{2.37}$$

The elastic **equation of constitutive relation** describes the deformation with respect to the stress as

$$\begin{aligned} \varepsilon_{xx} &= \frac{1}{E}[\sigma_{xx} - \nu(\sigma_{yy} + \sigma_{zz})], \quad \gamma_{yz} = \frac{2(1+\nu)}{E}\tau_{yz} \\ \varepsilon_{yy} &= \frac{1}{E}[\sigma_{yy} - \nu(\sigma_{xx} + \sigma_{zz})], \quad \gamma_{xz} = \frac{2(1+\nu)}{E}\tau_{xz} \\ \varepsilon_{zz} &= \frac{1}{E}[\sigma_{zz} - \nu(\sigma_{xx} + \sigma_{yy})], \quad \gamma_{yz} = \frac{2(1+\nu)}{E}\tau_{yz} \end{aligned} \tag{2.38}$$

or in a tensorial notation as

$$\varepsilon_{ij} = \frac{1}{E}[(1+\nu)\sigma_{ij} - \nu\sigma_{kk}\delta_{ij}] \ , \tag{2.39}$$

where E is the Young's Modulus of the material and ν is the Poisson's ratio. Conversely, the stresses can be stated in terms of strains by

$$\begin{aligned} \sigma_{xx} &= \frac{2\mu\nu}{1-2\nu}\Theta + 2\mu\varepsilon_{xx}, \quad \sigma_{yz} = \mu\gamma_{yz} \\ \sigma_{yy} &= \frac{2\mu\nu}{1-2\nu}\Theta + 2\mu\varepsilon_{yy}, \quad \sigma_{xz} = \mu\gamma_{xz} \\ \sigma_{zz} &= \frac{2\mu\nu}{1-2\nu}\Theta + 2\mu\varepsilon_{zz}, \quad \sigma_{xy} = \mu\gamma_{xy} \end{aligned} \tag{2.40}$$

or in a tensorial form as

$$\sigma_{ij} = \frac{2\mu\nu}{1-2\nu}\delta_{ij}\varepsilon_{mm} + 2\mu\varepsilon_{ij} \tag{2.41}$$

where Θ is the volumetric strain

$$\varepsilon_{mm} = \Theta = \varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz} \ . \tag{2.42}$$

Here μ is the shear modulus of the material given by

$$\mu = \frac{E}{2(1+\nu)} \ . \tag{2.43}$$

Equations (2.33) to (2.39) can be summarized into

$$\begin{aligned} \sigma_{ij,j} + b_i &= 0 \quad \text{in } \Omega \\ \sigma_{ij} n_j &= \bar{t_i} \quad \text{on } \Gamma_t \\ u_i &= \bar{u}_i \quad \text{on } \Gamma_u \end{aligned} \tag{2.44}$$

where Ω is the domain under consideration, Γ_t is the part of the boundary of Ω subjected to known tractions $\bar{t_i}$ and Γ_u is the part of the boundary of Ω with prescribed displacement $\bar{u}_i$. The boundary of Ω, i.e. $\partial\Omega$ is denoted as Γ and it is normally assumed $\Gamma = \Gamma_t \cup \Gamma_u$. Equation (2.44) is the strong forms of elasticity.

2.3.2 Weak form solution

The equilibrium equations in (2.33) can regarded as the field $\mathbf{u}(\mathbf{x})$ conforming to a set of PDEs

$$\mathbf{A}(\mathbf{u}) = \left\{ \begin{array}{c} A_1(\mathbf{u}) \\ A_2(\mathbf{u}) \\ \vdots \end{array} \right\} = \mathbf{0} \quad \text{in } \Omega \ , \tag{2.45}$$

which satisfy boundary conditions as

$$\mathbf{B}(\mathbf{u}) = \left\{ \begin{array}{c} B_1(\mathbf{u}) \\ B_2(\mathbf{u}) \\ \vdots \end{array} \right\} = \mathbf{0} \quad \text{on } \partial\Omega \ , \tag{2.46}$$

where $\mathbf{A}(\cdot)$ and $\mathbf{B}(\cdot)$ are vectors of partial differential operations. A strong form requires $\mathbf{u}(\mathbf{x})$ to satisfy equilibrium everywhere inside the domain and also to conform to the prescribed boundary conditions, while a weak form only requires a conformity in an integral sense. There are two main approaches to arrive at weak forms, firstly

the weighted residual method (WRM) and secondly finding an equivalent functional of the governing PDEs applied with the variational principle.

In the first approach using the WRM, the governing PDEs are written into an integration over the problem domain as

$$\int_{\Omega} \mathbf{v}(\mathbf{x}) \cdot \mathbf{A}(\mathbf{u}) + \int_{\Gamma} \bar{\mathbf{v}}(\mathbf{x}) \cdot \mathbf{B}(\mathbf{u}) = 0 \tag{2.47}$$

where $\mathbf{v}(\mathbf{x}) \in H^1$ and $\bar{\mathbf{v}}(\mathbf{x}) \in H^0$ are vectors of *test functions* and H^1 and H^0 are Sobolev spaces. Here a Sobolev space is a vector space of functions equipped with L_p norms for the functions and their derivatives up to a given order. The term *test functions* refers to a set of functions used to test the satisfaction of governing PDEs at any point inside the domain. If Equation (2.47) is valid for an arbitrary choice of $\mathbf{v}(\mathbf{x})$ and $\bar{\mathbf{v}}(\mathbf{x})$ from a complete function space, it can be proved the solutions to the weak form are equivalent to those of the strong form. The unknown field $\mathbf{u}(\mathbf{x})$ can be numerically approximated by expansions or polynomials which is termed the *trial functions* noted as $\mathbf{u}^h(\mathbf{x})$, and $\mathbf{u}^h(\mathbf{x}) \approx u(\mathbf{x})$. Approximate solutions to Equation (2.47) can be obtained by substituting a set of discretized points into (2.47). Practically, several methods are used in the WRM, namely the least squares collocation method, point collocation method, force moment method, subdomain method and Galerkin method. They vary by each other in the way of minimizing the residuals and choice of test functions.

In the second approach, the Rayleigh-Ritz method is commonly used based on a variational principle. The functional corresponding to the governing PDEs of the problem domain are firstly found and then its minimum is approximated by a linear combination of functions taken from a function space. In elasticity, the functional corresponding to the strong form can be derived from minimum potential energy or minimum residual energy, which are not detailed here. Interested readers can refer to [129] and [130].

System of algebraic equations can be finally obtained using either approach which are essentially the same. In both approaches, the Galerkin method can be used if by setting $\mathbf{v} = \mathbf{u}$ and $\bar{\mathbf{v}} = -\bar{\mathbf{u}}$ along boundaries (when test functions satisfy the Kronecker delta property). For example, in the first approach Equation (2.47)

becomes

$$\int_{\Omega} \mathbf{u}(\mathbf{x}) \cdot \mathbf{A}(\mathbf{u}) - \int_{\Gamma} \bar{\mathbf{u}}(\mathbf{x}) \cdot \mathbf{B}(\mathbf{u}) = \mathbf{0} \ . \tag{2.48}$$

We now clarify some concepts with the Galerkin approach and a variational principle. The Galerkin method is a way to approximate the field by enforcing the test functions $\mathbf{v}$ to be the same as the function space of the trial function. This can be used in both approaches, the WRM or the Rayleigh-Ritz method to arrive at the discrete forms. The variational principle is often associated with the functional in the second kind of approach. It can also be used in the first kind of approach to derive discrete form. For example, applying variation to $\mathbf{u}(\mathbf{x})$ in Equation (2.48), the weak form using the first approach can be obtained as

$$\int_{\Omega} \delta\mathbf{u}(\mathbf{x}) \cdot \mathbf{A}(\mathbf{u}) - \int_{\Gamma} \delta\bar{\mathbf{u}}(\mathbf{x}) \cdot \mathbf{B}(\mathbf{u}) = \mathbf{0} \ . \tag{2.49}$$

Weak form obtained by the first approach is equivalent to that by the second approach if there exists a general functional for the physical problem to be solved. In the following section, we will derive the weak forms from minimum potential energy and using a variational principle for the modified weak forms using the penalty method and the Lagrange multiplier method. The Nitsche-like method is derived from the governing equation directly falling into the first kind of approach and the corresponding discrete form is derived using variational principle.

2.3.3 Constrained Hamilton's principle

In this section, we will firstly look at Hamilton's principle which is a variational principle in mechanics to determine an equivalent single functional of a system. The constrained Hamilton's principle is later introduced to derive modified weak forms which can deal with essential boundary conditions using a meshless approximation. In continuum mechanics, the system energy of a solid can be defined as

$$L = T - \Pi_s + W_f \tag{2.50}$$

where T is the kinetic energy for dynamic problems, Π_s is the internal work, and W_f is the work done by external forces. The system energy L is a functional in T, W_f, and Π_s. Hamilton's principle is a variational principle which states "of all possible time histories of consistency which satisfies

1. the compatibility conditions

2. the essential boundary conditions (prescribed displacement)

3. the conditions at initial time t_1 and final t_2,

the history corresponding to the actual solutions makes the functional (L) a minimum". The mathematical expression of Hamilton's principle is

$$\delta \int_{t_1}^{t_2} L \mathrm{d}t = 0 \ . \tag{2.51}$$

The variational symbol δ implies the introduction a small perturbation to the system, which is also known as the *admissible conditions*. Since statics problems are considered here, T becomes zero. For elastic materials, the internal work is calculated by

$$\Pi_s = \frac{1}{2} \int_\Omega \boldsymbol{\varepsilon} \cdot \boldsymbol{\sigma} \mathrm{d}\Omega \tag{2.52}$$

where Ω is the problem domain, $\boldsymbol{\varepsilon}$ is the infinitesimal strain tensor and $\boldsymbol{\sigma}$ is the stress tensor. The external work can be expressed as

$$W_f = \int_\Omega \mathbf{u} \cdot \mathbf{b} \mathrm{d}\Omega + \int_{\Gamma_t} \mathbf{u} \cdot \bar{\mathbf{t}} \mathrm{d}\Omega \tag{2.53}$$

where $\mathbf{b}$ is the body force and Γ_t is part of the boundary of Ω with prescribed tractions $\bar{\mathbf{t}}$. For approximations with the Kronecker delta property, such as the FEM, applying Hamilton's principle is simple. The weak form as given in Equation (2.51) does not contain the information along essential boundaries and in the FEM the prescribed displacements there are directly assigned to nodes. This is however not the case with meshless methods where the approximation solution does not satisfy the requirement of Hamilton's principle due to the lack of the Kronecker delta property of the shape functions. Direct imposition of displacement on part of the boundary results in unwanted values on other nodes the boundary or inside the domain. A similar case is also met in surface fitting using MLS approximation that it is impossible to directly prescribe curvature values for several points at the same time as the values between points are co-related. For this reason Hamilton's principle

is modified to include additional constraints from essential boundary conditions. Consider there are a set of k conditions that the approximation fails to satisfy

$$\mathbf{C}(\mathbf{u}) = \left\{ \begin{array}{c} C_1(\mathbf{u}) \\ C_2(\mathbf{u}) \\ \vdots \\ C_k(\mathbf{u}) \end{array} \right\} = \mathbf{0} \tag{2.54}$$

where $\mathbf{C}$ is a given matrix of differential operators required by consistency or compatibility conditions. The purpose is to find the stationarity of functional L subjected also to the constraints of Equation 2.54. There are basically two methods used to accommodate these constraints: the penalty method and the Lagrange multiplier method.

2.3.4 Penalty method

The penalty method gives a constrained Hamilton's principle in the form of

$$\tilde{L} = L + \frac{1}{2}\int_{\Omega} \mathbf{C}(\mathbf{u}) \cdot \boldsymbol{\beta} \cdot \mathbf{C}(\mathbf{u}) \mathrm{d}\Omega \tag{2.55}$$

where $\boldsymbol{\beta}$ is a diagonal matrix with entries of known constant *penalty factors*

$$\boldsymbol{\beta} = \mathrm{diag}\left\{\beta_1, \beta_2, \ldots, \beta_n\right\} \tag{2.56}$$

and the number of degrees of freedom in the discretisation unchanged. It is usually assumed all components of $\boldsymbol{\beta}$ take the same value, noted as β. Therefore $\boldsymbol{\beta}$ can be taken outside the integration and be replaced by β

$$\tilde{L} = L + \frac{1}{2}\beta\int_{\Omega} \mathbf{C}(\mathbf{u}) \cdot \mathbf{C}(\mathbf{u}) \mathrm{d}\Omega \ . \tag{2.57}$$

Applying Hamilton's principle to Equation (2.57) leads to

$$\delta\int_{t_1}^{t_2} \tilde{L} \mathrm{d}t = \int_{t_1}^{t_2} \left(\delta L + \frac{1}{2}\beta\int_{\Omega} \delta(\mathbf{C}(\mathbf{u}) \cdot \mathbf{C}(\mathbf{u})) \mathrm{d}\Omega \right) \mathrm{d}t \ . \tag{2.58}$$

Then applying the variation using the chain rule, we get

$$\delta(\mathbf{C}\mathbf{u} \cdot \mathbf{C}(\mathbf{u})) = 2\delta\mathbf{C}(\mathbf{u}) \cdot \mathbf{C}(\mathbf{u}) \ , \tag{2.59}$$

and Equation (2.58) can be written as

$$\int_{t_1}^{t_2} \delta\tilde{L}\mathrm{d}t = \int_{t_1}^{t_2} \left(\delta L + \beta \int_{\Omega} \delta\mathbf{C}(\mathbf{u}) \cdot \mathbf{C}(\mathbf{u})\mathrm{d}\Omega \right) \mathrm{d}t \ . \tag{2.60}$$

Equation (2.60) is valid for arbitrary times from t_1 to t_2, so for statics problems the time integration vanishes, i.e.

$$\delta\tilde{L} = \delta L + \beta \int_{\Omega} \delta\mathbf{C}(\mathbf{u}) \cdot \mathbf{C}(\mathbf{u})\mathrm{d}\Omega \ . \tag{2.61}$$

Here the constraints along the essential boundary are

$$\mathbf{C}(\mathbf{u}) = \mathbf{u} - \bar{\mathbf{u}} \tag{2.62}$$

where $\bar{\mathbf{u}}$ is prescribed displacement at the essential boundary. By substituting Equations (2.52), (2.53) and (2.62) into (2.61), the weak form with penalty factors is obtained as

$$\int_{\Omega} \delta\boldsymbol{\varepsilon} \cdot \boldsymbol{\sigma} \, \mathrm{d}\Omega - \beta \int_{\Gamma_u} \delta\mathbf{u} \cdot (\mathbf{u} - \bar{\mathbf{u}}) \, \mathrm{d}\Gamma = \int_{\Omega} \delta\mathbf{u} \cdot \mathbf{b} \, \mathrm{d}\Omega + \int_{\Gamma_t} \delta\mathbf{u} \cdot \bar{\mathbf{t}} \, \mathrm{d}\Gamma \ . \tag{2.63}$$

2.3.5 Lagrange multiplier method

A constrained Hamilton's principle can also be formed using the Lagrange Multiplier method, which is written as

$$\tilde{L} = L + \int_{\Omega} \boldsymbol{\lambda} \cdot \mathbf{C}(\mathbf{u})\mathrm{d}\Omega \tag{2.64}$$

where $\boldsymbol{\lambda}$ is a vector of scalars,

$$\boldsymbol{\lambda}^T = \{\lambda_1, \lambda_2, \cdots, \lambda_k\} \tag{2.65}$$

The Lagrange multipliers $\boldsymbol{\lambda}$ are unknowns along the parts of the boundaries or in the domain which require enforced consistency or compatibility conditions. Therefore, the numbers of unknowns of the whole system is increased. The Lagrange multipliers should conform with the PDE governing the problem domain and the conditions are rigorously enforced, whilst the penalty method is only an approximation of conditions to be enforced. Using a similar process to the derivation of the weak

form with the penalty method in §2.3.4, the weak form with Lagrange multipliers can be derived as

$$\begin{aligned}&\int_\Omega \delta\boldsymbol{\varepsilon}\cdot\boldsymbol{\sigma}\mathrm{d}\Omega - \int_{\Gamma_u}\delta\boldsymbol{\lambda}\cdot(\mathbf{u}-\bar{\mathbf{u}})\,\mathrm{d}\Gamma - \int_{\Gamma_u}\delta\mathbf{u}\cdot\boldsymbol{\lambda}\,\mathrm{d}\Gamma\\ &= \int_\Omega \delta\mathbf{u}\cdot\mathbf{b}\,\mathrm{d}\Omega + \int_{\Gamma_t}\delta\mathbf{u}\cdot\bar{\mathbf{t}}\,\mathrm{d}\Gamma\end{aligned} \tag{2.66}$$

where

$$\lambda(\mathbf{x}) = N_I^\lambda(s)\lambda_I, \quad \mathbf{x}\in\Gamma_u$$
$$\delta\lambda(\mathbf{x}) = N_I^\lambda(s)\delta\lambda_I, \quad \mathbf{x}\in\Gamma_u$$

and $N_I^\lambda(s)$ are interpolation shape function for the Lagrange multipliers. We can use the same shape functions for $N_I(s)$ as for the displacement approximation. $N_I(s)$ can also be constructed by Lagrangian interpolation (see §2.4.4) even with a meshless method.

2.3.6 The Nitsche-like method

The above described weak forms by using the penalty method and the Lagrange multiplier method are used most commonly in meshless methods [122]. However, there are some drawbacks such as ill-conditioning with both and extra unknowns in the global stiffness matrix in the latter. Nitsche's method was proposed in [131] which eliminates those problems, however the determination of the weak form is not as straightforward. Here a Nitsche-like weak form is presented for elastostatics. To understand Nitsche's method for elasticity, it is first necessary to start from a different elliptic problem for which the method was originally explained, namely Poisson's equations in 1D as follows

$$-\Delta u = f \quad \text{in } \Omega \tag{2.67a}$$
$$u = \bar{u} \quad \text{on } \Gamma_u \tag{2.67b}$$
$$\nabla u\cdot\mathbf{n} = \bar{t} \quad \text{on } \Gamma_t \tag{2.67c}$$

where $\Delta = \nabla\cdot\nabla = \nabla^2$, u is the problem field over the domain Ω, $\partial\Omega$ is the boundary of Ω, the union of essential boundaries and natural boundaries, i.e. $\Gamma_u\cup\Gamma_n = \partial\Omega$

and $\mathbf{n}$ is the unit outer normal along the boundary. The term $\bar{u}$ denotes prescribed value of u along essential boundaries Γ_u and $\bar{t}$ is the flux of u in the direction of $\mathbf{n}$. The weak form associated with PDEs (2.67) is to find $u \in H^1$ such that $u = \bar{u}$ on Γ_u and satisfies

$$\int_\Omega -v\Delta u \, \mathrm{d}\Omega = \int_\Omega vf \, \mathrm{d}\Omega \tag{2.68}$$

for $\forall v \in H_0^1$, where v is the test function. Using integration by parts and noting that

$$\int_\Omega \nabla \left(v\nabla u\right) \, \mathrm{d}\Omega = \int_{\partial\Omega} v\nabla u \cdot \mathbf{n} \, \mathrm{d}\Gamma \ , \tag{2.69}$$

then the term on the *l.h.s.* of equation (2.68) can be written as

$$\int_\Omega -v\Delta u \, \mathrm{d}\Omega = \int_\Omega \nabla v \nabla u \, \mathrm{d}\Omega - \int_{\partial\Omega} v\nabla u \cdot \mathbf{n} \, \mathrm{d}\Gamma \ . \tag{2.70}$$

Substituting Equation (2.70) into (2.68) leads to

$$\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega - \int_{\partial\Omega} v\nabla u \cdot \mathbf{n} \, \mathrm{d}\Gamma = \int_\Omega vf \, \mathrm{d}\Omega \ . \tag{2.71}$$

Substituting Equation (2.67) into (2.71) and noting that $\Gamma_u \cup \Gamma_t = \partial\Omega$, the modified weak form can be obtained as

$$\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega - \int_{\Gamma_u} v\nabla u \cdot \mathbf{n} \, \mathrm{d}\Gamma = \int_\Omega vf \, \mathrm{d}\Omega + \int_{\Gamma_t} v\bar{t} \, \mathrm{d}\Gamma \ . \tag{2.72}$$

If $v = 0$ on Γ_u, which is the case in the FEM, then Equation (2.72) becomes

$$\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega = \int_\Omega vf \mathrm{d}\Omega + \int_{\Gamma_t} v\bar{t} \, \mathrm{d}\Gamma \ . \tag{2.73}$$

However, Equation (2.73) is not applicable for meshless methods such as the EFGM due to the lack of the Kronecker delta property. In Nitsche's method, the weak form includes flux terms along the essential boundary and takes the bilinear symmetric form of trial function u and test function v. So firstly the term on Γ_u in Equation (2.72) is preserved, and two additional terms on Γ_u are added

$$\begin{aligned} &\int_\Omega \nabla v \cdot \nabla u \, \mathrm{d}\Omega - \int_{\Gamma_u} v\nabla u \cdot \mathbf{n} \, \mathrm{d}\Gamma - \int_{\Gamma_u} u\nabla v \cdot \mathbf{n} \, \mathrm{d}\Gamma - \beta \int_{\Gamma_u} vu \, \mathrm{d}\Gamma \\ &= \int_\Omega vf \, \mathrm{d}\Omega + \int_{\Gamma_t} v\bar{t} \, \mathrm{d}\Gamma - \int_{\Gamma_u} \bar{u}\nabla v \cdot \mathbf{n} \, \mathrm{d}\Gamma - \beta \int_{\Gamma_u} v\bar{u} \, \mathrm{d}\Gamma \ . \end{aligned} \tag{2.74}$$

After rearrangement, Equation (2.74) becomes

$$\begin{aligned}&\int_\Omega \nabla v \cdot \nabla u \,\mathrm{d}\Omega - \int_{\Gamma_u} v \nabla u \cdot \mathbf{n} \,\mathrm{d}\Gamma - \int_{\Gamma_u} (u - \bar{u}) \nabla v \cdot \mathbf{n} \,\mathrm{d}\Gamma \\ &= \beta \int_{\Gamma_u} v \,(u - \bar{u}) \,\mathrm{d}\Gamma + \int_\Omega v f \,\mathrm{d}\Omega + \int_{\Gamma_t} v \bar{t} \,\mathrm{d}\Gamma \ . \end{aligned} \tag{2.75}$$

Compared with Equation (2.73), there are three additional terms; two flux terms and a penalty term. The difficulty in using Nitsche's method directly for elasticity is that the PDEs based on displacement are not in a form Nitsche originally addressed. In solid mechanics, the flux along boundaries corresponds to the force or traction, because the units of each term in the weak form should be energetic. Physically, the terms with flux are the work produced by tractions along essential boundaries. This can be proved by the generalized variational principle in [130]. With this finding and noting that $v = u$ in a Galerkin approach, we can write a Nitsche-like weak form for elasticity problems, working now with vector quantities as

$$\begin{aligned}&\int_\Omega \boldsymbol{\varepsilon} \cdot \boldsymbol{\sigma} \,\mathrm{d}\Omega - \int_{\Gamma_u} \mathbf{u} \cdot (\boldsymbol{\sigma} \cdot \mathbf{n}) \,\mathrm{d}\Gamma - \int_{\Gamma_u} (\boldsymbol{\sigma} \cdot \mathbf{n}) \cdot (\mathbf{u} - \bar{\mathbf{u}}) \,\mathrm{d}\Gamma \\ &= \beta \int_{\Gamma_u} \mathbf{u} \cdot (\mathbf{u} - \bar{\mathbf{u}}) \,\mathrm{d}\Gamma + \int_\Omega \mathbf{u} \cdot \mathbf{b} \,\mathrm{d}\Omega + \int_{\Gamma_t} \mathbf{u} \cdot \bar{\mathbf{t}} \,\mathrm{d}\Gamma \ . \end{aligned} \tag{2.76}$$

By applying the variational principle to Equation (2.76) and noting that variation is applied to $\mathbf{u}$, then we get a Nitsche-like weak form as

$$\begin{aligned}&\int_\Omega \delta\boldsymbol{\varepsilon} \cdot \boldsymbol{\sigma} \,\mathrm{d}\Omega - \int_{\Gamma_u} \delta\mathbf{t} \cdot (\mathbf{u} - \bar{\mathbf{u}}) \,\mathrm{d}\Gamma - \int_{\Gamma_u} \delta\mathbf{u} \cdot \mathbf{t} \,\mathrm{d}\Gamma \\ &= \beta \int_{\Gamma_u} \delta\mathbf{u} \cdot (\mathbf{u} - \bar{\mathbf{u}}) \,\mathrm{d}\Gamma + \int_\Omega \delta\mathbf{u} \cdot \mathbf{b} \,\mathrm{d}\Omega + \int_{\Gamma_t} \delta\mathbf{u} \cdot \bar{\mathbf{t}} \,\mathrm{d}\Gamma \ . \end{aligned} \tag{2.77}$$

2.3.7 Relations between weak forms

The major difference between the penalty method and the Nitsche-like method is that the latter takes the flux term (corresponding to work in mechanics) along essential boundaries into consideration and adds extra terms to make the symmetric bilinear form, so the inconsistency problem in the basic penalty method is overcome. Another way to understand this Nitsche-like method is to start from the Lagrange multiplier method and recognize that the unknown multipliers λ in Equation (2.66) along essential boundaries are tractions, i.e.

$$\lambda = \mathbf{t} \quad \text{on} \quad \Gamma_u \ , \tag{2.78}$$

and add a penalty term to the *r.h.s* of Equation (2.66) to give (2.77). In [132] a modified weak form, based on the Hu-Washizu variational principle, is used in the EFGM which is similar to the Nitsche-like weak form. The only difference between them is the absence of the penalty term in the former. To conclude this section, we give a relation between the weak forms in Figure 2.11.

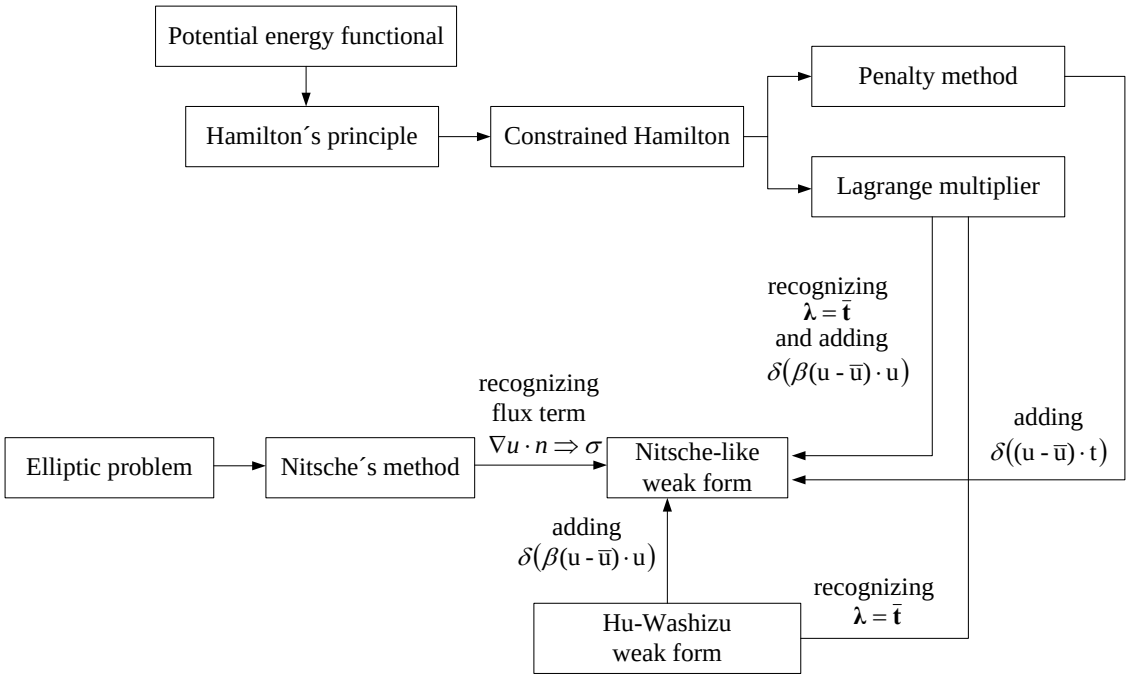


Figure 2.11: Relations between the modified weak forms described here.

2.4 Discretisation for linear elasticity

In this section, we discretise these weak forms for elastostatic problems. A detailed discretisation of the weak form is firstly given for the Nitsche-like method. For weak forms by the penalty and the Lagrange multiplier methods, the discretisation processes are similar and any differences will be highlighted.

2.4.1 Matrix forms

In the following derivation, matrix notation is used since it is clear and directly corresponds to programming requirements. The governing equations of elastotstatics,

namely (2.33), (2.35) and (2.39) can be expressed in matrix form as follows. The compatibility equations are expressed in matrix form as

$$\boldsymbol{\varepsilon} = \mathbf{L}\mathbf{u} \ , \tag{2.79}$$

and the constitutive equations can be expressed as

$$\boldsymbol{\sigma} = \mathbf{D}\mathbf{L}\mathbf{u} \tag{2.80}$$

where $\mathbf{L}$ is a matrix of differential operators which in 3D is,

$$\mathbf{L} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 & 0 \\ 0 & \dfrac{\partial}{\partial y} & 0 \\ 0 & 0 & \dfrac{\partial}{\partial z} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial z} & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial z} & 0 & \dfrac{\partial}{\partial x} \end{bmatrix} \ , \tag{2.81}$$

$\mathbf{D}$ is the constitutive matrix for an isotropic linear elastic material

$$\mathbf{D} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \dfrac{\nu}{1-\nu} & \dfrac{\nu}{1-\nu} & 0 & 0 & 0 \\ \dfrac{\nu}{1-\nu} & 1 & \dfrac{\nu}{1-\nu} & 0 & 0 & 0 \\ \dfrac{\nu}{1-\nu} & \dfrac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \dfrac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dfrac{1-2\nu}{2(1-\nu)} \end{bmatrix} . \tag{2.82}$$

Substituting Equations from (2.26) to (2.81) into (2.44), it can be written as

$$\begin{aligned} \mathbf{L}^{\mathbf{T}}\mathbf{D}\left(\mathbf{L}\mathbf{u}\right) + \mathbf{b} &= 0 \quad \text{in } \Omega \\ \mathbf{u} &= \bar{\mathbf{u}} \quad \text{on } \Gamma_t \\ \mathbf{D}\left(\mathbf{L}\mathbf{u}\right)\mathbf{n} &= \bar{\mathbf{t}} \quad \text{on } \Gamma_u \ . \end{aligned} \tag{2.83a}$$

Here, $\mathbf{n}$ is a matrix of the outer normals of a point lying on the boundary

$$\mathbf{n} = \begin{bmatrix} n_x & 0 & 0 \\ 0 & n_y & 0 \\ 0 & 0 & n_z \\ n_y & n_x & 0 \\ 0 & n_z & n_y \\ n_z & 0 & n_x \end{bmatrix} \tag{2.84}$$

and $\mathbf{b}$ is the body force vector

$$\mathbf{b} = \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix} . \tag{2.85}$$

Now the approximation to $\mathbf{u}$, $\mathbf{u}^h$ is used. The displacement component $u^h(x)$ of $\mathbf{u}^h$ is

$$u^h(\mathbf{x}) = \sum_I^n N_I(\mathbf{x}) u_I \ , \tag{2.86}$$

displacement component $v^h(x)$ is

$$v^h(\mathbf{x}) = \sum_I^n N_I(\mathbf{x}) v_I \ , \tag{2.87}$$

and displacement component $w^h(x)$ is

$$w^h(\mathbf{x}) = \sum_I^n N_I(\mathbf{x}) w_I \ . \tag{2.88}$$

Combining these three components together, we have

$$\mathbf{u}^h(\mathbf{x}) = \begin{Bmatrix} u^h \\ v^h \\ w^h \end{Bmatrix} = \sum_I^n \underbrace{\begin{bmatrix} N_I & 0 & 0 \\ 0 & N_I & 0 \\ 0 & 0 & N_I \end{bmatrix}}_{N_I} \underbrace{\begin{Bmatrix} u_I \\ v_I \\ w_I \end{Bmatrix}}_{\mathbf{u}_I} = \sum_I^n \mathbf{N}_I \mathbf{u}_I \ , \tag{2.89}$$

where n is the number of the nodes in support the point of interest $\mathbf{x}$, N_I is the shape function, e.g. EFGM shape function, of a supporting node I. By substituting

the displacement interpolation with equation (2.89) for strain product $\mathbf{Lu^h}$, we have

$$\mathbf{Lu^h}(\mathbf{x}) = \sum_I^n \mathbf{L}\mathbf{N}_I\mathbf{u}_I = \sum_I^n \begin{bmatrix} N_{I,x} & 0 & 0 \\ 0 & N_{I,y} & 0 \\ 0 & 0 & N_{I,z} \\ N_{I,y} & N_{I,x} & 0 \\ 0 & N_{I,y} & N_{I,z} \\ N_{I,z} & 0 & N_{I,x} \end{bmatrix} \begin{Bmatrix} u_I \\ v_I \\ w_I \end{Bmatrix} = \sum_I^n \mathbf{B}_I\mathbf{u}_I \tag{2.90}$$

where $N_{I,x}$, $N_{I,y}$ and $N_{I,z}$ are the derivatives of the shape functions with respect to the coordinates directions, and $\mathbf{B}_I$ is the strain-displacement matrix for node I.

2.4.2 The Nitsche-like method

We will focus on the discretisation of the Nitsche-like weak from in §2.3.6. Applying the variational principle to Equation (2.76) and replacing stress and strain with (2.79) and (2.80) gives

$$\begin{aligned} &\int_\Omega \delta(\mathbf{Lu})^T(\mathbf{DLu})\,\mathrm{d}\Omega - \int_{\Gamma_u} \delta\mathbf{u}^T\mathbf{n}^T(\mathbf{DLu})\,\mathrm{d}\Gamma - \int_{\Gamma_u} \delta(\mathbf{DLu})^T\mathbf{n}(\mathbf{u}-\bar{\mathbf{u}})\,\mathrm{d}\Gamma \\ &= \beta\int_{\Gamma_u} \delta\mathbf{u}^T(\mathbf{u}-\bar{\mathbf{u}})\,\mathrm{d}\Gamma + \int_\Omega \delta\mathbf{u}^T\mathbf{b}\mathrm{d}\Omega + \int_{\Gamma_t} \delta\mathbf{u}^T\bar{\mathbf{t}}\mathrm{d}\Gamma\ . \end{aligned} \tag{2.91}$$

Substituting Equation (2.90) and Equation (2.89)into Equation (2.91), we have

$$\begin{aligned} &\int_\Omega \delta\left(\sum_I^n \mathbf{B}_I\mathbf{u}_I\right)^T\left(\mathbf{D}\sum_J^n \mathbf{B}_J\mathbf{u}_J\right)\mathrm{d}\Omega - \int_{\Gamma_u} \delta\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)^T\mathbf{n}^T\left(\mathbf{D}\sum_J^n \mathbf{B}_J\mathbf{u}_J\right)\mathrm{d}\Gamma \\ &- \int_{\Gamma_u} \delta\left(\mathbf{D}\sum_I^n \mathbf{B}_I\mathbf{u}_I\right)^T\mathbf{n}\left(\sum_J^n \mathbf{N}_J\mathbf{u}_J - \bar{\mathbf{u}}\right)\mathrm{d}\Gamma \\ &= \beta\int_{\Gamma_u} \delta\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)^T\left(\sum_J^n \mathbf{N}_J\mathbf{u}_J - \bar{\mathbf{u}}\right)\mathrm{d}\Gamma + \int_\Omega \delta\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)^T\mathbf{b}\mathrm{d}\Omega \\ &+ \int_{\Gamma_t} \delta\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)^T\bar{\mathbf{t}}\mathrm{d}\Gamma\ . \end{aligned} \tag{2.92}$$

Note that the summation indices in the first term on the *l.h.s.* differ for convenience of later derivation. Considering the first term on the *l.h.s.* and transposing the first bracket term we have

$$\int_\Omega \delta\left(\sum_I^n \mathbf{B}_I\mathbf{u}_I\right)^T\left(\mathbf{D}\sum_J^n \mathbf{B}_J\mathbf{u}_J\right)\mathrm{d}\Omega = \int_\Omega \delta\left(\sum_I^n {\mathbf{u}_I}^T{\mathbf{B}_I}^T\right)\left(\mathbf{D}\sum_J^n \mathbf{B}_J\mathbf{u}_J\right)\mathrm{d}\Omega\,. \tag{2.93}$$

Note the summation, variation and integral are all linear operators and we can change their orders to get

$$\int_\Omega \delta \left(\sum_I^n \mathbf{u}_I{}^T \mathbf{B}_I{}^T \right) \left(\mathbf{D} \sum_J^n \mathbf{B}_J \mathbf{u}_J \right) \mathrm{d}\Omega = \sum_I^{n_t} \sum_J^{n_t} \delta \mathbf{u}_I{}^T \underbrace{\int_\Omega \mathbf{B}_I{}^T \mathbf{D} \mathbf{u}_J \mathrm{d}\Omega}_{\mathbf{K}_{IJ}} \mathbf{u}_J$$
$$= \sum_I^{n_t} \sum_J^{n_t} \delta \mathbf{u}_I{}^T \mathbf{K}_{IJ} \mathbf{u}_J \ . \tag{2.94}$$

Since in Equation (2.94) the integration is performed over the whole domain Ω, the summation in Equation (2.94) will involve all nodes.

$$\begin{aligned} \sum_I^{n_t} \sum_J^{n_t} \delta \mathbf{u}_I{}^T \mathbf{K}_{IJ} \mathbf{u}_J = {} & \delta \mathbf{u}_1^T \mathbf{K}_{11} \mathbf{u}_1 + \delta \mathbf{u}_1^T \mathbf{K}_{12} \mathbf{u}_2 + \cdots + \delta \mathbf{u}_1^T \mathbf{K}_{1n_t} \mathbf{u}_{n_t} \\ & + \delta \mathbf{u}_2^T \mathbf{K}_{21} \mathbf{u}_1 + \delta \mathbf{u}_2^T \mathbf{K}_{22} \mathbf{u}_2 + \cdots + \delta \mathbf{u}_2^T \mathbf{K}_{2n_t} \mathbf{u}_{n_t} \\ & \vdots \qquad \vdots \qquad \vdots \\ & + \delta \mathbf{K}_{n_t} \mathbf{u}_{n_t 1}{}^T \mathbf{u}_1 + \delta \mathbf{u}_{n_t}{}^T \mathbf{K}_{n_t 2} \mathbf{u}_2 + \cdots + \delta \mathbf{u}_{n_t}^T \mathbf{K}_{n_t n_t} \mathbf{u}_{n_t} \\ = {} & \delta \mathbf{U}^T \mathbf{K} \mathbf{U} \end{aligned} \tag{2.95}$$

where $\mathbf{U} = \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_t}$ is a vector collecting the all the nodal displacements and $\mathbf{K}$ is composed of square matrices. For 3D problems each square matrix is 3×3 and the overall stiffness matrix $\mathbf{K}$ is $3n_t \times 3n_t$.

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \cdots & \mathbf{K}_{1n_t} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \cdots & \mathbf{K}_{2n_t} \\ \cdots & \cdots & \ddots & \cdots \\ \mathbf{K}_{n_t 1} & \mathbf{K}_{n_t 2} & \cdots & \mathbf{K}_{n_t n_t} \end{bmatrix} \tag{2.96}$$

where n_t is the total number of nodes used to discretise the problem domain. Note that the flux terms are the work done by tractions along the outer normal of the boundary. Therefore if the displacement boundaries are assigned locally, mapping is needed to the global coordinates of the boundary using the transposition vector

$$\mathbf{r} = \begin{Bmatrix} l_x \\ l_y \\ l_z \end{Bmatrix} \tag{2.97}$$

where l_x, l_y and l_z are the projections of the unit outer normal of a point lying on boundary to the global axes x, y and z respectively. If the order of operators in

Equation (2.92) is changed, we obtain

$$\begin{aligned}
&\int_{\Gamma_u} \delta \left(\sum_I^n \mathbf{N}_I \mathbf{u}_I \right)^T \mathbf{r} \left(\mathbf{r}^T \mathbf{n}^T \mathbf{D} \sum_J^n \mathbf{B}_J \mathbf{u}_J \right) \mathrm{d}\Gamma \\
&= \int_{\Gamma_u} \delta \sum_I^n \mathbf{u}_I^T \mathbf{N}_I^T \mathbf{R} \mathbf{n}^T \mathbf{D} \sum_J^n \mathbf{B}_J \mathbf{u}_J \mathrm{d}\Gamma \\
&= \sum_I^{n_t} \sum_J^{n_t} \delta \mathbf{u}_I^T \underbrace{\int_{\Gamma_u} \mathbf{N}_I^T \mathbf{R} \mathbf{n}^T \mathbf{D} \mathbf{B}_J \mathrm{d}\Gamma}_{-\mathbf{L}_{IJ}} \mathbf{u}_J
\end{aligned} \tag{2.98}$$

where $\mathbf{R} = \mathbf{r}^T \mathbf{r}$. The term $\int_{\Gamma_u} \mathbf{N}_I^T \mathbf{R} \mathbf{N} \mathbf{D} \mathbf{B}_J \, \mathrm{d}\Gamma$ as $-\mathbf{L}_{IJ}$ gives the work along the essential boundary. Similarly, the third term becomes

$$\begin{aligned}
&\int_{\Gamma_u} \delta \left(\sum_I^n \mathbf{D} \mathbf{B}_I \mathbf{u}_I \right)^T \mathbf{n} \mathbf{r} \left(\mathbf{r}^T \sum_J^n \mathbf{N}_J \mathbf{u}_J - \bar{\mathbf{u}} \right) \mathrm{d}\Gamma \\
&= \int_{\Gamma_u} \delta \sum_I^n \mathbf{B}_I^T \mathbf{u}_I^T \mathbf{D} \mathbf{n} \mathbf{R} \sum_J^n \mathbf{N}_J \mathbf{u}_J \mathrm{d}\Gamma - \textstyle\int_{\Gamma_u} \delta \sum_I^n \mathbf{u}_I^T \mathbf{B}_I^T \mathbf{D} \mathbf{n} \mathbf{r} \bar{\mathbf{u}} \mathrm{d}\Gamma \\
&= \sum_I^{n_t} \sum_J^{n_t} \delta \mathbf{u}_I^T \underbrace{\int_{\Gamma_u} \mathbf{B}_I^T \mathbf{D} \mathbf{n} \mathbf{R} \mathbf{N}_J \mathrm{d}\Gamma}_{-\mathbf{G}_{IJ}} \mathbf{u}_J - \sum_I^{n_t} \delta \mathbf{u}_I^T \underbrace{\int_{\Gamma_u} \mathbf{B}_I^T \mathbf{D} \mathbf{n} \mathbf{r} \bar{\mathbf{u}} \mathrm{d}\Gamma}_{-\mathbf{P}_I} \\
&= -\delta \mathbf{U}^T \mathbf{G} \mathbf{U} + \delta \mathbf{U}^T \mathbf{P}
\end{aligned} \tag{2.99}$$

where the term $\int_{\Gamma_u} \mathbf{B}_I^T \mathbf{D}^T \mathbf{N}^T \mathbf{R} \mathbf{N}_J \mathrm{d}\Gamma$ is noted as $-\mathbf{G}_{IJ}$. The penalty term becomes

$$\begin{aligned}
&\beta \int_{\Gamma_u} \delta \left(\mathbf{r}^T \sum_I^n \mathbf{N}_I \mathbf{u}_I \right)^T \left(\mathbf{r}^T \sum_J^n \mathbf{N}_J \mathbf{u}_J - \bar{\mathbf{u}} \right) \\
&= \beta \sum_I^{n_t} \sum_J^{n_t} \delta \mathbf{u}_I^T \underbrace{\int_{\Gamma_u} \mathbf{N}_I^T \mathbf{R} \mathbf{N}_J \mathrm{d}\Gamma}_{\mathbf{H}_{IJ}} \mathbf{u}_J - \beta \sum_I^{n_t} \delta \mathbf{u}_I \underbrace{\int_{\Gamma_u} \mathbf{N}_I^T \mathbf{r} \bar{\mathbf{u}} \mathrm{d}\Gamma}_{\mathbf{Q}_I} \\
&= \beta \sum_I^{n_t} \sum_J^{n_t} \delta \mathbf{u}_I^T \mathbf{H}_{IJ} \mathbf{u}_J - \beta \sum_I^{n_t} \delta \mathbf{u}_I \mathbf{Q}_I \\
&= \beta \delta \mathbf{U}^T \mathbf{H} \mathbf{U} - \beta \delta \mathbf{U}^T \mathbf{Q}
\end{aligned} \tag{2.100}$$

and the body forces are calculated as

$$\int_\Omega \delta \left(\sum_I^n \mathbf{N}_I \mathbf{u}_I \right)^T \mathbf{b} \, \mathrm{d}\Omega = \sum_I^{n_t} \delta \mathbf{u}_I^T \int_\Omega \mathbf{N}_I^T \mathbf{b} \, \mathrm{d}\Omega = \sum_I^{n_t} \delta \mathbf{u}_I^T \mathbf{f}_I^b = \delta \mathbf{U}^T \mathbf{F}^b \, . \tag{2.101}$$

The traction terms is rearranged into

$$\int_{\Gamma_t} \delta \left(\sum_I^n \mathbf{N}_I \mathbf{u}_I \right)^T \bar{\mathbf{t}} \, \mathrm{d}\Omega = \sum_I^{n_t} \delta \mathbf{u}_I^T \int_{\Gamma_t} \mathbf{N}_I^T \bar{\mathbf{t}} \, \mathrm{d}\Gamma = \sum_I^{n_t} \delta \mathbf{u}_I^T \mathbf{f}_I^t = \delta \mathbf{U}^T \mathbf{F}^t \, . \tag{2.102}$$

Equations (2.101) and (2.102) are combined together to form the nodal force $\mathbf{F}_I$ as

$$\begin{aligned}
&\int_\Omega \delta \left(\sum_I^n \mathbf{N}_I \mathbf{u}_I \right)^T \mathbf{b} \, \mathrm{d}\Omega + \int_\Omega \delta \left(\sum_I^n \mathbf{N}_I \mathbf{u}_I \right)^T \bar{\mathbf{t}} \, \mathrm{d}\Omega \\
&= \sum_I^{n_t} \delta \mathbf{u}_I^T \mathbf{f}_I^b + \sum_I^{n_t} \delta \mathbf{u}_I^T \mathbf{f}_I^t = \sum_I^{n_t} \delta \mathbf{u}_I^T \mathbf{F}_I \\
&= \delta \mathbf{U}^T \mathbf{F} \; .
\end{aligned} \tag{2.103}$$

By substituting Equations (2.94), (2.98) to (2.100) and (2.103) into (2.92), we have

$$\delta \mathbf{U}^T \mathbf{K} \mathbf{U} + \delta \mathbf{U}^T \mathbf{G} \mathbf{U} - \delta \mathbf{U}^T \mathbf{P} + \beta \delta \mathbf{U}^T \mathbf{H}^T \mathbf{U} - \beta \delta \mathbf{U}^T \mathbf{Q} - \delta \mathbf{U}^T \mathbf{F} = \mathbf{0} \; . \tag{2.104}$$

Cancelling the term $\delta \mathbf{U}$ and noting the *r.h.s* of the equation is zero, we obtain the global equations for the Nitsche-like weak form as

$$\left[\mathbf{K} + \mathbf{L} + \mathbf{G} + \beta \mathbf{H} \right] \{ \mathbf{U} \} = \{ \mathbf{P} + \beta \mathbf{Q} + \mathbf{F} \} \; , \tag{2.105}$$

where $\mathbf{K}$, $\mathbf{L} + \mathbf{G}$ and $\mathbf{H}$ are all square symmetric matrices and add up to form the global stiffness matrix; and $\mathbf{P}$, $\mathbf{Q}$, $\mathbf{F}$ are all vectors contributing to the nodal forces. Since all the terms in Equations (2.93), (2.98), (2.99) and (2.100) are symmetric, the global stiffness matrix obtained for the Nitsche-like weak form is also symmetric.

2.4.3 Penalty method

The numerical discretization of the penalty method is in general the same as the Nitsche-like method. Since the discretisation of the penalty method is a subset of Nitsche-like method, it can be obtained by simply removing the second and third terms on the *l.h.s* of Equation (2.105) to get

$$\left[\mathbf{K} + \beta \mathbf{H} \right] \{ \mathbf{U} \} = \{ \beta \mathbf{Q} + \mathbf{F} \} \; . \tag{2.106}$$

The value of the penalty factor β is set to be $10^4 \times E$ in all the test examples in the thesis (where E is the Young's modulus of the material) and note the dimension of β is force/length (depending on the units used in a problem). The penalty method can be seen as a simplified version of the Nitsche-like method. It is the simplest method amongst the three options and the global stiffness matrix is banded and symmetric. In this thesis, the penalty method will generally be used while the other two will occasionally be employed.

2.4.4 Lagrange multiplier method

Following the process in §2.4.2 to discretise Equation (2.66), we will have the discretised form of the Lagrange multiplier method

$$\int_\Omega \delta\left(\sum_I^n \mathbf{B}_I\mathbf{u}_I\right)^T\left(\mathbf{D}\sum_J^n \mathbf{B}_J\mathbf{u}_J\right)\mathrm{d}\Omega - \int_{\Gamma_u}\delta\boldsymbol{\lambda}^T\left(\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)-\bar{\mathbf{u}}\right)\mathrm{d}\Gamma \tag{2.107}$$
$$-\int_{\Gamma_u}\delta\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)^T\boldsymbol{\lambda}\mathrm{d}\Gamma = \int_\Omega \delta\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)^T\mathbf{b}\mathrm{d}\Omega + \int_{\Gamma_t}\delta\left(\sum_I^n \mathbf{N}_I\mathbf{u}_I\right)^T\bar{\mathbf{t}}\mathrm{d}\Gamma$$

where terms with Lagrange multiplier vector $\boldsymbol{\lambda}(\mathbf{x})$ contribute to the global stiffness and nodal forces. The Lagrange multiplier $\boldsymbol{\lambda}(\mathbf{x})$ is an unknown function of the local coordinates along the boundary, as shown in Figure 2.12 and can be interpolated by the boundary nodes as

$$\boldsymbol{\lambda}(\mathbf{x}) = \sum_I^{n_\lambda} N_I^\lambda(s)\lambda_I \quad \mathbf{x}\in\Gamma_u \tag{2.108}$$

where n_λ is the number of nodes used for interpolation and λ_I is the Lagrange multiplier unknown at a node. The interpolation $N_I(s)$ can take any form of interpolation such as Lagrangian

$$N_I^\lambda(s) = \frac{(s-s_0)(s-s_1)\cdots(s-s_{I-1})(s-s_{I+1})(s-s_{n_\lambda})}{(s_I-s_0)(s_I-s_1)\cdots(s_I-s_{I-1})(s_I-s_{k+1})(s_I-s_{n_\lambda})}\ , \tag{2.109}$$

where s, s_1, s_2 and $s_{n\lambda}$ are the local coordinates of the a point lying on the boundary as shown in Figure 2.12. Substituting Equation (2.108) into (2.66) and rearranging the terms gives

$$\underbrace{\int_\Omega\sum_I^n\sum_J^n \mathbf{B}_I^T\mathbf{D}\mathbf{B}_J\mathrm{d}\Omega\mathbf{u}_J}_{\mathbf{KU}} - \underbrace{\int_{\Gamma_u}\left(\sum_J^{n_\lambda}\sum_I^n \mathbf{N}_J^\lambda(s)\mathbf{N}_I\mathrm{d}\Gamma\mathbf{u}_I - \sum_J^{n_\lambda}\mathbf{N}_J^\lambda(s)\bar{\mathbf{u}}\right)}_{[-\mathbf{M}^T\mathbf{U}+\mathbf{R}]} \tag{2.110}$$
$$-\underbrace{\int_{\Gamma_u}\left(\sum_I^{n_\lambda}\sum_J^n \mathbf{N}_I^\lambda(s)\mathbf{N}_J\mathrm{d}\Gamma\mathbf{u}_I\right)}_{-\mathbf{M}\boldsymbol{\lambda}} = \underbrace{\int_\Omega\sum_I^n \mathbf{N}_I^T\mathbf{b}\mathrm{d}\Omega + \int_{\Gamma_t}\sum_I^n \mathbf{N}_I^T\bar{\mathbf{t}}\mathrm{d}\Gamma}_{\mathbf{F}}\ .$$

So the global stiffness matrix from Equation (2.110) can be obtained as

$$\begin{bmatrix} \mathbf{K} & \vdots & \mathbf{M} \\ \cdots & & \cdots \\ \mathbf{M}^T & \vdots & \mathbf{0} \end{bmatrix}\begin{Bmatrix} \mathbf{U} \\ \lambda \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \mathbf{R} \end{Bmatrix} \tag{2.111}$$

where the matrices $\mathbf{R}$, $\mathbf{M}$ and $\mathbf{K}$ are calculated according to Equation (2.107), and the load vector $\mathbf{F}$ is the same as in Equation (2.103).

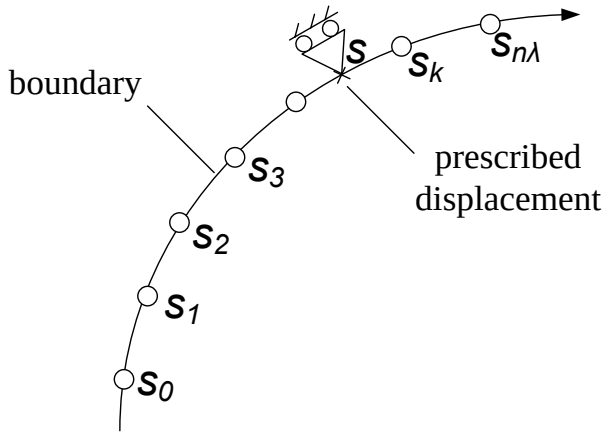


Figure 2.12: The local coordinate system along an essential boundary showing nodes and using Lagrange multipliers.

2.4.5 Remarks

The penalty method is an approximation of essential boundary conditions and the choice of the penalty factor should not be too large or too small. If $\beta = 0$, the essential boundary conditions are not enforced at all, because any field $\mathbf{u}$ will satisfy the requirement of Equation (2.54). If β becomes infinity, then the conditions are fully enforced as the weak form requires each term in Equation (2.54) to be zero. Normally, large constant values are chosen for β, so that the boundary conditions can be satisfied approximately, but not exactly. It should be mentioned that if the penalty is not large enough, then the conditions are not properly enforced. However, if β is too large, it results in an ill-conditioned matrix problem. If we compare the penalty method with the Nitsche-like method, we will find the only difference between these two methods are additional flux terms in the latter. In analyses for infinitesimal strain elastostatic problems, they all perform fairly well in terms of a global energy error used in the test examples. These flux terms have been found effective to improve the accuracy and stabilize constraints in shell analysis [133]. However there is no rigorous proof of available as to why or how they work. More tests could be done to compare the compatibility conditions along boundaries using different methods. The computational cost is more expensive in the Nitsche-like method and the computations of unit outer normals along boundaries are required which can be quite complicated in 3D. The Lagrange multiplier method is said to

be more rigorous than the penalty method since the Lagrange multiplier method applies the boundary conditions exactly [119, 133]. However it has some other numerical drawbacks. Firstly, the terms on the lower right corner of the global stiffness matrix are all zeros so that the global stiffness matrix is not positive definite [122]. Another problem with this method are the additional unknowns λ_I appearing in the global stiffness matrix so that the matrix is no longer banded as shown in Figure 2.13. The number of additional unknowns depends on the number of nodes lying on essential boundaries. In 3D problems, boundary conditions can be applied to surfaces meaning all the surface nodes lying on boundary will be used as unknowns, which is computationally unfavourable for large problems.

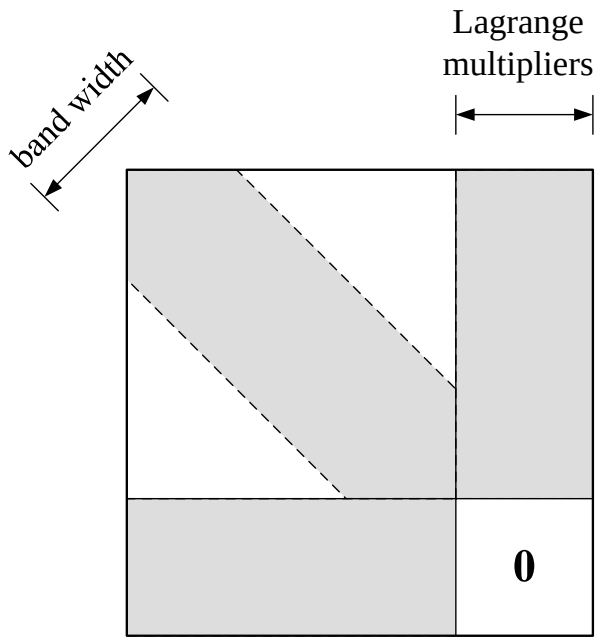


Figure 2.13: Non zeros terms in the global stiffness matrix obtained using the Lagrange multiplier method.

2.5 Integration scheme

In the earliest papers of the EFGM, a background mesh is used to perform the weak form integration. A background mesh divides the problem domain into a set of smaller cells and within each cell a quadrature scheme is used to integrate the weak

form as shown in Figure 2.14. For example, we can use 4×4 Gauss quadrature scheme over a square cell, Hammer integration for a triangular cell [134], and for 3D problems we can use $4 \times 4 \times 4$ Gauss integration. The background mesh does not have to conform with the problem geometry. However it requires additional geometric operations to trim off parts of the cells which intersect inner boundaries such as holes, inclusion and cracks and outer boundaries. Due to the use of a background mesh, the EFGM is regarded as not "truly meshless" by some compared with other meshless methods, e.g. the MLPG method [35] and the SPH method [119]. Dolbow and Belytschko [121] suggested the number of integration points be

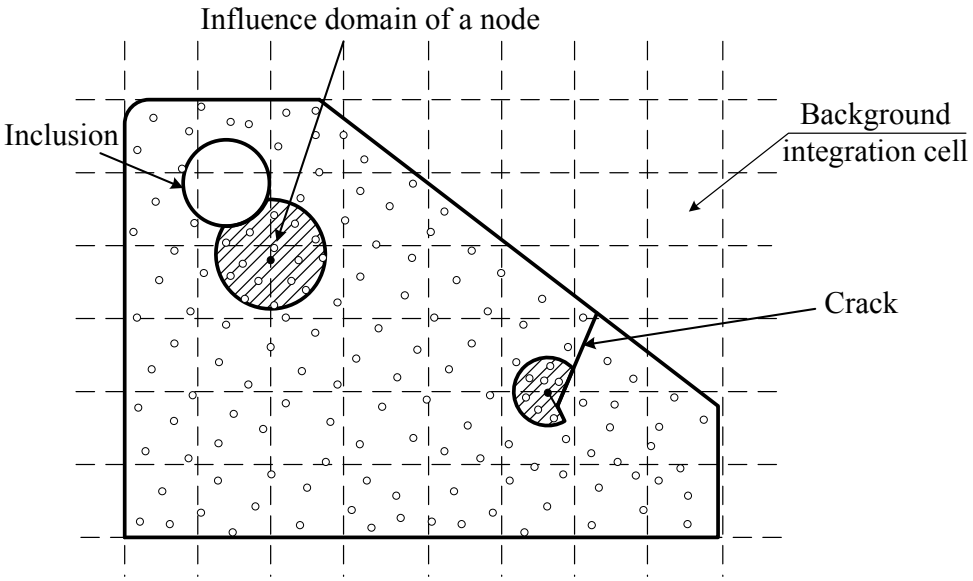


Figure 2.14: The background mesh used in the EFGM for integration. (reproduced from [3])

$9-10$ times the number of nodes to obtain acceptable results, indicating a need for many more points in the EFGM than in the FEM. Recalling that the shape functions at each integration point involves matrix inversion, which is computationally expensive, then generally it can be seen that the efficiency of the EFGM is much lower than the FEM. Various efforts have been made to develop an efficient and accurate integration scheme in the EFGM. Beissel and Belytschko [135] developed a nodal integration scheme similar to SPH where the quadrature point only samples

at nodes. The number of integration points needed is the same as the number of nodes, which greatly reduces the computational cost. However the method suffers from some new problems such as a slow rate of error convergence compared with background mesh integration, and also a quadratic basis is required to maintain the consistency condition which entails additional computational cost in obtaining the shape functions. Dolbow and Belytschko [136] suggested the integration accuracy can be largely improved with relatively few cells if the background mesh is partitioned with respect to its overlapping with nodal supports. However this results in complicated algorithms of geometric operations which is equivalent to or perhaps more expensive than, meshing especially in 3D. Chen et al. [137] proposed a conforming nodal integration using the smoothed strain terms which is more stable than direct nodal integration. The method was tested to show almost the same or slightly better accuracy than using Gauss integration. However, again there are additional costs in adding constraint terms and generating Voronoi cells around each node which could be regarded as equivalent to meshing. Fries and Belytschko [138] and Duflot and Hung [139] proposed a PU based integration scheme to remove the background mesh and called the EFGM with this scheme as a "truly meshless method". In this scheme, patches are generated around each node and are allowed to overlap each other as shown in Figure 2.15. The idea is to apply a PU function over each path so that the integration points in one patch are independent from those in other patches. Suppose $f(\mathbf{x})$ is a function that needs to be integrated over the problem domain, the PU-based integration gives

$$\int_{\Omega} f(\mathbf{x})\mathrm{d}\Omega = \sum_{k=1}^{l} \int_{\Omega\cap\Omega_k} w_k(\mathbf{x}) f(\mathbf{x})\mathrm{d}\Omega$$

where $w_k(\mathbf{x})$ forms a partition of unity $\sum_{k=1}^{l} w_k(\mathbf{x}) = 1 \,\forall \mathbf{x} \subset \Omega$, k is the index of patch and l is the total number of patches and here l should be equal to the total number of nodes. The weight function $w_k(\mathbf{x})$ can be constructed by any type of PU, such as the Shepard shape functions or the EFGM shape functions. For patches intersecting a boundary, there are integration points falling outside the domain. In this case $w_k(\mathbf{x})$ is simply set zero for these points. Thus additional computational costs are required to determine if a point falls in side or outside the boundary. It

should be noted that PU integration is different from the nodal integration. In nodal integration, the discretisation is rewritten and integration only samples nodal coordinates, while in PU integration the formulation is the same as in a standard EFGM and a number of integration points are arranged in the patch. Recently Liu

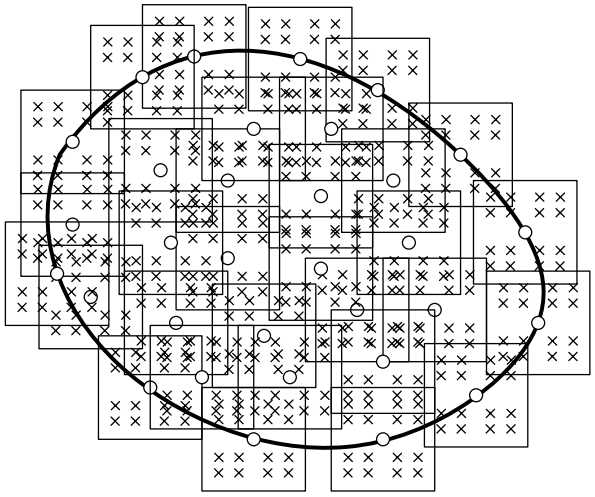


Figure 2.15: Nodal patches used in the partition of unity integration.

and Belytschko [140] derived the divergence free condition in the PU integration and proposed an improved method which has a better convergence rate than the previous PU integration methods. The PU integration method makes the EFGM performs more like a meshless method. However this method is only applicable to explicit analysis where there is no stiffness matrix to invert, unlike here.

2.6 Displacement and stress recovery

The nodal displacements $\mathbf{u} = \{u_1, u_2, \cdots, u_n\}^T$ do not corresponds to the displacement at the node $\mathbf{u}^h(\mathbf{x}_I)$. They are fictitious nodal parameters used to recover the field. The displacements at node I are calculated by

$$\mathbf{u}^h(\mathbf{x}_I) = \sum_J^n \phi_J(\mathbf{x}_I)\mathbf{u}_J = \mathbf{N}\mathbf{u} \ , \tag{2.112}$$

where $\mathbf{u}^h(\mathbf{x}_I)$ is the vector of approximated displacements at I, J is the index of nodes in support of node I, $\phi_J(\mathbf{x}_I)$ is the shape function of J and $\mathbf{u}_J$ is the vector of nodal values. It requires a MLS approximation at $\mathbf{x}_I$ which is slightly more

complicated than in the FEM as regards postprocessing. Similarly the stress at a point $\mathbf{x}$ can be obtained as

$$\boldsymbol{\sigma} = \mathbf{D} \begin{bmatrix} N_1(\mathbf{x})_{,x} & 0 & \cdots & N_n(\mathbf{x})_{,x} & 0 \\ 0 & N_1(\mathbf{x})_{,y} & \cdots & 0 & N_n(\mathbf{x})_{,y} \\ N_1(\mathbf{x})_{,y} & N_1(\mathbf{x})_{,x} & \cdots & N_n(\mathbf{x})_{,y} & N_n(\mathbf{x})_{,x} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ \vdots \\ u_n \\ v_n \end{Bmatrix} . \tag{2.113}$$

2.7 Implementation issues

The numerical implementation of the EFGM has many similarities to the FEM, for example in both methods the shape functions are calculated for each integration point and the the local stiffness matrix contribution of associated nodes is formed and stored. On the other hand, there are new features in the EFGM, such as nodal support, shape functions by the MLS approximation and modified weak forms, requiring special implementation, which is the subject of this section. Firstly focus will fall on the overall analysis process and storage and communication of data issues. Some algorithms and schemes are proposed to improve the efficiency and robustness which have been implemented in the EFGM program developed during this PhD used for analyses of fracture later in this thesis. Though the implementation described here is for the EFGM, the ideas are applicable to other meshless methods based on the concept of nodal supports.

2.7.1 Overview of analysis

The EFGM for 3D elastostatic problems has been coded in Matlab 7.0 according to the discretisation in §2.4. The analysis process of the program is shown in Figure 2.16. There are three main stages, namely preprocessing, solving and postprocessing. Firstly, the geometry model is built and discretised into scattered nodes by `gmsh` [141, 142]. In this program, the background mesh is independent from the nodal arrangement so it can be refined independently for adaptive analysis. The nodal coordinates and background cells are exported from `gmsh` into two separate ASCII

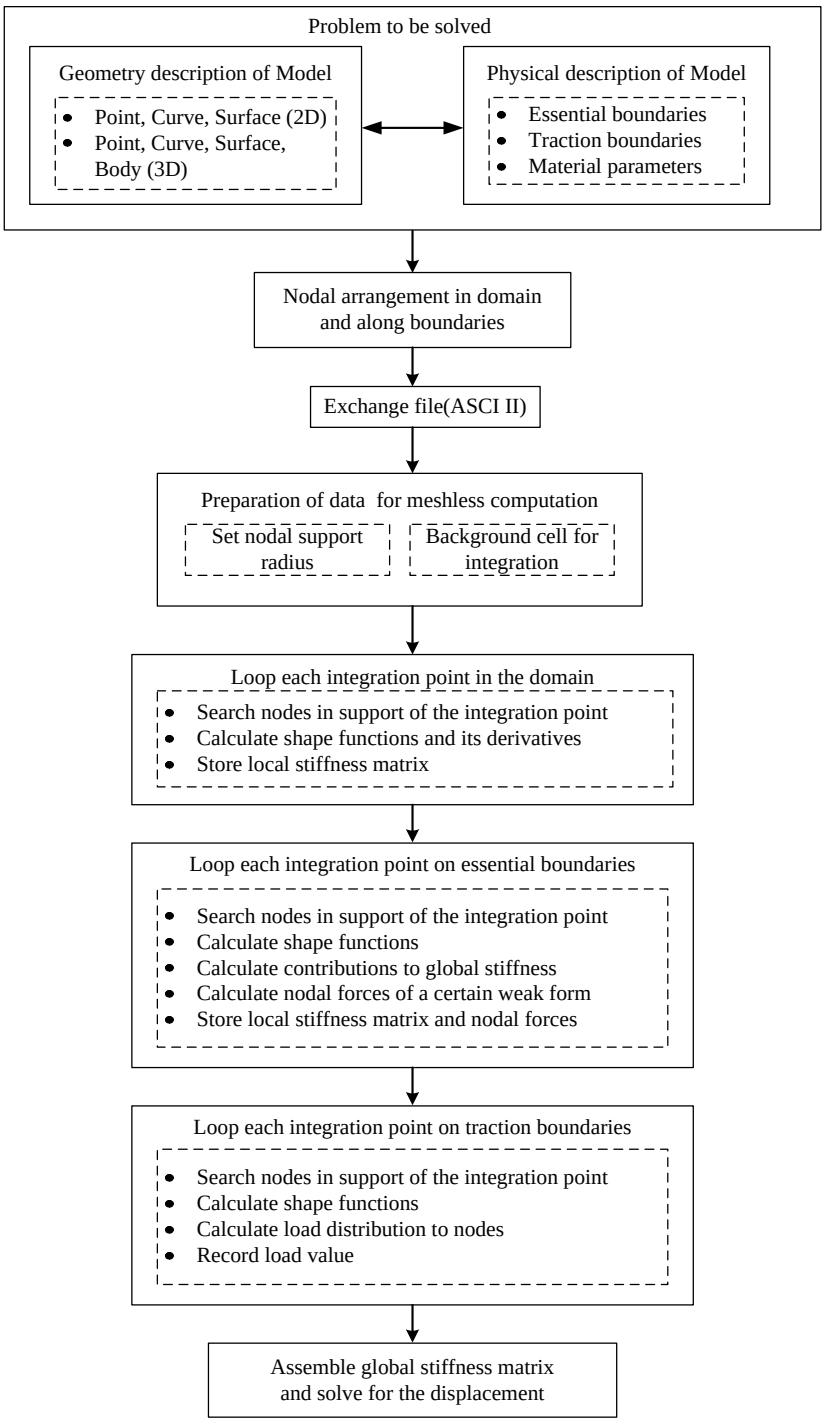


Figure 2.16: The flowchart of the EFGM analysis program.

files and are read into the Matlab program. Vectors storing the nodal information and cell information are generated. Based on the background cells, integration points are generated using a certain quadrature scheme over the problem domain and also along the boundaries. By looping over each integration point in the domain and on the boundaries, the nodes in support of the point are found to calculate the local stiffness matrices. These local matrices are stored and reduced into compact form. At the end of the loop, these matrices are finally assembled into the global stiffness matrix to solve for the nodal unknowns.

2.7.2 Bucket-query scheme for searching nodes in support

In order to calculate shape functions at a point of interest $\mathbf{x}$, we need to find all the nodes in support of the point needs to be found. This can be a very expensive procedure if we loop all the nodes for every integration point to determine if the support radius of a node is greater or smaller than the distance between the node and point. Since nodal support is locally defined, it is only necessary to include nodes in the vicinity of the point as candidate nodes in support, while nodes distant from the points can be excluded at the beginning. To define this vicinity region, a bucket scheme proposed as in [41] is used. The idea is to allot nodes into a number of buckets so that each bucket contains almost same number of nodes. For a point of interest, only the nodes in the same bucket of the point will be used as candidate nodes. An octree scheme is used in [108] which is similar to this bucket scheme. Here we propose a bucket-query scheme which is an improvement of the original bucket scheme [41]. Firstly a regular grid is set up as shown in Figure 2.17. The grid size is determined by the maximum radius of nodal support. Buckets are generated from these grid points as shown in the figure. The coordinates of the bottom left point of the grid are the minimum coordinates in x, y and z directions of the problem domain and the coordinates of the top right point are the maximum coordinates of the problem domain. Then we define

$$x_{i,j}^B = \frac{x_{max}^B - x_{min}^B}{N_i^B} \quad i = 1, 2, 3 \quad j = 1, 2, \cdots N_i^B \ . \tag{2.114}$$

where $x^B_{i,j}$ denotes the jth grid points in ith dimension of the buckets, the superscript B denotes "buckets" and x^B_{min} and x^B_{max} are the minimum and maximum x coordinates of the problem domain respectively. So if a node falls between the grid points, it is put into the corresponding bucket. The procedure for creating the node buckets are summarised as follows.

1. `Loop` over each node $\mathbf{x}_I$
2. `Loop` over each dimension i
   find j such that $x^B_{i,j} < \mathbf{x}_I(i) < x^B_{i,j+1}$
   The bucket number of the node in ith dimension is j
   `end`
3. put the node into the bucket

where $\mathbf{x}_I(i)$ denotes the ith coordinate of $\mathbf{x}_I$. To find the candidate nodes of a point shown as the cross in Figure 2.17, we firstly use the query grids which is half the size of the bucket grid. All the adjacent query grids are selected shown as the area in the figure.

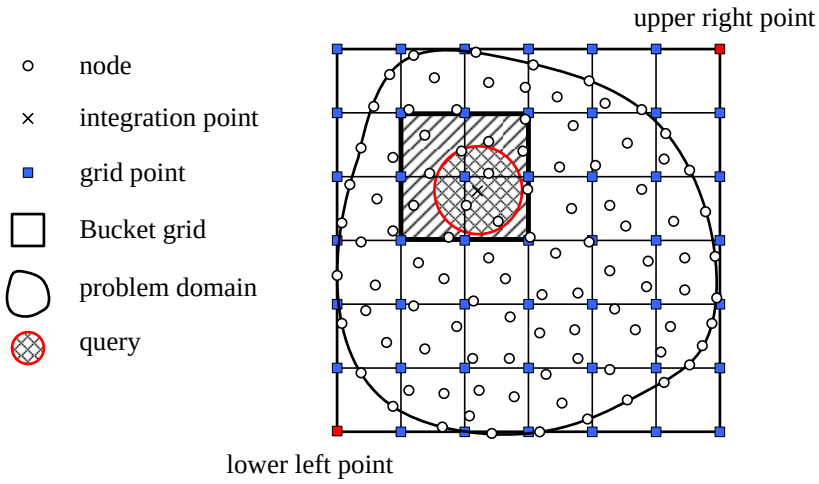
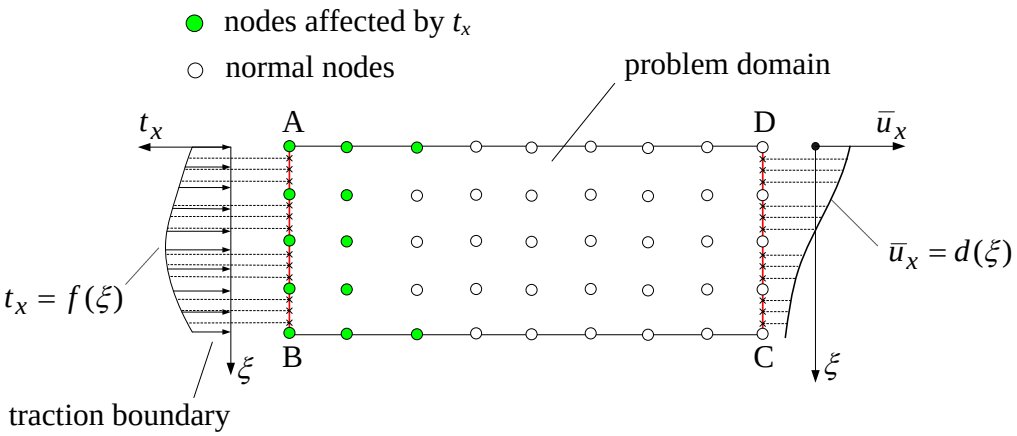


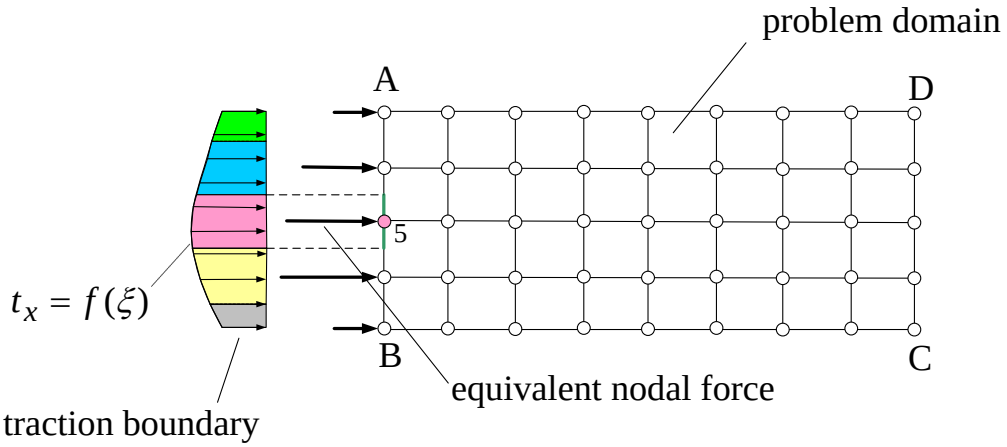
Figure 2.17: Bucket query scheme for searching nodes in support.

2.7.3 Boundary conditions

In the FEM, the boundary conditions are applied directly to nodes. For traction boundary conditions, equivalent nodal forces are calculated and then applied directly



(a) boundary conditions applied to integration points



(b) boundary conditions applied to nodes directly

Figure 2.18: The difference of applying the boundary conditions between the EFGM and the FEM.

to the nodes. For example, if a traction $t_x(\xi)$ needs to be applied to a boundary edge, for example line AB in Figure 2.18(b), then each node lying on AB will take its share of the tractions. The equivalent nodal forces, e.g. at node 5, is the traction integrated over the influence length of 5, which is the distance between the mid-points of its neighbouring nodes on AB shown as the thick line in the figure. Since in 2D there are two degrees of freedom at each node, the force on node 5 in the x direction corresponds to F_9 (subscript 9 is from $2 \times 5 - 1$ with 2 degrees of freedom at a node) in the *r.h.s.* load vector. Thus the global system equations will take the following form

$$\begin{bmatrix} K_{11} & \cdots & K_{1n} \\ \vdots & \ddots & \vdots \\ K_{91} & \cdots & K_{9n} \\ \vdots & \vdots & \vdots \\ K_{n1} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_5 \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} F_1 \\ \vdots \\ F_9 \\ \vdots \\ F_n \end{Bmatrix} \tag{2.115}$$

It can be seen that in the FEM, the traction boundaries will only affect nodes lying on that boundary. However, this is not the case in meshless methods. For example, if we apply tractions directly to the nodes on AB, for example nodal force in x direction applied to node 5, and suppose nodes 1, 2, 3 and 5 are in support of node 5, then the nodal forces are calculated by

$$K_{91}u_1 + K_{93}u_2 + K_{95}u_3 + K_{99}u_5 = F_9 \tag{2.116}$$

where K_{91}, K_{93}, K_{95} and K_{99} are stiffnesses of node 1, 2, 3 and 5 associated with the x degree of freedom of node 5. Since the MLS approximation gives non-zero values at $\mathbf{x}_5$, i.e.

$$N_1(\mathbf{x}_5) \neq 0, \quad N_2(\mathbf{x}_5) \neq 0, \quad N_3(\mathbf{x}_5) \neq 0 \ , \tag{2.117}$$

therefore K_{91} etc. will be non-zero and F_9 will produce non-zero nodal forces at nodes 1, 2 and 3. Thus the load is diffused between several nodes due to the nature of the MLS approximations. If not implemented correctly, this will lead to inconsistency problems in the solution. Thus when the traction boundary conditions are applied in meshless methods, it is applied to integration points lying on the boundary instead of

nodes. Edge AB is arranged with some integration points using a certain quadrature rule and the traction values at these integration points are evaluated using $t_x(\xi)$ as shown in Figure 2.18(a). Then, the shape functions at each integration point are computed and are multiplied with the traction values to get the nodal forces.

$$F_I = \sum_{i}^{n_{Gp}} w_i t_x(\xi) N_I(\xi) u_I \tag{2.118}$$

where i is the index of integration point, n_{Gp} is the total number of integration points along the boundary, w_i is the weight of the integration point and F_I is the equivalent nodal forces in the EFGM.

For essential boundary conditions in the FEM prescribed displacements are directly applied to corresponding nodes. For example, if displacement $\bar{u}_x$ needs to be applied at node 5, then $u_5 = \bar{u}_x$. To implement this in the FEM, there are two common approaches: the penalty method and row-column elimination. The penalty method here refers to the numerical scheme to solve the global stiffness matrix in context of linear algebra, which is not the same as the penalty method for the weak form in §2.3.4 though the idea of introducing large stiffness is similar. The method takes out the row corresponding to u_5 and

$$K_{91}u_1 + \cdots + \beta K_{95}u_5 + \cdots + K_{9n}u_n = \beta \bar{u}_x \tag{2.119}$$

where β is a large value constant so that $u_5 \approx \bar{u}_x$. The second approach substitutes $\bar{u}_x$ into u_5 to eliminate the row and condenses the columns containing u_5 so that u_5 will not appear in the vector of unknowns. In the EFGM, the essential boundary conditions are applied to integration points. For example, if displacement is to be assigned to AB as shown in Figure 2.18(a), we first generate integration points by a certain quadrature rule based on the local coordinate system of AB. Then we loop each integration point on AB to calculate the shape functions. For each integration point, there will be several nodes in support which are not necessarily lying on AB, for instance the green nodes shown in the figure. There will be contributions from these supporting nodes to the global stiffness matrix and also to the nodal forces.

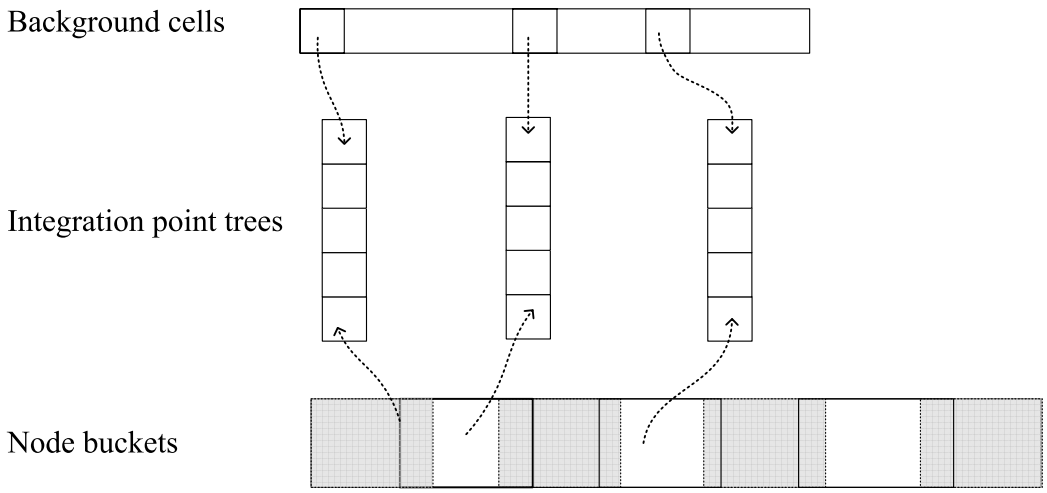


Figure 2.19: Dividing the integration points into several groups.

2.7.4 Data storage and assembling of stiffness matrix

In the EFGM the loop is performed over each integration point and the integration points can be divided into a number of smaller groups as shown in Figure 2.19. As there is no topology constraint between groups, they can be looped over independently to calculate the local stiffness matrices. Thus at the beginning of the analysis, the integration points are evenly divided into a number of groups and the size of each group can be a certain size, here a figure of 5000 is used. Then for each group we loop over each integration point to calculate the local stiffness matrix as shown in Figure 2.20. At the end of each loop, the local matrices are written into a binary file. Therefore for each group of integration points, there is a corresponding file recording the local stiffness matrices. For a small problem there is little difference in terms of computational efficiency using this partitioning scheme. However for 3D analysis, it has been found that as the memory allocated in the solving process is largely reduced the computation time is greatly reduced as demonstrated later in this thesis. Each group of integration points will only involve one or a few buckets of nodes as shown in Figure 2.19, so that only affected buckets of nodes need to be read in, which is an ideal property in doing parallel computation for large problem.

In the FEM, the size of the local (element) stiffness matrix is determined by the element type, so does not vary throughout the analysis a single elements is used. In

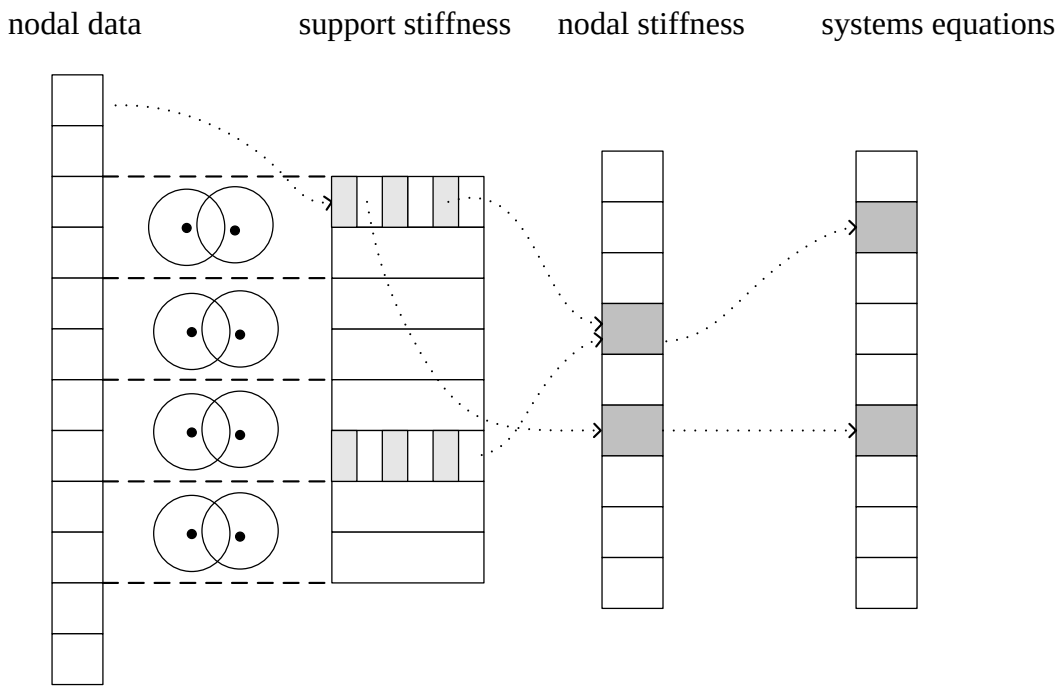


Figure 2.20: The data structure in the EFGM program.

the EFGM, the "local stiffness" matrix at each integration point $\mathbf{x}$ is given by

$$\mathbf{K}(\mathbf{x}) = \sum_{I}^{n} \sum_{J}^{n} \mathbf{B}_I{}^T \mathbf{D} \mathbf{B}_J \tag{2.120}$$

where n is the total number of nodes in support of $\mathbf{x}$, I and J are index of supporting nodes. So the size of $\mathbf{K}$ varies between integration points since each can be supported by different numbers of nodes. It is more expedient to store the nodal stiffness matrices rather than local stiffness matrices. The local stiffness matrix at each integration point is condensed into nodal stiffness vectors which are of the same size as shown in Figure 2.21. Two global vectors are used , a "local stiffness vector" for storing the nodal stiffness matrices and an "index mapping vector" for storing the map between the position of the local stiffness matrix to the global stiffness matrix as shown in Figure 2.21. In the index mapping vector, the first row stores the global position index and the second row store the column number in the local stiffness vector. The second row of the index mapping vector is initialized with -1 meaning no stiffness at start. When the local stiffness matrices are calculated and stored, pairs of nodes linked by the stiffness matrix are recognized and for each pair of nodes a unique index of the position in the global stiffness is generated as shown

in Figure 2.21.

Traction boundaries and essential boundary conditions are applied to integration points along the boundaries and the computing process is similar to the domain integration points. Traction boundaries will only contribute to nodal forces while essential boundaries can affect both nodal stiffness and nodal forces depending on the choice of weak form. After all of these operations, nodal stiffness matrices stored in the files are imported and assembled to form the global stiffness matrix. The index map is $n_{df} \times n_{df}$ where n_{df} indicates the total of degrees of freedom. When the penalty method or the Nitsche-like method is used, $n_{df} = \text{problem dimension} \times n_t$, where n_t is the total number of nodes. For example, in 2D $n_{df} = 2 \times n_t$. When the Lagrange multiplier method is used, there are additional degrees of freedom along the essential boundaries, so $n_{df} = \text{problem dimension} \times (n_t + n_\lambda)$ where n_λ is the number of nodes involved in the Lagrangian interpolant (see Equation (2.107) and (2.108)).

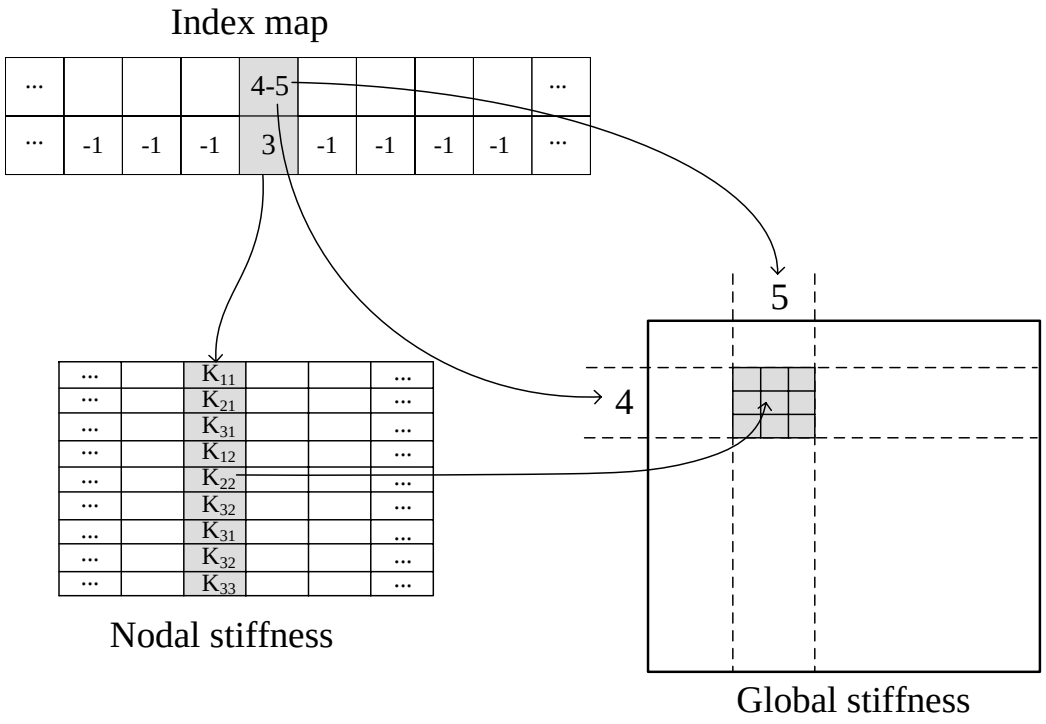


Figure 2.21: The vector storing nodal stiffness and the vector mapping nodal index to global stiffness position.

2.8 Conclusions

In this chapter, the formulation of the EFGM has been described highlighting nodal support and weight functions as two features distinct from the FEM. The EFGM shape functions are generated by the MLS approximation which is smooth and of high order differentiability. However this "smoothness" causes problems in applying the essential boundary conditions and the weak form used in the FEM is no longer valid. Direct imposition of nodal values results in an incompatible field solution. Two common approaches used solve this problem, namely the penalty method and Lagrange multiplier method, have been derived from Hamilton's principle. It has been shown that both of these methods have drawbacks in implementation such as ill-conditioning or the addition of extra unknowns. Nitsche's method, which is originally proposed for a different elliptic problem, is here extended for solid mechanics problems and works better in practice. Different weak forms have also been shown to be related to each other by the addition or removal of constraining terms on essential boundaries. Discretisation of the weak forms has been detailed and implementation issues are discussed. In terms of the computational cost, the EFGM shape functions are more expensive than the FEM shape functions since matrix inversion is involved not only in the solving stage but also in postprocessing to recover displacement and stress. However, in the EFGM there is no topology constraint when partitioning the meshless model into smaller groups which is ideal for parallel computation. Efficient data storage and algorithms have been proposed.

Chapter 3

The use of orthogonal basis functions for computing shape functions in the EFGM

3.1 Introduction

In the previous chapter, the formulation of the EFGM has been derived where shape functions are derived via a calculation involving the inversion of a small matrix. In this chapter, the accuracy in obtaining these shape function is investigated. An alternative implementation of the EFG method, published soon after the original [3] proposed an alternative approach using orthogonal bases to avoid the matrix inversion in the formulation of the shape functions. In this chapter we revisit this topic and show that the difficulties associated with the use of a polynomial basis remain present in the orthogonal case. We also show that certain terms in the derivative expressions are omitted in the "alternative implementation" of the EFGM which can lead to errors. Finally a new approach is proposed which avoids inversion while maintaining accuracy. This new approach is used for test examples later in this thesis.

The chapter is structured as follows. Firstly the MLS approximation and the expressions to derive shape functions using a polynomial basis are presented. Next the alternative use of an orthogonal basis is outlined and the removal of the need

for matrix inversion is shown not to remove the degradation in solution accuracy, which arises from the spatial distribution of nodes and not from the basis used. In §3.3 missing terms in the shape function derivatives are shown to lead to errors. Finally alterations are proposed to the new implementation of the EFG method in §3.4 that remedy both of these issues satisfactorily.

3.2 Use of orthogonalization

For certain arrangements of supporting nodes, $\mathbf{A}$ in Equation (2.4) can become ill-conditioned, as will be highlighted later. If $\mathbf{A}$ is ill-conditioned, the solution for $\mathbf{a}(\mathbf{x})$ will be prone to error, and hence also the shape functions $\mathbf{\Phi}$. Lu *et al.* [132] proposed overcoming this problem by using an orthogonal basis $\tilde{\mathbf{p}}$ which leads to a diagonal $\mathbf{A}$ and hence a trivial inversion process. While apparently a reasonable solution to this problem (a claim also repeated in later references, e.g. [143]) ill-conditioning is not removed using the orthogonal basis, as we will show below. The Schmidt orthogonalization procedure to obtain $\tilde{\mathbf{p}}$ is described in [132] and is not repeated here.

To make comparisons $\tilde{\mathbf{A}}(\mathbf{x})$, $\tilde{\mathbf{a}}(\mathbf{x})$ and $\tilde{\mathbf{B}}$ denote the equivalent expressions constructed from the orthogonal basis $\tilde{\mathbf{p}}(\mathbf{x})$. If the basis used is orthogonal then using (2.5)

$$\tilde{A}_{jk} = \sum_{I}^{n} w_I(\mathbf{x})\tilde{p}_j(\mathbf{x}_I)\tilde{p}_k(\mathbf{x}_I) = 0,\ j \neq k. \tag{3.1}$$

From (3.1) $\tilde{\mathbf{A}}(\mathbf{x})$ now becomes a diagonal matrix

$$\tilde{\mathbf{A}}(\mathbf{x}) = \begin{bmatrix} \sum_{I}^{n} w_I(\mathbf{x})\tilde{p}_1^2(\mathbf{x}_I) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sum_{I}^{n} w_I(\mathbf{x})\tilde{p}_m^2(\mathbf{x}_I) \end{bmatrix} \tag{3.2}$$

and the elements of $\tilde{\mathbf{B}}$ are

$$\tilde{B}_{jI}(\mathbf{x}) = w_I(\mathbf{x})\tilde{p}_j(\mathbf{x}_I). \tag{3.3}$$

Therefore (2.4) can be solved without matrix inversion by

$$\tilde{a}_j(\mathbf{x}) = \frac{\sum_I^n \tilde{B}_{jI} u_I}{\tilde{A}_{jj}} \tag{3.4}$$

and and the shape functions $\tilde{\phi}(\mathbf{x})$ are given as

$$\tilde{\phi}_I(\mathbf{x}) = \sum_j^m \frac{\tilde{B}_{jI}\tilde{p}_j(\mathbf{x})}{\tilde{A}_{jj}} = w(\mathbf{x}_I)\sum_j^m \frac{\tilde{p}_j(\mathbf{x}_I)\tilde{p}_j(\mathbf{x})}{\tilde{A}_{jj}} \tag{3.5}$$

The aim of the orthogonalization process is to remove inaccuracies in the MLS approximation by avoiding the inversion of an ill-conditioned matrix. Here we show that this process does not result in better accuracy as expected as the source of inaccuracy (causing the ill-conditioning) is not removed by this procedure. As both the orthogonal basis $\tilde{\mathbf{p}}(\mathbf{x})$ and Pascal basis $\mathbf{p}(\mathbf{x})$ are complete and span the entire finite spaces (2D or 3D), a linear relation exists between them, i.e. each $\tilde{p}_j(\mathbf{x})$ can be written as a linear combination of $p_i(\mathbf{x})$ as

$$\tilde{p}_j(\mathbf{x}) = \sum_i^m t_{ji} p_i(\mathbf{x}), \quad j = 1, \cdots, m \tag{3.6}$$

and in matrix form as

$$\tilde{\mathbf{p}}(\mathbf{x}) = \mathbf{T}\mathbf{p}(\mathbf{x}) \tag{3.7}$$

where $\mathbf{T}$ is a transformation matrix of coefficients t_{ji}. The completeness condition requires that $\det(\mathbf{T}) \neq 0$ and the Schmidt orthogonalization makes

$$t_{ji} = \left\{ \begin{array}{ll} 0, & i > j \\ 1, & i = j \end{array} \right\}. \tag{3.8}$$

Thus $\mathbf{T}$ is a lower triangular matrix mapping $\mathbf{p}(\mathbf{x})$ to $\tilde{\mathbf{p}}(\mathbf{x})$ with all diagonal entries of unit value

$$\mathbf{T} = \begin{bmatrix} 1 & & & 0 \\ t_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ t_{m1} & t_{m2} & \cdots & 1 \end{bmatrix}_{m\times m}. \tag{3.9}$$

Maintaining the convention that quantities derived from the orthogonal basis are written $\tilde{\mathbf{A}}(\mathbf{x})$ etc. from (2.12) and (3.7) we have

$$\tilde{\mathbf{P}} = \mathbf{T}\mathbf{P}. \tag{3.10}$$

By substituting (3.10) into

$$\tilde{\mathbf{A}}(\mathbf{x}) = \tilde{\mathbf{P}}(\mathbf{x})\mathbf{W}(\mathbf{x})\tilde{\mathbf{P}}^T, \tag{3.11}$$

we have

$$\tilde{\mathbf{A}}(\mathbf{x}) = \mathbf{T}\mathbf{A}(\mathbf{x})\mathbf{T}^T. \tag{3.12}$$

When $\mathbf{A}$ is ill-conditioned its determinant is close to zero so

$$|\det(\mathbf{A})| = |\det\left(\mathbf{P}\mathbf{W}\mathbf{P}^T\right)| \leq |\det(\mathbf{P}\mathbf{I}\mathbf{P}^T)| \cong 0 \tag{3.13}$$

since the weight function is non-negative and bounded in value $w_I(\mathbf{x}) \leq 1$. Noting that $\det(\mathbf{T}) = 1$ from (3.9), we have

$$\det(\tilde{\mathbf{A}}) = \det(\mathbf{A}) \cong 0 \tag{3.14}$$

so the use of the orthogonal basis leads to a similarly ill-conditioned system. $\det(\tilde{\mathbf{A}}) = 0$ implies one or more of $\tilde{A}_{jj}$ is close to zero and hence a small perturbation in $\tilde{\mathbf{B}}(\mathbf{x})$ will be amplified into a large error in $\mathbf{a}(\mathbf{x})$ and the shape function $\boldsymbol{\phi}_I(\mathbf{x})$. The inaccuracy associated with an ill-conditioned $\mathbf{A}(\mathbf{x})$ is an inherent round-off error which can never be ameliorated via the orthogonalization process. Its source is a poor spatial arrangement of nodes in support at $\mathbf{x}$. With a linear basis in 2D, this occurs when three nodes lie on a line. Another way of viewing this is to note that the area enclosed by a simplex formed from the nodes in support must be non-zero. In this case $\det(\mathbf{P}) \cong 0$, if assuming $m = n$ so $\mathbf{P}$ becomes a square matrix, which leads to (3.13). The only possible solution is to improve the nodal arrangement to make non-zero areas (2D) or volumes (3D). This has not been highlighted before in the literature to our knowledge. It is important to note that this source of ill-conditioning is not the same as mentioned in some other references. In [55] a technique is used where the origin is shifted to the point of interest when calculating the basis matrix. This avoids accuracy problems due to the size of the arguments in the basis but is not related to the problem cited above which is due to node topology. Reference [138] mentions the importance of nodal arrangements but only insofar as they affect the integration necessary for the weak form, again not the problem that is highlighted here.

Another important feature should be noted, that the shape functions derived using either basis are identical. Substituting (3.10) into

$$\tilde{\mathbf{B}}(\mathbf{x}) = \tilde{\mathbf{P}}(\mathbf{x})\mathbf{W}(\mathbf{x}), \tag{3.15}$$

we can show that

$$\tilde{\mathbf{B}}(\mathbf{x}) = \mathbf{T}\mathbf{B}(\mathbf{x}). \tag{3.16}$$

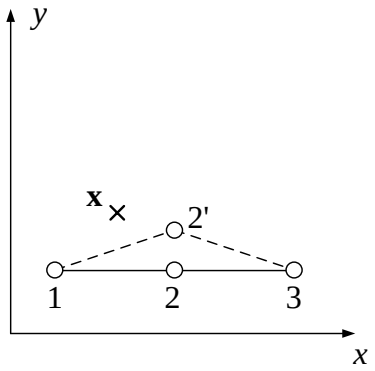
Then substituting (3.16) and (3.11) into

$$\tilde{\phi}(\mathbf{x}) = \tilde{\mathbf{p}}(\mathbf{x})^T \tilde{\mathbf{A}}(\mathbf{x})^{-1} \tilde{\mathbf{B}}(\mathbf{x}) \tag{3.17}$$

we can see that

$$\tilde{\phi}(\mathbf{x}) \equiv \phi(\mathbf{x})\,. \tag{3.18}$$

3.2.1 Example: simple degradation cases



node	coordinates (*x*, *y*)
1	1, 1
2	2, 1
3	3, 1

Figure 3.1: The nodal arrangement for the study of degradation cases.

An example is now presented to demonstrate the degradation in accuracy due to nodal arrangement. Results are compared using the standard Pascal basis and bases derived from the orthogonalization process. Figure 3.1 shows three nodes (labelled 1 to 3) used to provide a MLS approximation at $\mathbf{x}$, a point located at $(1.5, 1.5)$. Firstly, node 2 is located on the line between nodes 1 and 3, which is the degradation case in 2D mentioned above. Node 2 is then moved off the line to produce several close-to-degradation cases. The support radius is set at 2.0 units for all nodes. (This value is

not significant but means that all three nodes are in support at $\mathbf{x}$.) Table 3.1 shows the results from calculations varying the location of node 2. Results are provided for cases where node 2 is located 0.001, 0.01 and 0.1 units above the line joining nodes 1 and 3 (cases b, c, d) in addition to the inline case (a). Results are also given where node 2 is perturbed by 0.0001 units ($a_\epsilon, b_\epsilon, c_\epsilon, d_\epsilon$) to examine the effect of inaccuracy in nodal coordinates. Estimates of the reciprocal condition number of $\mathbf{A}$ and the minimum values of $\tilde{A}_{jj}$ are given along with the values of shape functions, which are identical for both approaches as proved above. (Note that the MLS approximation leads to negative shape functions which nevertheless form a partition of unity.)

Case	Coordinates of node 2	$rcond(\mathbf{A})$	$min\tilde{A}_{jj}$	ϕ_1	ϕ_2	ϕ_3
(a)	2,1	0	0	NaN	NaN	NaN
(a_ϵ)	2,1.0001	6.25e-11	8.12e-10	-2499.25	5000.00	-2499.75
(b)	2,1.0010	6.25e-9	8.12e-8	-249.25	500.00	-249.75
(b_ϵ)	2,1.0011			-226.52	454.54	-227.02
(c)	2,1.0100	6.21-e7	8.21e-6	-24.25	50.00	-24.75
(c_ϵ)	2,1.0101			-24.00	49.50	-24.50
(d)	2,1.1000	5.77e-5	8.33e-4	-1.75	5.00	-2.25
(d_ϵ)	2,1.1001			-1.75	5.00	-2.25

Table 3.1: Degradation cases and perturbation of shape functions.

Case a shows that no shape functions can be derived for the degradation case as $\mathbf{A}$ is singular, or there is a zero on the diagonal of $\tilde{\mathbf{A}}$. As node 2 is moved away from the line, ill-conditioning decreases rapidly and the range of the shape functions decreases by two orders of magnitude. The effect of slight perturbations in the coordinates of node 2 are also seen to be of much greater significance when close to the degradation case regardless of the basis used. If the relative error in the shape functions caused by the small perturbation in cases $b_\epsilon, c_\epsilon, d_\epsilon$ is defined as

$$e_{\boldsymbol{\phi}} = \frac{||\boldsymbol{\Phi}(\mathbf{x}) - \boldsymbol{\Phi}_\epsilon(\mathbf{x})||}{||\boldsymbol{\Phi}(\mathbf{x})||} \tag{3.19}$$

where $\|\cdot\|$ is the L_2 norm of a vector (refer to Equation (1.21)). We plot the variation in error with condition of $\mathbf{A}$ and $\tilde{A}_{jj}$ (Figure 3.2) the power law dependence between

condition and error is clear. For practical problems, the significance of this effect could be considerable as errors in coordinate values may be encountered where nodes are fitted to a curve or a surface for the discretization of a domain, and in retrieving stress fields based on nodal coordinates.

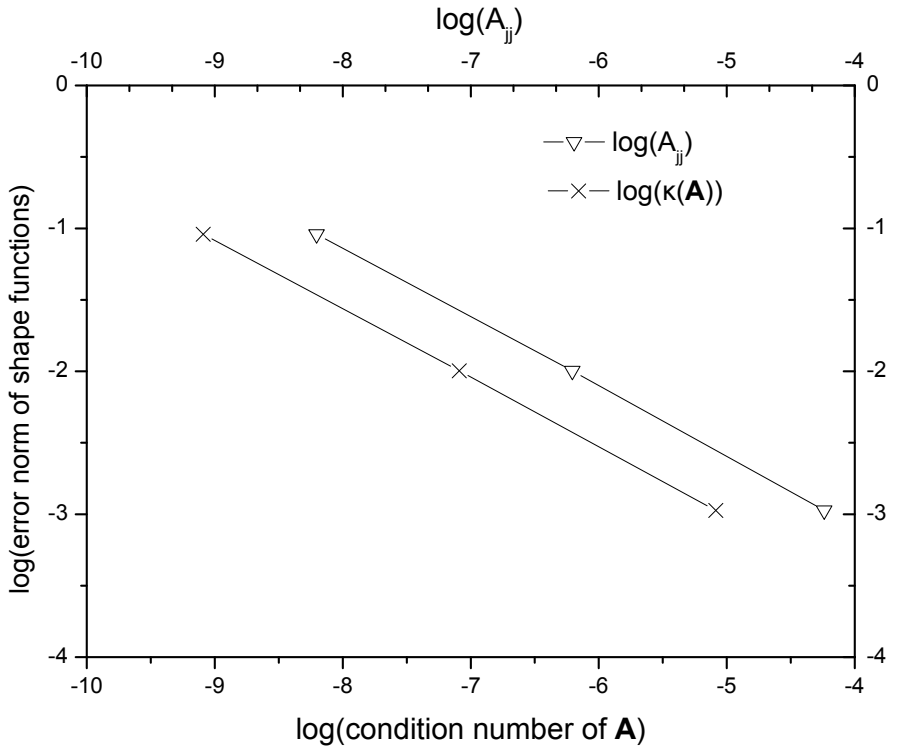


Figure 3.2: Shape function relative errors for degradation cases.

3.3 The effect of missing terms in the derivatives

A general procedure to obtain shape functions is by solving Equation (2.8), where $\phi_I(\mathbf{x})$ is a result of all the terms in $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$ which can be expressed by

$$\phi_I(\mathbf{x}) = \mathbf{F}\left(p_j(\mathbf{x}), p_i(\mathbf{x}_I)p_j(\mathbf{x}_I), w_I(\mathbf{x})\right) \quad i,\, j = 1, \ldots, m. \tag{3.20}$$

$\mathbf{F}$ can be decomposed into $\mathbf{F}_1$ and $\mathbf{F}_2$, the part having independent influence from each basis and the part having mutual influence between bases

$$\phi_I(\mathbf{x}) = \mathbf{F}_1\left(\tilde{p}_j(\mathbf{x}), \tilde{p}_j^2(\mathbf{x}_I), w_I(\mathbf{x})\right) + \mathbf{F}_2\left(\tilde{p}_j(\mathbf{x}), \tilde{p}_i(\mathbf{x}_I)\tilde{p}_j(\mathbf{x}_I), w_I(\mathbf{x})\right) \quad i \neq j. \tag{3.21}$$

The use of an orthogonal basis then makes $\mathbf{F}_2$ zero as all the off-diagonal terms in $\tilde{\mathbf{A}}$ become zero. In this special case, the shape functions $\phi_I(\mathbf{x})$ become

$$\phi_I(\mathbf{x}) = \mathbf{F}_1 \left(\tilde{p}_j(\mathbf{x}), \tilde{p}_j^2(\mathbf{x}_I), w_I(\mathbf{x}) \right) . \tag{3.22}$$

However, it does not follow that the matrix of derivatives of $\tilde{\mathbf{A}}(\mathbf{x})$ is also diagonal. This appears to have been assumed in [132] since the expression given there (Equation 17) is

$$\phi_I(\mathbf{x}) = w_I(\mathbf{x}) \sum_j^m C_{jI}(\mathbf{x}) \tag{3.23}$$

where

$$C_{jI}(\mathbf{x}) = \frac{\tilde{p}_j(\mathbf{x}, \mathbf{x}) \tilde{p}_j(\mathbf{x}_I, \mathbf{x})}{\tilde{A}_{jj}} \ . \tag{3.24}$$

Note that here $\tilde{A}_{jj}$, which is defined by Equation (3.1), is equivalent to $b_j(\mathbf{x})$ in [132]. Then it is obvious to see that Equation (3.23) has a zero-value term hidden behind

$$\phi_I(\mathbf{x}) = w_I(\mathbf{x}) \sum_j^m C_{jI}(\mathbf{x}) + \underbrace{\mathbf{F}_2(\tilde{p}_i(\mathbf{x}), \tilde{p}_i(\mathbf{x}_I) \tilde{p}_j(\mathbf{x}_I), w_I(\mathbf{x}))}_{\text{zero}} \quad i \neq j. \tag{3.25}$$

In [132], the last term on the *l.h.s* is omitted when determining the shape function derivatives. It should be noted that the omission here is not the same as the diffuse derivatives of the DEM discussed in [120]. This term vanishes when the shape functions are formed, but not necessarily for shape function derivatives, i.e.

$$\phi_I(\mathbf{x})_{,k} = \left(w_I(\mathbf{x}) \sum_j^m C_{jI}(\mathbf{x}) \right)_{,k} + \underbrace{\mathbf{F}_2(\tilde{p}_j(\mathbf{x}), \tilde{p}_i(\mathbf{x}_I) \tilde{p}_j(\mathbf{x}_I), w_I(\mathbf{x}))_{,k}}_{\text{non-zero}} . \tag{3.26}$$

To investigate if this omission is indeed significant a standard patch test was carried out. Figure 3.3 shows the patch size and nodal distribution. The background integration cells are eight right-angled triangles using a 4 point Gauss integration scheme in each. Displacements producing constant strain are prescribed along the four boundaries. Three formulations using MLS approximations are compared: the original Pascal basis approach including off-diagonal terms (termed method A), the Diffuse element method (DEM) [28] and the new EFG formulation in [132]. The

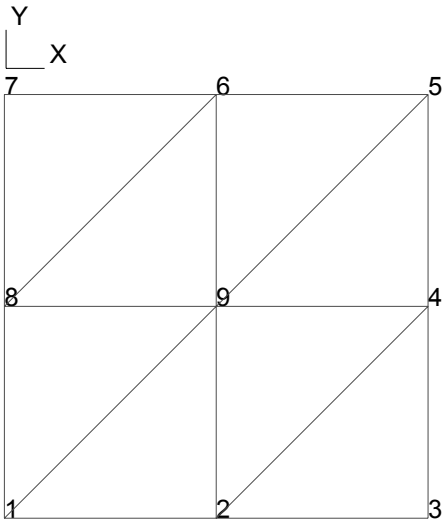


Figure 3.3: Patch test and nodal arrangement.

shape function derivatives, in any basis, are given by Equations (2.9) and (2.10). In the DEM, the derivatives of shape functions (known as diffuse derivatives) are given by

$$\boldsymbol{\phi}_{,k} = \mathbf{p}_{,k}^{T}\mathbf{A}^{-1}\mathbf{B}. \tag{3.27}$$

which is the first term in Equation (2.10). While the derivatives used in [132] ignore the off-diagonal terms in $\mathbf{A}(\mathbf{x})$, the derivatives of $\mathbf{A}_{,k}^{-1}$ used there can be expressed as

$$\mathbf{A}_{,k}^{-1} = -\mathbf{A}^{-1}\mathrm{diag}(\mathrm{diag}(\mathbf{A}_{,k}))\mathbf{A}^{-1}\,. \tag{3.28}$$

The weight function used is the exponential function from [3] (refer to §2.2.2 for the definition). Here, the relative error norm of displacement is defined as

$$r_u = \frac{\|\mathbf{u}^{num} - \mathbf{u}^{exact}\|}{\|\mathbf{u}^{exact}\|}\,. \tag{3.29}$$

where $\|\cdot\|$ is the L_2 norm of a vector (refer to Equation (1.21)). Figure 3.4 plots r_u with varying d_I/c. In all the tests essential boundary conditions are imposed directly.

Method A and the DEM pass the patch test for all values of d_I/c, while the formulation from [132] shows a loss of accuracy and sensitivity to d_I/c. This formulation does not pass the patch test and shows an error norm of 0.2 with node 9

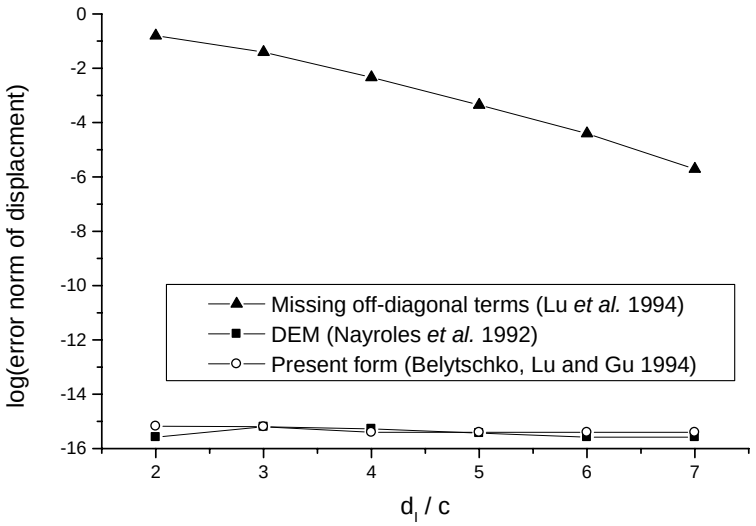


Figure 3.4: Displacement error norm with varying dI/c.

located in the centre (compare with the first row of Table 2 in [132]). This error is clearly caused by the omission of off-diagonal elements. The DEM formulation performs much better than [132] and it is interesting to see the extra term in the derivatives of $\mathbf{A}^{-1}\mathbf{B}$ makes the method more unstable than not adding it at all (DEM).

3.4 An alternative approach

Here we propose a new approach which maintains some of the advantages of the method in [132] using orthogonal bases, but avoids the errors associated with omission of off-diagonal terms in the derivatives. Firstly we focus on the relations between the derivatives of the shape functions using the two types of polynomial basis. The shape function derivatives using the orthogonal basis can be written generally as

$$\tilde{\boldsymbol{\phi}}_{,k} = \tilde{\mathbf{p}}_{,k}^T \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{B}} + \tilde{\mathbf{p}}^T \left(\tilde{\mathbf{A}}_{,k}^{-1} \tilde{\mathbf{B}} + \tilde{\mathbf{A}}^{-1} \tilde{\mathbf{B}}_{,k}^T \right) \tag{3.30}$$

where $\tilde{\mathbf{A}}_{,k}^{-1}$ is computed by

$$\tilde{\mathbf{A}}_{,k}^{-1} = -\tilde{\mathbf{A}}^{-1} \tilde{\mathbf{A}}_{,k} \tilde{\mathbf{A}}^{-1} \,. \tag{3.31}$$

From Equations (3.7) (3.11) (3.16), the derivatives of $\tilde{\mathbf{p}}(\mathbf{x})$, $\tilde{\mathbf{A}}(\mathbf{x})$ and $\tilde{\mathbf{B}}(\mathbf{x})$ can be written as

$$\tilde{\mathbf{p}}_{,k} = (\mathbf{T}\mathbf{p}(\mathbf{x}))_{,k} \tag{3.32a}$$

$$\tilde{\mathbf{A}}_{,k} = (\tilde{\mathbf{P}}(\mathbf{x})\mathbf{W}(\mathbf{x})\tilde{\mathbf{P}}(\mathbf{x})^T)_{,k} \tag{3.32b}$$

$$\tilde{\mathbf{B}}_{,k} = (\tilde{\mathbf{P}}(\mathbf{x})\mathbf{W}(\mathbf{x}))_{,k} \ . \tag{3.32c}$$

It appears from the above that it is necessary to differentiate $\mathbf{T}$ and $\tilde{\mathbf{P}}$ with respect to $\mathbf{x}$. However this is not so as firstly,

$$\tilde{\mathbf{P}}(\mathbf{x}_I, \mathbf{x})_{,k} = (\mathbf{T}\mathbf{P})_{,k} = \mathbf{T}_{,k}\mathbf{P} + \mathbf{T}\mathbf{P}_{,k}. \tag{3.33}$$

Now, it is clear that $\mathbf{P}_{,k} = 0$ and it can also be shown that the derivatives of $\mathbf{T}$ are zero. From (3.18), the relation between the derivatives of the shape functions from the two bases is

$$\tilde{\boldsymbol{\phi}}_{,k} \equiv \boldsymbol{\phi}_{,k} \ . \tag{3.34}$$

Substituting (3.7), (3.12) (3.16) and (3.31) into (3.30) gives

$$\mathbf{T}_{,k} = 0 \tag{3.35}$$

to satisfy (3.34). Thus, there is no need to obtain derivatives of $\mathbf{T}$ and $\tilde{\mathbf{P}}$ for shape function evaluation. The derivatives of $\tilde{\mathbf{p}}$, $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ can therefore be computed as those from the Pascal basis as

$$\tilde{\mathbf{p}}_{,k} = \mathbf{T}\mathbf{p}(\mathbf{x})_{,k} \tag{3.36a}$$

$$\tilde{\mathbf{A}}_{,k} = \tilde{\mathbf{P}}\mathbf{W}_{,k}\tilde{\mathbf{P}}^T \tag{3.36b}$$

$$\tilde{\mathbf{B}}_{,k} = \tilde{\mathbf{P}}\mathbf{W}_{,k} \tag{3.36c}$$

where $\mathbf{W}_{,k}$ is computed from the nodal data. The above analysis shows $\mathbf{T}(\bar{\mathbf{x}}, \mathbf{x}_I)$ is only needed for finding the orthogonal basis as $\mathbf{T} = const.$ The shape function derivatives, like the shape functions themselves, can be constructed from the Pascal basis **or** from an orthogonal basis. Another way to understand this is that the MLS approximation should exactly reproduce any polynomial field and we are free

to choose any polynomial basis as long as it is complete. Equation (18) in [132] expresses the derivatives of shape functions as

$$\phi_{I,k}(\mathbf{x}) = w_{I,k}\sum_{j}^{m} C_{jI} + w_I \sum_{j}^{m} C_{jI,k}(x)$$

where

$$C_{jI,k}(\mathbf{x}) = \left[\, q_{j,k}(\mathbf{x},\mathbf{x})q_j(\mathbf{x}_I,\mathbf{x}) + q_j(\mathbf{x},\mathbf{x})q_{j,k}(\mathbf{x}_I,\mathbf{x}) - b_{j,k}(\mathbf{x})C_{jI}(\mathbf{x}) \,\right] / b_j(\mathbf{x}) \quad (3.37)$$

$$b_{j,k}(\mathbf{x}) = \sum_{J}^{n} \left[\, w_{J,k}(\mathbf{x}) + q_j^2(\mathbf{x}_J,\mathbf{x}) + 2w_J(\mathbf{x})q_j(\mathbf{x}_J,\mathbf{x})q_{j,k}(\mathbf{x}_J,\mathbf{x}) \,\right] .$$

The term $q_{j,k}(\mathbf{x}_I,\mathbf{x})$ comes from the nodal coordinates of supporting nodes once a polynomial basis is defined, $q_{j,k}(\mathbf{x}_I,\mathbf{x})$ should be regarded as constant and therefore there is no need to differentiate.

3.5 Implementation and examples

Implementation of this new form of the EFG method takes the following steps:

1. Obtain the orthogonal basis using nodal data $\mathbf{U} = \{u_I, \mathbf{x}_I\}$
2. Compute the shape functions $\boldsymbol{\phi}_I$
   (a) Compute $\mathbf{P}(\mathbf{x})$ using the orthogonal basis
   (b) Compute A_{jj} using the obtained orthogonal basis
   (c) Compute $\boldsymbol{\phi}_I$ according to (3.5)
3. Compute $\boldsymbol{\phi}_{I,k}$
   (a) Compute derivatives of A according to (3.36b)
   (b) Compute derivatives of A^{-1} according to (2.10) with the A^{-1} as diagonal matrix
   (c) Compute derivatives of B according to (3.36c)
   (d) Get the value of $\boldsymbol{\phi}_{I,k}$ using (3.36).

The key difference from the implementation in [132] is that the shape functions are derived using the orthogonal basis approach while the shape function derivatives are obtained from the Pascal basis.

Example: Timoshenko's cantilever problem

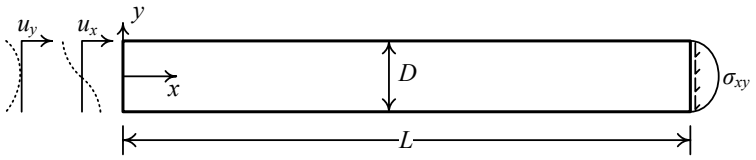


Figure 3.5: The model for the Timoshenko cantilever beam problem

Timoshenko's cantilever beam problem [128] (as shown in Figure 3.5) is widely used to validate meshless methods, although there are many cases of misuse particularly for demonstration of adaptive analysis. The essential boundary conditions are more complex than usually assumed. The analytical displacement field $\{u_x, u_y\}$ is cubic and is given as follows:

$$u_x = \frac{Py}{6EI}\left[(6L - 3x)\,x + (2+\nu)\,y^2 - \frac{3D^2}{2}(1+\nu)\right] \tag{3.38}$$

$$u_y = -\frac{P}{6EI}\left[3\nu y^2\,(L-x) + (3L-x)\,x^2\right]. \tag{3.39}$$

The stress field $\{\sigma_x, \sigma_y, \sigma_{xy}\}$ is given by

$$\sigma_{xx} = \frac{P\,(L-x)\,y}{l}, \tag{3.40a}$$

$$\sigma_{yy} = 0 \tag{3.40b}$$

$$\sigma_{xy} = -\frac{P}{2I}\left(\frac{D^2}{4} - y^2\right), \tag{3.40c}$$

The performance of the proposed new EFGM formulation is compared to the results from the DEM. The key difference between these two methods is the extra terms in the shape function derivatives for the former. To study the performance of the two methods, the polynomial basis is varied between linear and quadric, and coarse and fine background integration meshes are used (see Figure 3.6). In all cases the

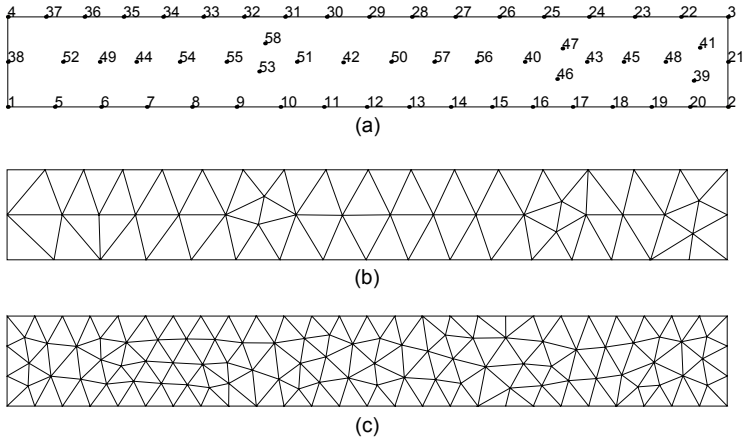


Figure 3.6: The nodal arrangement and background integration cells for the cantilever beam problem (a) nodal distribution (b) coarse background mesh (c) fine background mesh.

number of nodes is kept constant and elastic material parameters $E = 1 \times 10^5$ and $\nu = 0.25$ are used.

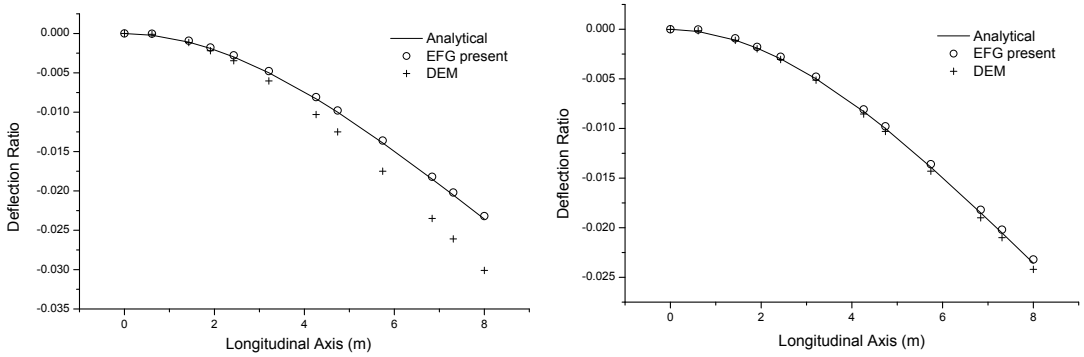
Deflections along the centreline of the cantilever (normalised with respect to its length) are plotted in Figures 3.7(a) to 3.7(d). The relative error of displacement in Equation (3.29) is used. Errors using this measure of the deflections along the x axis are listed in Table 3.2. The results in Table 3.2 show the following:

Cell	Basis	EFG present	DEM
coarse	linear	0.0190	0.2678
coarse	quadric	0.0140	0.0265
fine	linear	0.0131	0.0149
fine	quadric	0.0109	0.0128

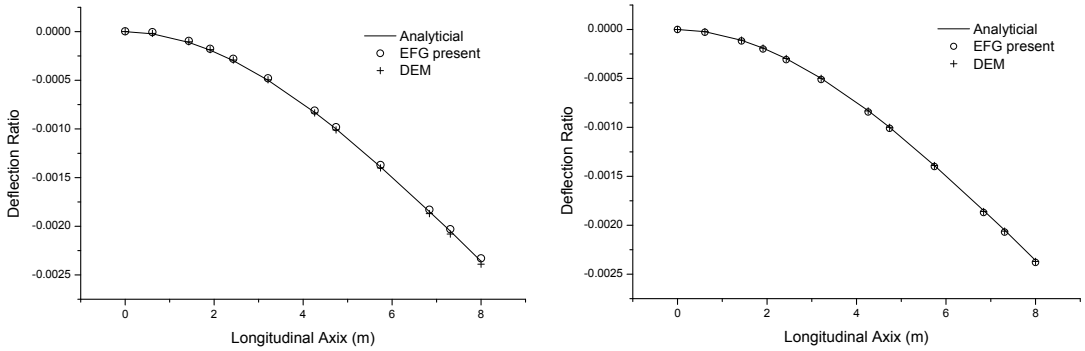
Table 3.2: Relative errors of deflection ratios by present EFG and DEM.

1. There is an evident improvement of accuracy using DEM as a direct result of more integration points or a higher basis.

2. The present EFG formulation is less sensitive to the choice of polynomial basis and the number of integration points. Given the same polynomial basis and



(a) linear basis and coarse background mesh. (b) quadric basis and coarse background mesh.



(c) linear basis and fine background mesh. (d) quadric basis and fine background mesh.

Figure 3.7: Deflection ratio results for the cantilever beam problem.

the same distribution of integration points, DEM is less accurate than the new EFG method.

3. Due to the high degree of overlapping of the MLS approximation, the new EFG method, which includes all the terms for the derivatives, can simulate the cubic displacement field for this problem even with a linear basis. Incomplete or truncated derivatives of the displacement approximation requires finer integration cells or a higher order basis to achieve the same accuracy.

4. The accuracy of the new EFG formulation and DEM can be improved by using more integration points or using higher polynomial basis. This does not conflict the first point although the new formulation of EFG is less sensitive to the order of basis and integration cells.

3.6 Discussion

This chapter has dealt with a number of issues affecting the "new" formulation of the EFG method proposed in [132]. The Schimdt orthogonalization process in the MLS approximation removes the need for matrix inversion which is desirable for efficient computational algorithms. However, the source of difficulties in inverting is not dealt with by adopting the orthogonal basis and inaccuracies remain. Terms in the shape function derivatives appear to have been omitted in the formulation in [132] which has been shown to lead to inaccuracies and failure to pass patch tests. It has been shown that the shape functions and shape function derivatives can be determined using different bases, a procedure which maintains the positive features of the use of orthogonalization while avoiding the inaccuracies of the original approach. The results also indicate that adaptive analysis can be performed without increasing the number of unknowns (i.e. nodes) either by (a) increasing the density of background integration cells or (b) increasing the basis order. The development described in this chapter has been implemented in the code used later in the thesis for fracture modelling.

Chapter 4

Error control and adaptivity in the EFGM

4.1 Introduction

Ideally, 3D fracture modelling using meshless methods should contain means of estimating errors and allowing adaptive analysis, as is now common with the FEM. This chapter therefore investigates discretisation error control and adaptivity in the EFGM highlighting the differences from the FEM. It is demonstrated that the (now) conventional procedures for error analysis and adaptivity used in the finite element method require careful application in the EFGM, otherwise competing sources of error work against each other. The discretisation error is here split into contributions arising from an inadequate number of degrees of freedom, e_h and from an inadequate basis e_p. Numerical studies given in this chapter show that for the EFGM this error cannot be easily split into component parts. Furthermore, we note that arbitrarily setting the size of nodal supports (as is commonly proposed in many papers) causes severe difficulties in terms of error control and solution accuracy. While numerical tests are performed only for the EFGM, the conclusions are applicable to other meshless methods based on the concept of nodal support.

4.2 Errors e_p and e_h

Errors in numerical modelling arise from a number of sources. Here we are concerned only with the discretisation error, defined as the error arising in the solution due to incomplete satisfaction of the governing equations and boundary conditions [144]. Other sources of error result from the quality of the mathematical model and round-off due to machine precision [144]. The focus of error control in the finite element method (FEM) has been on the discretisation error and there is a large body of literature on its measurement, and on adaptive schemes which make use of measures of error [145, 146]. Most work has focussed on defining a contribution to the solution error e_h associated with a measure of element size h that reduces at $O(h)$, or a contribution to error e_p from the order of basis function p used in the element that reduces as $O(h^p)$ [144]. This idea has been developed into the h-adaptive and p-adaptive procedures, where respectively element size is changed or the order of basis function is changed in order to improve the accuracy of results. The control of the error in the FEM is then achieved by a successive adaptive analysis guided by a knowledge of error distribution from an error estimation process.

On first view it would seem straightforward to apply these techniques for the FEM to meshless methods, however on further inspection some serious issues arise. Dealing with h-adaptivity first, examples of its use in the EFGM are given in [3, 98, 120, 143] where extra nodes are inserted in the domain at locations of high stress gradients. An adaptive analysis procedure based on the background integration cells (necessary in the EFGM) is proposed in [41] by gradual reduction of cell size. These techniques are linked since both reduce error by satisfying the governing equation at a larger number of points, so that the weak form becomes "stronger". The presence of these examples implies h-adaptive procedures with meshless methods as eminently feasible, considering that no remeshing is needed (as is the case with the h-adaptive FEM) and injecting extra nodes or quadrature points can be done at trivial additional computational cost. However some counter-examples to this assertion can be found in the literature. In [70], an h-adaptive analysis of background integration cells, based on an *a posteriori* error estimate is proposed. A cantilever beam is analyzed but there is clearly some spurious refinement close to the free end of the

beam, where no refinement would be expected (see Figure 9 in [70]). This shows that error does not necessarily reduce just by refining the background nodal distribution. In [147], two schemes of *a posteriori* error estimates are compared by studying a number of 2D potential and elasticity problems. The error estimates are effective in the examples, however, the error is not reduced as expected where extra nodes are added (e.g. see Figures 6-11 in [147]). In [136], a new background integration scheme is proposed to reduce the integration error. The idea is to generate background cells such that the influence of the nodal support is constrained to certain integration points. The background integration cell is partitioned according to the intersections of nodal supports. In this way any point inside the cell will have a consistent number of supporting nodes, meaning the basis will be less variable inside the cell. However, this scheme forces the EFGM to behave like the FEM, and some would regard this as veering too far away from a truly meshless approach.

p-adaptive procedures work to reduce the discrepancy between the exact solution space and the approximation space in which the discretised solution exists. If the approximation scheme contains a basis for the exact solution, then the discretisation error associated with p-adaptivity, $e_p = 0$. In these methods, several factors contribute to e_p, such as the choice of basis and the approximation or interpolation method. For example, a Fourier expansion, being based on a series of orthogonal sine and cosine functions, is preferable for simulating the heat conduction problem, while in fracture mechanics a radial basis function with angular enrichment is able to capture the singular stress field around a crack tip [143, 148]. In either case, the use of a low-order polynomial basis will lead to unsatisfactory results, however this is fairly standard for shape functions in the FEM. Another example can be found in the volume locking problem in the FEM where the basis is unable to represent certain deformation modes and hence the solution "locks". Interestingly, this problem is less widely reported in the EFGM [3] although it does appear in problems with volumetric constraints in plasticity [65]. In meshless methods such as the EFGM the underlying basis is usually a low order polynomial however this does not limit the space of the shape functions to be low order as the coefficients in the basis change for each point in the domain, and are determined by the nodal supports and the choice

of weighting function. Indeed an oft-quoted advantage of meshless methods is the smoothness of their approximations [36]. This feature of meshless methods can be easily demonstrated by that fact that in the EFGM even the use of a linear basis can accurately model a cubic field of displacement produced in Timoshenko's cantilever beam problem as shown in [120]. In contrast, using three-noded triangular finite elements it is impossible accurately to model this problem.

So, the most crucial point to note when considering error control is that in the FEM one can uncouple h- and p-adaptivity, but in meshless methods like the EFGM one cannot, i.e. changing the density of nodes both changes the error e_h and also changes the space of the shape functions and hence the error e_p. Error control (and hence adaptivity) is therefore harder to achieve in a meshless method as compared to the FEM. In special case, the EFGM shape functions can be made like the FEM shape functions by using constant weight function and setting nodal supports size compact. If so, the role of weight function disappears and approximation is no long "moving" as a point moves which is not the situation of the MLS approximation considered here. The general situation in the MLS approximation is that the weight function is non constant and the approximation changes as a point of interest moves in the domain. So the purpose of this chapter is to highlight these issues and to suggest some solutions. We focus on the EFGM throughout this chapter but the findings are applicable to all MLS-based methods. In §4.3 we summarise and compare interpolation in the FEM with approximation in the EFGM and provide a means of comparing nodal support parameters. In §4.4 an analysis of the factors affecting the approximation error in the EFGM is attempted, making comparisons to the limited published work elsewhere. In §4.5 numerical studies are presented both for errors in approximation of functions, and errors in analyses of elastic problems.

4.3 Interpolation in the FEM and approximation in the EFGM

The fundamental differences between the FEM and EFGM are the ways of approximating the field, consequently the difference relies in how the shape functions are

obtained. In the FEM, the interpolation is defined with in each element which is piecewise continuous. Therefore the displacement obtained is only continuous between elements while terms related to its derivatives such as stress are discontinuous. In the EFGM, the field is approximated in a "moving" manner and the approximated field obtained is not only continuous but also differentiable displacement field up to high order (smooth stress) in a global sense. Here we compare the ways of constructing the shape functions in these two methods and it will be shown that the FEM shape functions is only a subset of the EFGM shape function as has been indicated by Askes [52].

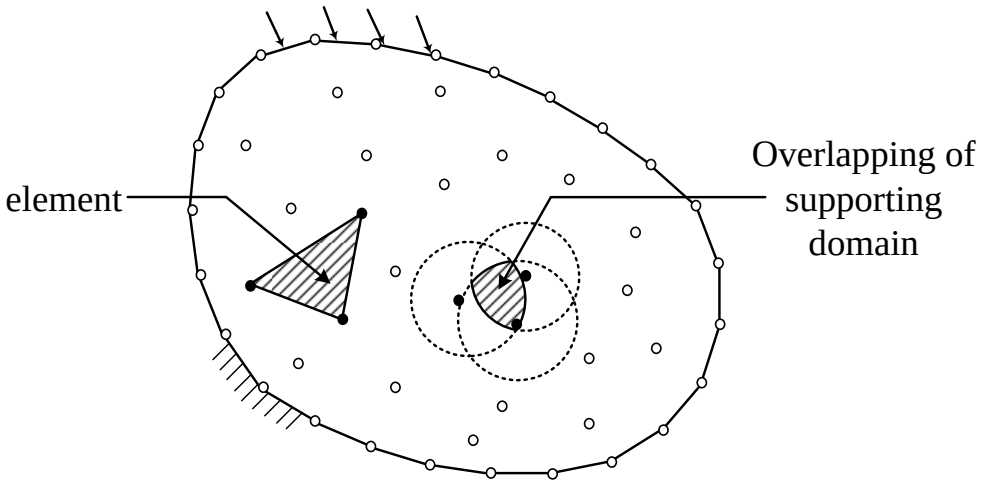


Figure 4.1: The concept of subdomain approximation: Ω_s corresponds to an element in the FEM and overlapping nodal supports in MLS-based meshless methods.

4.3.1 Interpolation in the FEM

Consider an unknown field $\mathbf{u}(\mathbf{x})$, e.g. displacement in solid mechanics, over domain Ω interpolated by a polynomial basis

$$u(\mathbf{x}) \approx u^h(\mathbf{x}) = \sum_j^m p_j(\mathbf{x}) a_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \tag{4.1}$$

where $u(\mathbf{x})$ denotes the approximate value of $u(\mathbf{x})$, $\mathbf{p}(\mathbf{x})$ is the set of appropriate polynomials of length m, e.g. $m = 3$ for a linear basis in 2D or a quadratic basis in 1D, and $\mathbf{a}(\mathbf{x})$ is a vector of coefficients $\mathbf{a}(\mathbf{x}) = \{a_1, \cdots, a_m\}^T$. The basis

$\mathbf{p}^T(\mathbf{x}) = \{p_1(\mathbf{x}), \ldots, p_m(\mathbf{x})\}$ is normally built using Pascal's triangle in 2D and Pascal's pyramid in 3D. The problem domain is divided into a set of smaller subdomains $\left\{\Omega_s : \Omega_s \in \Omega, \Omega \subseteq \cup_{s=1}^{S} \Omega_s\right\}$, where Ω_s provides a finite cover of Ω and S is the total number of divisions. In the FEM, the problem domain is discretised into elements which do not overlap; subdomains Ω_s become elements shown as the shaded triangle in Figure 4.1 and S corresponds to the total number of elements. Solving $\mathbf{a}(\mathbf{x})$ requires m constraints and in the FEM this is done by enforcing

$$u_I^h = u_I, \quad I = 1, \ldots, n \tag{4.2}$$

where $n \equiv m$ is the number of nodes of the element which contains $\mathbf{x}$, $u_I^h \equiv u^h(\mathbf{x}_I)$ and I is the index of vertex nodes. Equations (4.1) and (4.2) can be combined to give

$$\mathbf{P}\mathbf{a} = \mathbf{u}, \quad I = 1, \ldots, n \tag{4.3}$$

where $\mathbf{u} = \{u_1, \ldots, u_n\}$ is a vector collecting supporting nodal values and $\mathbf{P}$ is an $n \times m$ matrix defined by

$$\mathbf{P}^T = \Big[\mathbf{p}(\mathbf{x}_1), \ldots, \mathbf{p}(\mathbf{x}_n)\Big]_{n \times m} . \tag{4.4}$$

As $n \equiv m$, we can invert $\mathbf{P}$ to solve for $\mathbf{a}$

$$\mathbf{a}(\mathbf{x}) \equiv \mathbf{a} = \mathbf{P}^{-1}\mathbf{u} = \text{const} \tag{4.5}$$

and so Equation (4.1) becomes

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a} = \underbrace{\mathbf{p}^T(\mathbf{x})\mathbf{P}^{-1}}_{\mathbf{\Phi}}\mathbf{u} \tag{4.6}$$

where $\mathbf{\Phi} = [\Phi_1, \Phi_2, \cdots, \Phi_n]$ is the vector of finite element shape functions. It should be noted that when isoparametric element is used $\mathbf{a}(\mathbf{x})$ in Equation (4.5) is constant only in local coordinate system and non constant in global coordinate system. However given the element topology and nodal values, $\mathbf{a}(\mathbf{x})$ is uniquely defined within each element and the form of interpolation is invariant within each element.

4.3.2 The moving least squares approximation in meshless methods

MLS-based meshless methods, like the EFGM, are developed using shape functions just as with the FEM. The difference is in the means by which the shape functions are determined. Though the field is approximated in the same way as in the FEM by using Equation (4.1), the unknown coefficients $\mathbf{a}(\mathbf{x})$ are solved by the MLS approximation which are different from the interpolation in the FEM. Through the process of minimizing *weighted* residual (refer to Equation (2.3)), the coefficients $\mathbf{a}(\mathbf{x})$ are given as

$$\mathbf{a}(\mathbf{x}) = \left(\mathbf{P}^T\mathbf{W}\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{W}\hat{\mathbf{u}} \ . \tag{4.7}$$

Unlike the $\mathbf{a}$ in Equation (4.5), the coefficients $\mathbf{a}(\mathbf{x})$ here are dependent on $\mathbf{x}$ and vary as $\mathbf{x}$ moves through regions of overlapping support. J is minimized but is not necessarily zero. For the special case of $J = 0$ Equation (4.7) becomes equivalent to (4.5) and $u_I^h \equiv \hat{u}_I$. In this (unusual) case, $u^h(\mathbf{x})$ becomes an interpolation of $\mathbf{u}$ although the fact that $J = 0$ alone is not a good indicator of the quality of fitting, for the same reasons as for the FEM. Comparing MLS-based meshless methods to the FEM we can see that in the latter, a node has influence on a point if the point falls within the elements sharing that node, and nodes are connected with each other through the element topology. In an MLS-based meshless method, a node exerts influence on any point that falls within the support of the node (Figure 4.1). Nodes are "connected" with each other if their nodal supports overlap. Thus, the concept of a finite element can be compared to the overlapping of supports, shown in 2D as the intersection of circles in Figure 4.1.

4.4 Factors affecting the approximation error

In §4.2 we defined error contributions e_p and e_h and indicated that for meshless methods it is not possible to measure these independently. In this section, factors contributing to a general error measure e_u are analyzed. We consider an arbitrary

problem domain with an unknown field which can be expressed in an exact form as

$$u(\mathbf{x}) = \hat{\mathbf{p}}^T(\mathbf{x})\hat{\mathbf{a}}(\mathbf{x}) \tag{4.8}$$

where $\hat{\mathbf{p}}(\mathbf{x})$ is a polynomial basis vector of length $\hat{m}$ and $\hat{\mathbf{a}}$ is a vector of $\hat{m}$ coefficients. It should be noted here the "$\wedge$" symbol no longer represents the fictitious values as in §4.3.2. In the MLS approximation, the field is approximated as

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{a}(\mathbf{x}) \ . \tag{4.9}$$

Generally, the exact solution is not polynomial so that $\hat{m} \neq m$ and $\hat{\mathbf{p}}(\mathbf{x})$ has some other bases noted as $\bar{\mathbf{p}}$ which are not contained in $\mathbf{p}(\mathbf{x})$ so that

$$\hat{\mathbf{p}}^T(\mathbf{x}) = \left\{\mathbf{p}^T(\mathbf{x})|\bar{\mathbf{p}}^T(\mathbf{x})\right\} \tag{4.10}$$

and the length of $\bar{\mathbf{p}}(\mathbf{x})$ is noted as $\bar{m}$ and the relation holds $\hat{m} = m + \bar{m}$. Correspondingly, $\hat{\mathbf{P}}$ can also be decomposed into two parts, $\mathbf{P}$ and $\bar{\mathbf{P}}$

$$\hat{\mathbf{P}}_{n\times\hat{m}} = \left[\mathbf{P}_{n\times m}|\bar{\mathbf{P}}_{n\times(\hat{m}-m)}\right] = \left[\mathbf{P}|\,\mathbf{0}\,\right] + \left[\mathbf{0}\,|\bar{\mathbf{P}}\right] \ . \tag{4.11}$$

The coefficients $\hat{\mathbf{a}}$ can also be decomposed into two parts

$$\hat{\mathbf{a}} = \begin{Bmatrix} \tilde{\mathbf{a}} \\ - \\ \bar{\mathbf{a}} \end{Bmatrix} = \begin{Bmatrix} \tilde{\mathbf{a}} \\ - \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \mathbf{0} \\ - \\ \bar{\mathbf{a}} \end{Bmatrix} \tag{4.12}$$

where $\tilde{\mathbf{a}}$ is a vector of length m and $\bar{\mathbf{a}}$ of length $\bar{m}$. The coefficients $\bar{\mathbf{a}}$ correspond to $\bar{\mathbf{p}}(\mathbf{x})$ and $\tilde{\mathbf{a}}$ to $\mathbf{p}(\mathbf{x})$. It should be noted that in general $\tilde{\mathbf{a}}$ in the exact solution are not the same as $\mathbf{a}$ in the MLS approximation. Extra information is included in $\mathbf{a}$ due to the absence of $\bar{\mathbf{p}}(\mathbf{x})$ in the basis function. The exact solution at any point is given by Equation (4.8), thus the exact solution of a node in support of $\mathbf{x}$ is $u(\mathbf{x}_I) = \hat{\mathbf{p}}^T(\mathbf{x}_I)\hat{\mathbf{a}}$ and for all the nodes in support

$$\hat{\mathbf{U}} = \begin{bmatrix} \hat{\mathbf{p}}^T(\mathbf{x}_1) \\ \hat{\mathbf{p}}^T(\mathbf{x}_2) \\ \vdots \\ \hat{\mathbf{p}}^T(\mathbf{x}_n) \end{bmatrix} \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_m \end{Bmatrix} = \hat{\mathbf{P}}\hat{\mathbf{a}} \ . \tag{4.13}$$

Substituting (4.13) for the fictitious $\hat{\mathbf{U}}$ in Equation (4.7) gives

$$\begin{aligned}
\mathbf{a}(\mathbf{x}) &= \left(\mathbf{P}^T\mathbf{W}\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{W}(\hat{\mathbf{P}}\hat{\mathbf{a}}) \qquad (4.14)\\
&= \left(\mathbf{P}^T\mathbf{W}\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{W}\left([\mathbf{P}\,\bar{\mathbf{P}}]\begin{Bmatrix}\tilde{\mathbf{a}}\\ \bar{\mathbf{a}}\end{Bmatrix}\right) \qquad \text{(from (4.11) and (4.12))}\\
&= \left(\mathbf{P}^T\mathbf{W}\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{W}\mathbf{P}\tilde{\mathbf{a}} + \left(\mathbf{P}^T\mathbf{W}\mathbf{P}\right)^{-1}\mathbf{P}^T\mathbf{W}\bar{\mathbf{P}}\bar{\mathbf{a}}\\
&= \tilde{\mathbf{a}} + \underbrace{\left(\mathbf{P}^T\mathbf{W}\mathbf{P}\right)^{-1}(\mathbf{P}^T\mathbf{W})\bar{\mathbf{P}}}_{\mathbf{M}}\,\bar{\mathbf{a}}
\end{aligned}$$

where $\tilde{\mathbf{a}}$ is exactly reproduced in the MLS approach. It can be seen from Equations (4.10), (4.12) and (4.14) that if $\mathbf{p}(\mathbf{x})$ is a sufficient basis for the exact solution, then $\mathbf{M}$ vanishes and $\mathbf{a} = \tilde{\mathbf{a}}$. The second term on the *r.h.s.* is the MLS coefficient corresponding to $\bar{\mathbf{a}}$. Now u^h becomes

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\tilde{\mathbf{a}} + \mathbf{p}^T(\mathbf{x})\mathbf{M}(\mathbf{x})\bar{\mathbf{a}} \qquad (4.15)$$

and the exact solution becomes

$$u(\mathbf{x}) = \tilde{u}(\mathbf{x}) + \bar{u}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\tilde{\mathbf{a}} + \mathbf{p}^T(\mathbf{x})\bar{\mathbf{a}} \qquad (4.16)$$

where $\tilde{u}(\mathbf{x})$ is the part of the real solution which can be produced exactly. To estimate the relative error between $u^h(\mathbf{x})$ and $u(\mathbf{x})$, we define e_u

$$e_u = \|u^h(\mathbf{x}) - u(\mathbf{x})\| \qquad (4.17)$$

where $\|\cdot\|$ is the L_2 norm of a vector (refer to Equation (1.21)). Then the relative exact error becomes

$$e_u = \|\mathbf{p}^T(\mathbf{x})(\mathbf{I} - \mathbf{M}(\mathbf{x}))\bar{\mathbf{a}}\| \qquad (4.18)$$

where $\mathbf{I}$ is identity matrix. Equation (4.18) shows that the error from the MLS approximation is problem dependent because of $\bar{\mathbf{a}}$. Given a certain nodal arrangement, the choice of basis function and size of nodal support determines $\mathbf{W}$ and hence e_u.

Rigorous analysis of errors in MLS-based approximations beyond that outlined above is challenging. In [149], an error estimate is given in terms of size of nodal

support. The bound of the error in Sobolev spaces for the MLS approximation and its first derivatives are given respectively as

$$\|u^h(\mathbf{x}) - u(\mathbf{x})\| \leq CR^{m+1}\|u_I - u(\mathbf{x})\|_\infty \quad \mathbf{x} \in \Omega \tag{4.19}$$

and

$$\|u^h_{,k}(\mathbf{x}) - u_{,k}(\mathbf{x})\| \leq \frac{CR^{m+1}}{R}\|u_I - u(\mathbf{x})\|_\infty \quad \mathbf{x} \in \Omega. \tag{4.20}$$

where R is the size of nodal support and C is a constant independent of R. Here, a Sobolev space is a vector space of functions equipped with L_p norms for the functions and their derivatives up to a given order. $\|\cdot\|_\infty$ is the maximum norm and for a vector $\{v_1, v_2, \cdots v_n\}$, $\|\|_\infty = \max(|v_1|, |v_2|, \ldots |v_n|)$. Equations (4.19) and (4.20) indicate that changes in R affect both the error of the field and also of its derivatives. Error estimates similar to Equations (4.19) and (4.20) are also used in [150]. It must be mentioned that Equation (4.19) is derived with some assumptions such as shape function values being positive for all nodes in support, and the existence of a Lagrangian basis passing through all those nodes. These assumptions are too restrictive to be satisfied in real analysis and the error estimates are however too generic to be directly applied in practice, however they still provide some guidance. Firstly, the MLS approximation requires compact support to reduce the field error. Secondly, the size of nodal support plays a role similar to nodal spacing in the error approximation. Thirdly, the choice of the size of nodal support is a choice between the approximation of the field and the approximation of the derivatives of the field. The presence of the parameter C on the *r.h.s.* of Equations (4.19) and (4.20) indicates once again the error to be problem dependent.

An alternative approach yielding similar conclusions is given in [151] which is concerned with the link between the error estimate of the MLS approximation and the condition number of the matrix $\mathbf{A}$ (corresponding to $\mathbf{P}^T\mathbf{W}\mathbf{P}$ here), which is dependent on nodal arrangement. The error converges quickly with respect to h for a "good" nodal arrangement, indicating a low condition number for $\mathbf{A}$. It can be shown that $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{P})$ and $\mathbf{A}$ becomes singular or ill-conditioned when $\text{rank}(\mathbf{P}) < m$. Thus it is clear that when the size of the nodal support changes, the nodal arrangement for certain points will change and hence the approximating

power of the MLS method. This is identical to the discussion concerning $\mathbf{M}$ in Equation (4.18) above. A common way to set the support size is to first choose a parameter related to the nodal spacing (d_{min} or 3rd/4th nearest node, say) and secondly scale this by a certain factor (see Equation (2.19)). The choice must yield an invertible $\mathbf{A}$ matrix at every quadrature point in order that the problem can be solved successfully. However, the consequences of choosing different values and the ability to judge *a priori* if the choice is most appropriate for the problem in terms of the error have not been well addressed yet. Although this is not a new issue and has been noticed and tested in [3, 119], the effects of the scale size of nodal support on the errors and the relation between contributions to error from e_h and e_p has not yet been clearly understood and discussed before. In the following discussion and numerical studies, we focus on how α, which is defined in Equation (2.19), plays a role in the error of results. It is the trends or behavior that are of interest rather than specific values of α for specific cases.

4.5 Numerical studies

In §4.4, the sources of approximation errors using the MLS approach were analyzed. It was shown that the scale factor α has an important influence on the error. Although it is impossible exactly to compute the error from α, it is possible to investigate this link numerically. In this section, we report the results of numerical studies carried out to investigate this link. Firstly, one-dimensional curve fitting problems are used where $\hat{u}_I$ are known values, and secondly a range of elasticity problems are solved using the EFGM.

4.5.1 One-dimensional curve fitting problems

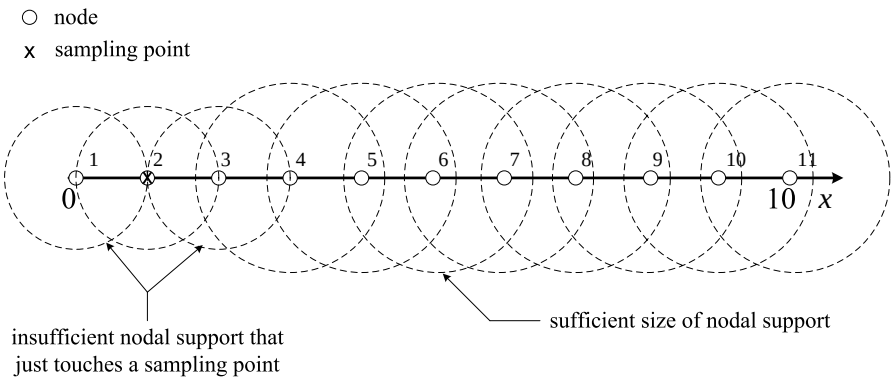


Figure 4.2: The nodal arrangement for 1D data fitting problems and nodal support.

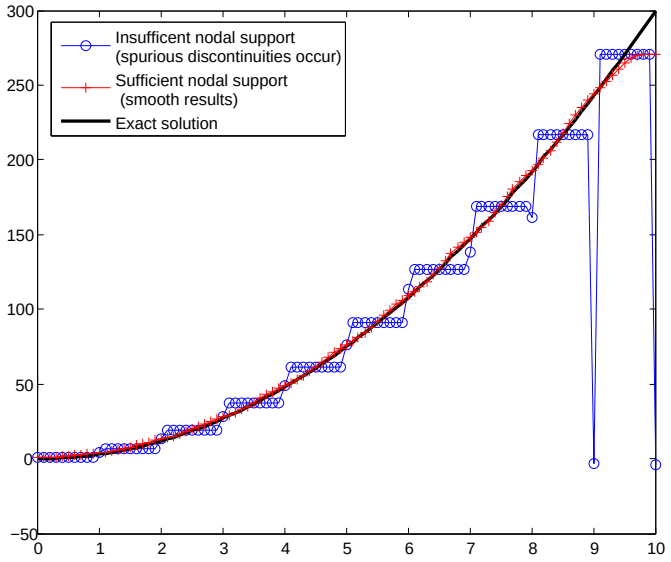
We first consider errors in function approximations using MLS. We consider a one-dimensional domain $x \in [0, 10]$ with 11 uniformly distributed nodes at unit spacings as shown in Figure 4.2. Tests are carried out to assess the ability of the MLS approximation to match these specified functions.

$$^1u = x^3; \quad ^2u = \sin x; \quad ^3u = e^x,$$

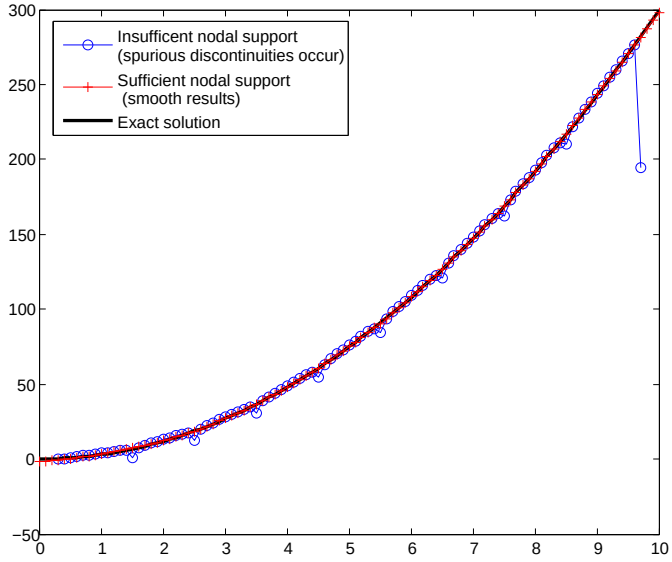
whose derivatives are

$$^1u_{,x} = 3x^2; \quad ^2u_{,x} = \cos x; \quad ^3u_{,x} = e^x.$$

Function ^1u is a polynomial, where $u_{,x}$ is one order lower than u; function ^2u has alternating signs between u and $u_{,x}$ and ^3u is identical to its derivative. For each of these three functions, two curves are produced by the MLS approach according to §2.2; one fitting the function and the other fitting the derivative. In all cases 100 sampling points are used to span the 1D domain. The exact function values at the points are used for the fictitious nodal values $\hat{u}_I$ for the MLS approximation. Then shape functions are determined and hence the approximation $u^h(\mathbf{x})$. The derivative approximation is obtained by taking the same fictitious values and using the derivatives of the shape functions. The approximation error e_u in each case is calculated as $e_u = \|^i u^h(\mathbf{x}) - {}^i u(\mathbf{x})\|$ over the 100 points and $e_{u_{,x}}$ similarly. The purpose of the tests is to find the relation between the errors from the MLS approach depending on the size of the nodal support: i.e. how well can a given function be approximated; does accuracy change with different orders of basis functions and different weight functions, and can problem dependency be verified as discussed in



(a) linear basis



(b) quadratic basis

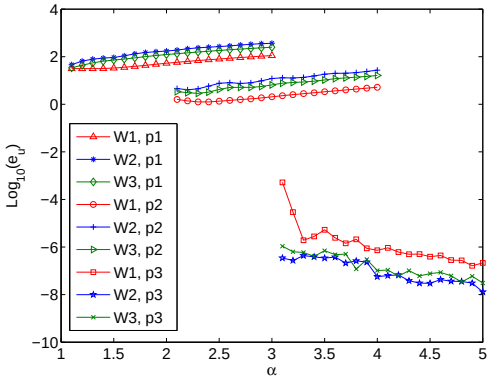
Figure 4.3: A comparison of the first order derivatives of ${}^1u$ with varying support showing discontinuities.

§4.4. Throughout the tests, the nodal spacing is unchanged and only nodal supports are varied. The minimum size of nodal support is dependent on the choice of basis function used; a linear basis requires two supporting nodes. Three nodes are needed for a quadratic basis. The nodal support size is set according to Equation (2.19) and each of the tests starts with an α where all the sampling points have sufficient nodes in support. As the nodal spacing here is 1, c_I is 1 for all the nodes, and α using a linear basis should be larger than 1, otherwise $\mathbf{A}$ will become singular or ill-conditioned. As a demonstration of this behaviour Figure 4.3 shows curves produced in the test for ${}^1u_{,x}$. Spurious jumps in the curves are produced when the nodal support is insufficient to match the basis function in parts of the domain, although a curve is still produced.

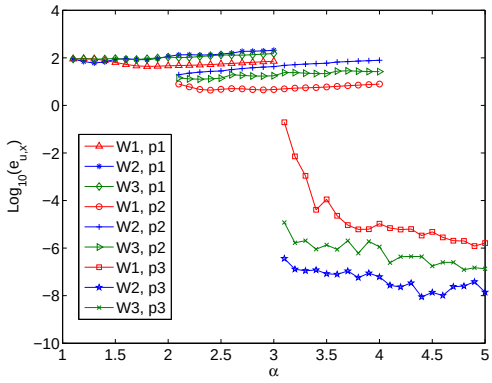
The main test results for 1D curve fitting are now described. Test results are shown in Figure 4.4 for each function, weight function w^1, w^2, w^3 and basis, linear ($p1$), quadratic ($p2$) and cubic ($p3$). Errors are given on a log vertical scale against α. To accentuate differences between functions the error scales are different. For comparisons between functions the linestyles used are universal. Apart from two sets of results with very low errors, all plots show general increases in error with α; as the nodal supports grow so the local nature of the approximation is reduced. The two sets of results with very low errors (${}^1u$ with a cubic basis) can be explained as the basis exactly reproduces the function and its derivative, and the error we see is round-off and machine-based.

In almost all cases the lowest errors are found using weight function w^1 (rational), followed by w^3 (conical) and then w^2 (exponential). The only exception to this is for the case of the derivative of ${}^3u$ where w^1 gives higher errors than either of the other weight functions. Variations in errors for the functions themselves are generally monotonically increasing with α although for functions ${}^2u$ and ${}^3u$, the rational weight function w^1 shows an optimum α value close to, but slightly larger than the minimum α possible. The difference, however, is relatively small so that the general conclusion, that low α gives low field variable error, is sound.

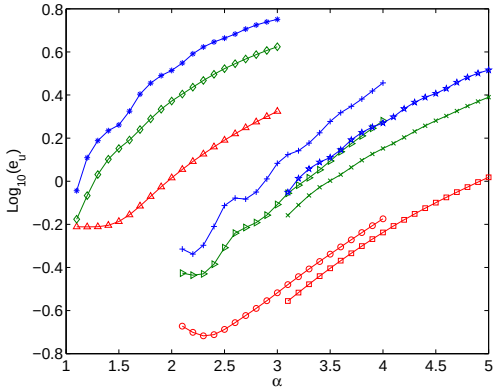
The approximations to derivatives show quite different behaviour to that for the functions themselves. Ignoring the round-off error cases mentioned above, 14 of 24



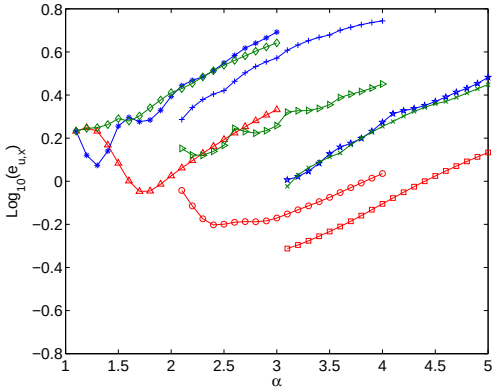
(a) u^h error for ${}^1u$



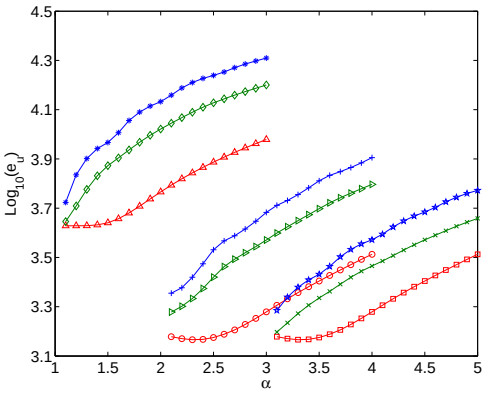
(b) $u^h{}_{,x}$ error for ${}^1u$



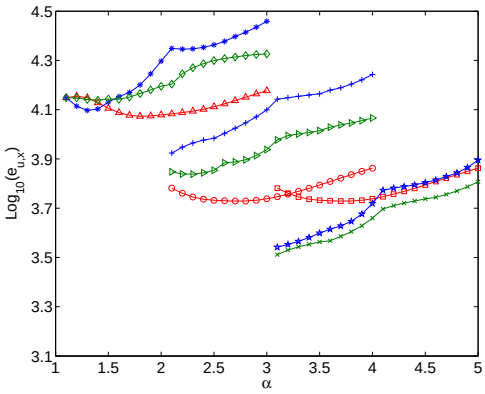
(c) u^h error for ${}^2u$



(d) $u^h{}_{,x}$ error for ${}^2u$



(e) u^h error for ${}^3u$



(f) $u^h{}_{,x}$ error for ${}^3u$

Figure 4.4: Errors in u^h and $u^h{}_{,x}$ against α for 1D curve fitting problems. (Linestyles are common to all plots)

derivative approximations show clear minimums in the plots of error against α with those based on w^1 particularly in evidence. In most cases these minimum errors occur at values of α significantly larger than the minimum value necessary. The conclusion to be drawn is that, unlike the function approximation, for the derivative approximation the minimum α does not lead to the minimum error, and that while the use of w^1 appears to give better results for the function approximation, its use leads to variations in derivative error that are non-monotonic. Therefore if both a function approximation and its derivative are required it may prove harder to control error via change in α. Another interesting observation is that a higher order basis does not necessarily guarantee a better solution if the basis function does not contain the correct "information". This is evident in Figure 4.4(f) where for a small range of α the quadratic basis performs better than the cubic basis. The choice of weight function in MLS-based meshless methods is often thought not to be critical to the quality of results, e.g. [121]. However here we can see that the approximations at the heart of these methods are clearly affected by the choice of weight function. However, it may be that the approximations required in boundary value problems are so much more complex than the functions approximated here, that the difference in behaviour is attenuated.

4.5.2 Testing on boundary value problems

The numerical study is now extended to boundary value problems, demonstrating the error control possible in typical infinitesimal strain, linear elasticity problems when using the EFGM. Generally, the computation process is similar to the data fitting problems above due to the use of the MLS procedure to obtain the shape functions. However, in a data fitting problem, the fictitious nodal values $\hat{u}_I$ are given already and the problem is to calculate $u^h(\mathbf{x})$ throughout the domain. The solution process does not involve the derivatives at the point of interest, although these were studied in the section above. In an elasticity analysis, $\hat{u}_I$ become the *unknown* displacements at nodal degrees of freedom and are determined via discretisation of a governing PDE of the problem using a weak form, such as a Galerkin approach (in the EFGM), Petrov-Galerkin approach (in the MLPG) or a variational method

[130]. Such a solution requires derivatives $u^h{}_{,x}$. The procedure for elasto-static analysis using the EFGM is fully described in §2.3 and 2.4, and will not be repeated here.

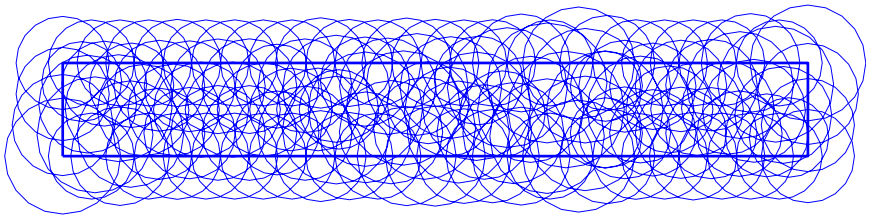


Figure 4.5: Plot of nodal support for the beam problem with $\alpha = 1.5$.

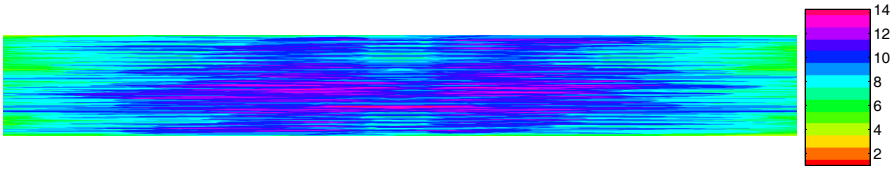


Figure 4.6: Plot of number of nodes in support for the beam problem ($\alpha = 1.5$).

The first problem is Timoshenko's cantilever beam [128] as already used in §3.5. A large number of EFGM analyses were carried out to study the relation between e_u and α when employing a lower order (and hence cheaper) linear basis function, varying material properties and natural boundary conditions (to investigate problem dependency). A modified version of the error measure in (4.17) is used, i.e.

$$e_u = \left\{ \int_\Omega \left(u^h(\mathbf{x}) - u^*(\mathbf{x}) \right)^2 \mathrm{d}\Omega \right\}^{\frac{1}{2}} , \tag{4.21}$$

with a similar expression for the derivative error. The integrals in (4.21) are determined by Gaussian quadrature. Examples of the nodal supports and numbers of

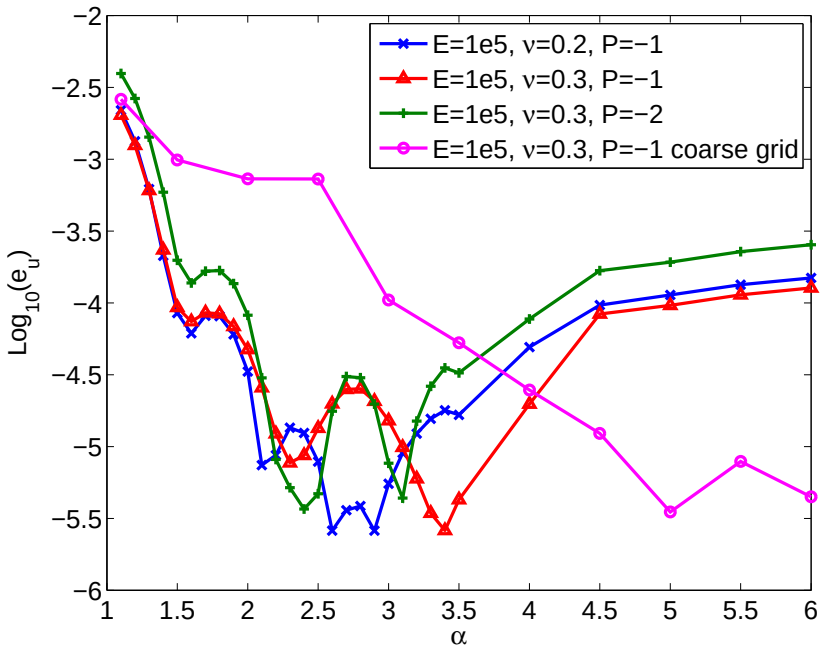


Figure 4.7: Error in predicted displacement in the beam problem against α.

nodes in support for this problem are shown in Figures 4.5 and 4.6. Figure 4.7 shows the variation in e_u, plotted on a log scale against α. Three plots are for the case of a fine nodal grid while a fourth is for a coarse grid. Comparing the former three results we can see that the optimum value for α lies well away from the minimum required, i.e. $2.0 < \alpha < 3.5$. The variation in error is erratic and non-monotonic, especially close to the minimums. Changing both the material properties (i.e. ν from 0.2 to 0.3) and the applied traction boundary condition (i.e. P), alters the behaviour of the error with respect to α indicating clear problem dependency. The fourth result, for the coarse grid, is shown to demonstrate that a finer mesh does not guarantee a better accuracy unless the nodal support is properly chosen. At high values of α the coarse grid wins. This confirms the hypothesis above, that h-refinement may on its own not work and attention must be paid to the coincident p-refinement. An alternative *a posteriori* error estimate for the EFGM is proposed in [4, 57] as $e_{uf} = ||u^h(\mathbf{x}_I) - \hat{u}(\mathbf{x}_I)||$, i.e. the difference between the fictitious nodal value and MLS approximation value at the node. When applied to the same data set as used to plot Figure 4.7 this error measure leads to the plots in Figure 4.8. e_{uf}

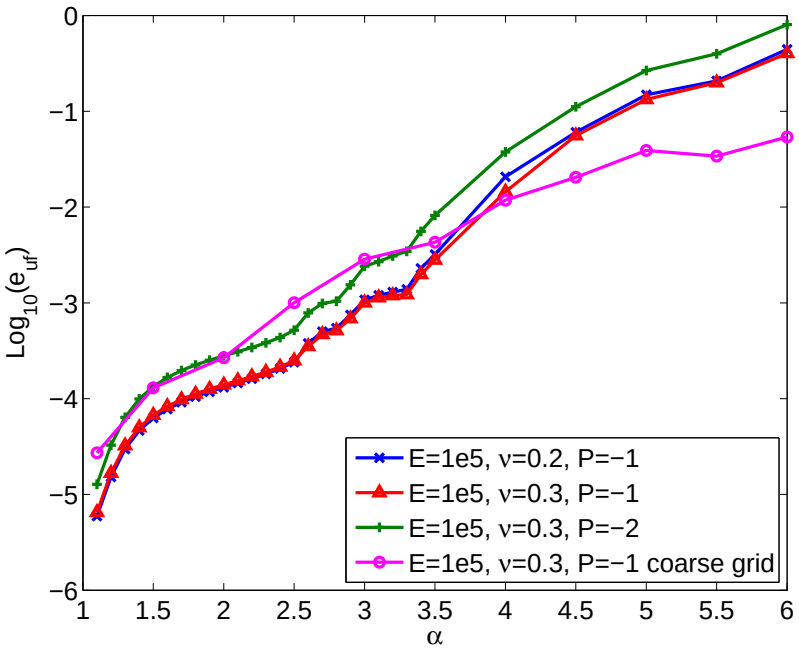


Figure 4.8: Fictitious nodal error e_{uf} [4] against α for the beam problem.

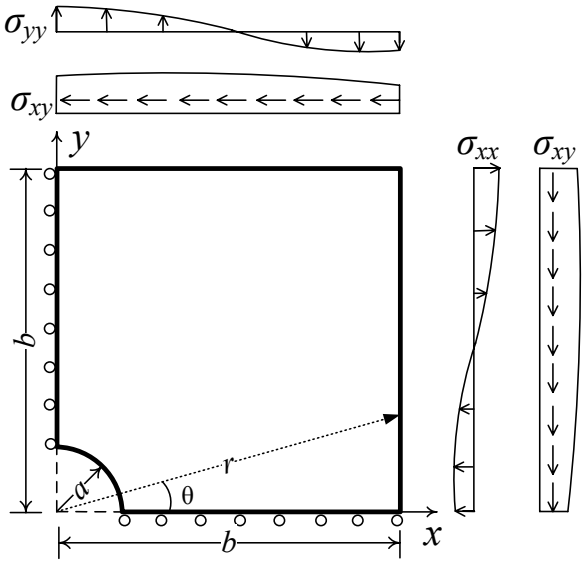
is seen to increase with α for all analyses, with no problem or grid dependency, a quite different picture to the behaviour when employing the error measure e_u and an indication that error measures must be chosen carefully for these methods.

A second elasticity problem, an infinite plate with a circular hole of radius $a = 1$ subjected to a far field traction in the x direction, is also analysed using the EFGM. Material properties are $E = 1000, \nu = 0.3$ and plane strain conditions are assumed. Due to symmetry, a finite portion of the plate $b = 5$ is taken for analysis as shown in Figure 4.9. This problem has been used previously in [3, 132] with the EFGM to study error convergence with respect to nodal spacing and different orders of basis functions, since it has an analytical solution for stresses and displacements as

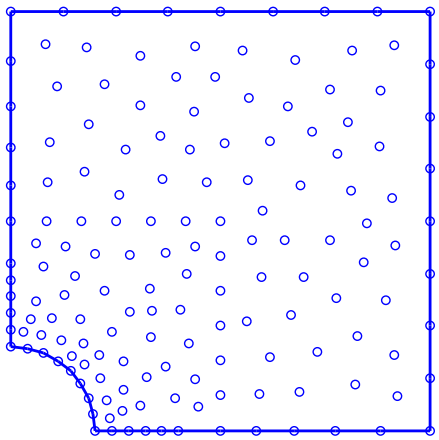
$$\sigma_{xx} = 1 - \frac{a^2}{r^2}\left(\frac{3}{2}\cos(2\theta) + \cos(4\theta)\right) + \frac{3a^4}{2r^4}\cos(4\theta) \tag{4.22}$$

$$\sigma_{xy} = -\frac{a^2}{r^2}\left(\frac{1}{2}\sin(2\theta) + \sin(4\theta)\right) + \frac{3a^4}{2r^4}\sin(4\theta) \tag{4.23}$$

$$\sigma_{yy} = -\frac{a^2}{r^2}\left(\frac{1}{2}\cos(2\theta) - \cos(4\theta)\right) - \frac{3a^4}{2r^4}\cos(4\theta) \tag{4.24}$$



(a) Symmetric model and boundary conditions



(b) Nodal arrangement

Figure 4.9: A portion of the infinite plate with a circular hole subjected to an unidirectional tensile load of 1.0 in the x direction.

$$u = \frac{a}{8G}\left\{\frac{r}{a}(\kappa+1)\cos\theta + \frac{2a}{r}\left[(1+\kappa)\cos\theta + \cos(3\theta)\right] - \frac{2a^3}{r^3}\cos(3\theta)\right\} \quad (4.25)$$

$$v = \frac{a}{8G}\left\{\frac{r}{a}(\kappa-3)\sin\theta + \frac{2a}{r}\left[(1-\kappa)\sin\theta + \sin(3\theta)\right] - \frac{2a^3}{r^3}\sin(3\theta)\right\} \quad (4.26)$$

where G is the shear modulus and κ is the Kolosov constant where $\kappa = 3 - 4\nu$ for a plane strain assumption, and r and θ are defined as shown in Figure 4.9(a). Boundary conditions specified in [128] are applied on the right and upper edges as shown in Figure 4.9(a). An EFGM model with 157 nodes is used for all tests as shown in Figure 4.9(b) and an example of the nodes in support across the domain is shown in Figure 4.10. Figure 4.11 shows the variation in error, calculated as for the beam problem, for a large number of analyses in which α is varied. For displacements it is clear that either linear or a quadratic bases give similar levels of accuracy providing one chooses α correctly, but of course it is not possible to do this *a priori*. However, choosing a small value for α, as is often suggested, would have disastrous results particularly for the quadratic basis. In that case low values of α clearly do not provide sufficiently detailed information to produce good coefficients for the quadratic basis, and so the error rises. Figure 4.12 shows the stress concentration factor (SCF) for σ_{xx} at the hole edge on the y-axis (analytical solution is 3) derived from the displacement results, again plotted with respect to α. The picture here is a little more comforting, as the variation from the analytical solution is less marked than when considering displacements. This plot confirms the surprising result in [3] that a quadratic basis performs worse than a linear basis, however this is only true at low values of α, as covered in that paper. Over a larger range it can be seen that the best choice of basis changes.

4.6 Discussion

Meshless methods such as the EFGM offer the enticing prospect of easy adaptive analysis since they avoid the need for remeshing of elements. However any adaptive procedure must have a robust error measure and a clear procedure for refinement or derefinement. The control of error in the EFGM has been shown here to be a competition between the field and its derivatives. The limited analytical studies of

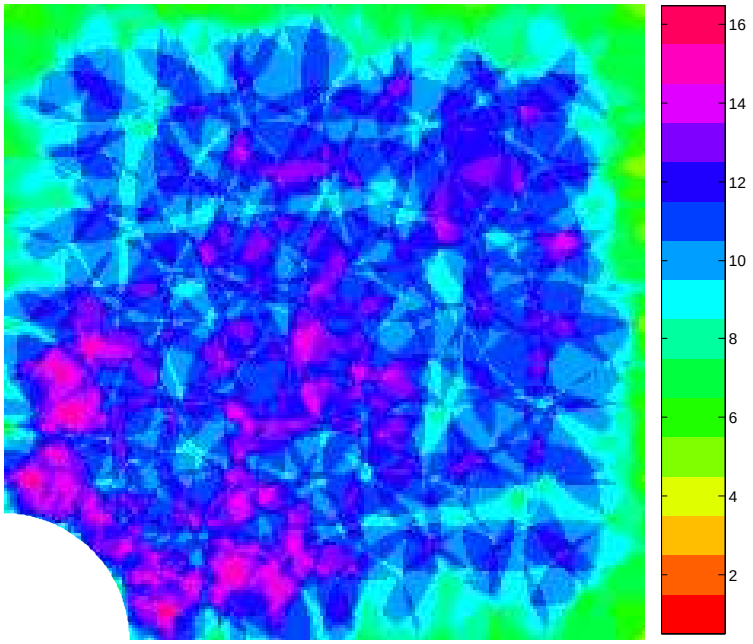


Figure 4.10: Plot of number of nodes in support for the plate problem ($\alpha = 1.5$).

this problem to date indicate that the latter requires the size of support to be large while the former requires a smaller support, although more work needs to be done in this area. For simple data fitting, the numerical studies here have confirmed the hypothesis that nodal support should be as compact as possible providing the requirement on minimum number of supporting nodes is met. In these problems there appear to be optimal choices of weight functions. However, for boundary value problems, such as elasticity, minimizing the error is a balance between approximating the field and its derivatives, and it is difficult to assess appropriate refinement or basis change strategies without conducting some sensitivity analyses in which the grid, basis and nodal support measure are changed. The results from the elasticity problems show that the relation between error and the dimensionless measure of size of nodal support α is also dependent on the problem field itself. A fine nodal arrangement can generally achieve better accuracy than a coarse arrangement, however, this is only valid when the value of α is properly chosen. An arbitrary choice of α from a suggested range can result in a large error. Therefore, the control of α should precede successive adaptivity analyses.

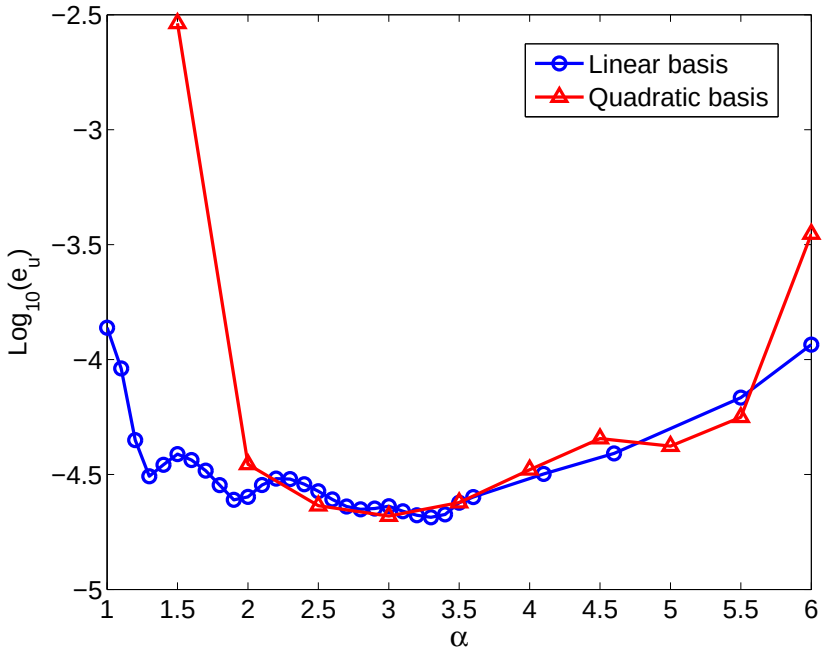


Figure 4.11: Error e_u against α for the infinite plate problem.

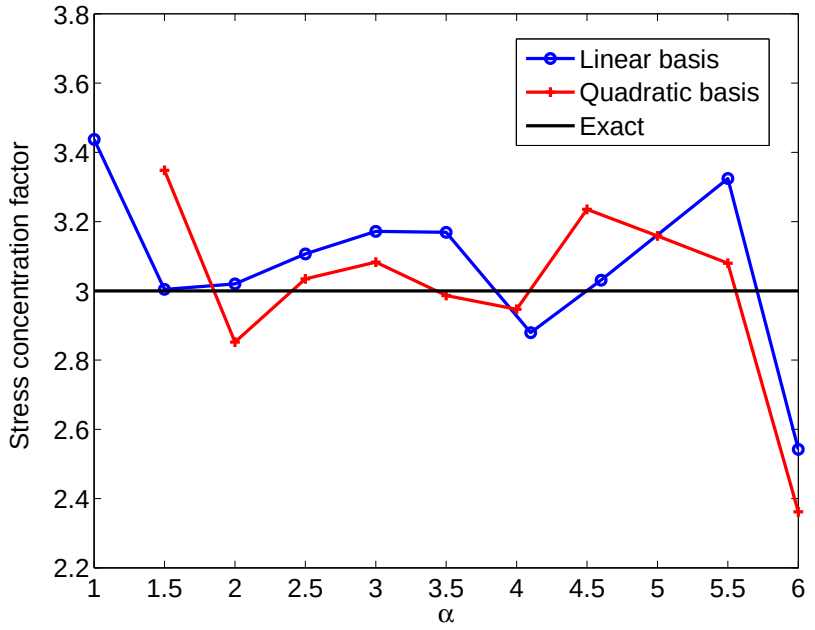


Figure 4.12: The stress concentration factor against α for the infinite plate problem.

Chapter 5

The EFGM for fracture modelling

5.1 Introduction

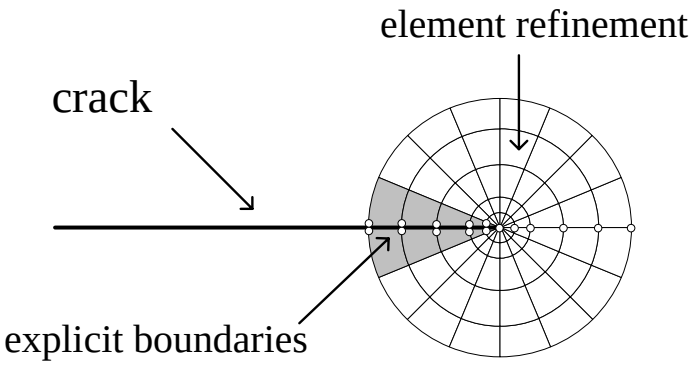
Previous chapters have focused on the use of the EFGM for continuum modelling. Here we introduce the two additional modelling features particular to fracture, namely singular stress fields and a displacement jump across a crack. We will review some existing methods in the EFGM for introducing these features, giving the term "enrichment" to cover both, as is common in the literature and noting that these methods can be categorized into extrinsic or intrinsic enrichment. While definitions of these terms vary between publications it is useful, we think, to distinguish the two as one requiring extra unknowns (extrinsic) and the other requiring no extra unknowns (intrinsic). Apart from the field approximation, fracture parameters such as the stress intensity factors (SIFs) are needed for the analysis of crack state and to judge crack propagation. The calculations are preformed as postprocessing and are not linked to a particular method used for stress analysis. Current methods applicable in the EFGM, as have been adopted in the FEM or BEM for the extraction of SIFs, are described and compared.

Fracture modelling using numerical methods is well-advanced in 2D using techniques such as the XFEM. The use of meshless methods for these problems lags somewhat behind but the potential benefits of no meshing (particularly in 3D) prompt continued research into their development. Meshless methods are believed to be better suited to solve problems with moving boundaries such as the modelling

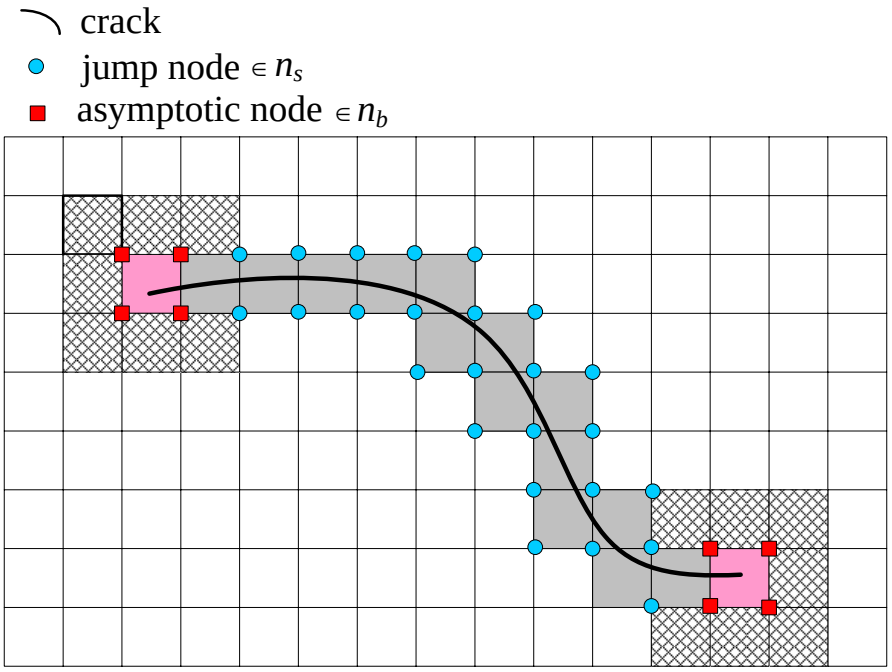
of crack propagation. The EFGM [3], is developed for crack modelling in a number of references [96, 99, 118, 120, 152, 153]. In the context of linear elastic fracture mechanics, the vicinity of a crack tip is dominated by discontinuous displacement field and singular stress field. To capture these features, special methods are needed in addition to the continuum formulation. In the basic FEM, a crack is explicitly represented by element edges or faces and the displacement jump caused by the crack can be naturally caught (see Figure 5.1(a)). In the XFEM, the displacement jump is introduced at the nodes according to the element intersection with the crack (see Figure 5.1(b)). In the EFGM, nodal supports intersecting with a crack need to be modified (as explained below, see Figure 5.1(c)) and special computational geometry techniques are required to introduce the displacement jump. For the stress singularity a solution is to enrich the approximation space so that it is better able to model a singularity. This idea has been developed both in the XFEM and the EFGM.

5.2 Extrinsic enrichment

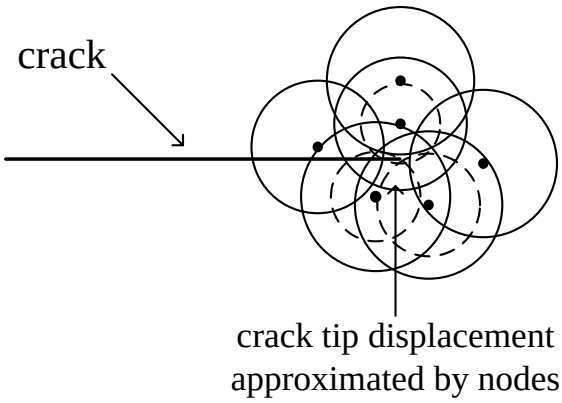
Extrinsic jump enrichment [154] in the EFGM adopts the approach used in the XFEM which is based on the partition of unity (PU) concept [32, 33] (refer to §2.2.3 for PU definition). The crack morphology is included by enrichment with *a priori* knowledge of the analytical solution for the displacement field around a crack tip in an infinite domain. Additional unknowns corresponding to enrichment terms are added to the global stiffness matrix. In the following, two methods are described, an asymptotic enrichment method and enriched trial functions, both of which add additional terms in approximation. They are reviewed here as comparison to methods using intrinsic enrichment but are not used the development of EFGM for fracture modelling in this thesis. In LFEM with the presence of singularity, it is assumed there is no plastic deformation ahead of crack tip. However, for the modelling cohesive cracks which can be found in materials such as concrete and bulk polymer, the following methods are believed not applicable [155].



(a) Crack geometry is explicitly meshed in the FEM



(b) Crack geometry is recognized by the nodes of which the elements are partially or completely cut by the crack in the XFEM



(c) Crack geometry is approximated using the MLS approximation in the EFGM

Figure 5.1: Comparison of crack discontinuity representation in the FEM, XFEM and EFGM.

5.2.1 Asymptotic enrichment

In the XFEM, the enriched terms are associated with nodes and the type of the enrichment is determined by the relation between the nodes and the crack. Nodes are divided into three groups, namely the nodes having their connecting elements completely cut by the crack, those having their connecting elements partially cut by the crack and those not interfering with the crack. In the EFGM a nodal support is equivalent to an element where a node exerts influence on the approximation over a point of interest. The nodes are classified into three groups in a similar way, namely the nodes having their support completely cut by the crack, those whose support contains a crack tip inside and those have no intersection with a crack. In Figure 5.2, four nodal supports of different intersection with a crack are shown. Nodes 1 and 2 are completely cut by the crack, the former above the crack and the latter beneath the crack, node 4 is partially cut by the crack, i.e. containing the crack tip, and node 3 has no intersection with the crack at all. For nodes 1 and 2, the displacement jump needs to be considered while the branch enrichment is introduced at node 4 to capture the singular stress field.

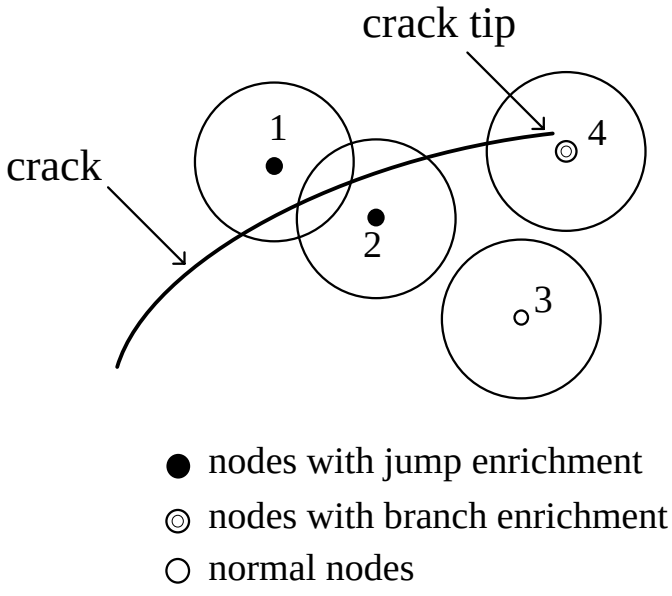


Figure 5.2: Nodal support completely cut, partially cut or non-intersecting a crack.

In the extrinsic enrichment in the EFGM, two additional terms are added to the

approximation for displacement u^h at a given point $\mathbf{x}$, which becomes

$$u^h(\mathbf{x}) = \sum_{I}^{n} N_I(\mathbf{x}) u_I + \sum_{J \in w_s}^{n_s} N_J(\mathbf{x}) H(\phi(\mathbf{x})) \alpha_J + \sum_{K \in w_b}^{n_b} N_K(\mathbf{x}) (\beta_{Kj} B_j(r,\theta)) \ , \qquad (5.1)$$

where the third term on the *r.h.s.* is the asymptotic enrichment, n is the total number of nodes in support of $\mathbf{x}$, N_I are the shape functions generated by the MLS approximation, I is the index of a supporting node, α_J and β_{Kj} are additional unknowns that will appear in the global stiffness equations, w_s is the set of nodes having its support completely cut by a crack and w_b collects all the nodes each having a crack tip inside its support. Here $H(\phi(\mathbf{x}))$ is the Heaviside step function

$$H(\phi(\mathbf{x})) = \begin{cases} 1 & \text{if above the crack surface} \\ -1 & \text{otherwise} \end{cases} \ , \qquad (5.2)$$

where $\phi(\mathbf{x})$ is the signed distance normal to the crack as shown in Figure 5.3. The asymptotic enrichment captures the near tip displacement field using the Westergaard expansion [143, 156] as

$$B(r,\theta) = \left\{ \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta \right\} \qquad (5.3)$$

where (r,θ) is the local coordinate system at the crack tip as shown in Figure 5.3. Extrinsic techniques have been developed into meshless crack "particle" methods for

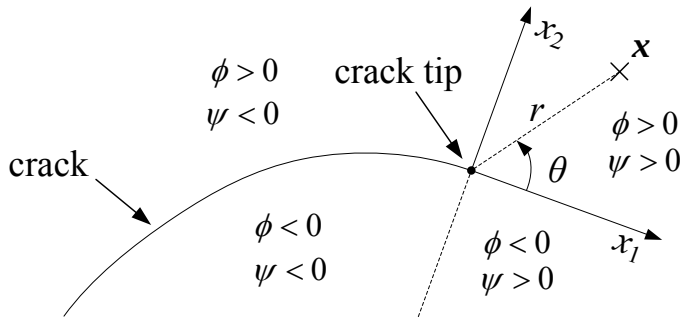


Figure 5.3: The local coordinate system at a crack tip and the signed distance to the crack surface.

3D crack modelling in a number of references [102, 103, 105, 113] and shifted weight functions are used as an alternative to the asymptotic basis in (5.3) for cohesive crack modelling. Extrinsic enrichment can lead to an ill-conditioned global stiffness

matrix [157] as is the case with many other PU methods, due to the additional unknowns at nodes which do not correspond to the physical degrees of freedom [126]. This problem can be anticipated to heighten for 3D problems where the proportion of additional unknowns would be higher. Béchet *et al.* [157] discusses this problem, proposing a special preconditioning which, however, adds to the computational expense.

5.2.2 Enriched trial function

Enriched trial functions used to capture the singular stress field are proposed in [143]. The method is similar to the previous techniques developed in the FEM by Benzley [158]. In the following, the formulation of calculating $u^h(\mathbf{x})$ is described for displacement component in x direction while those of deriving y and z directions are similar. Compared to the approximation in Equation (2.2), the trial function of the EFGM is augmented to include the asymptotic approximation for the displacement near the crack tip as

$$u^h(\mathbf{x}) = \underbrace{\mathbf{p}(\mathbf{x})^T\mathbf{a}(\mathbf{x})}_{\text{classic}} + \underbrace{\sum_j^{n_c}[k_1^j Q_1^j(\mathbf{x}) + k_2^j Q_2^j(\mathbf{x})]}_{\text{enriched}} \tag{5.4}$$

where the definition of the $\mathbf{p}(\mathbf{x})$ and other terms are the same as in §2.2.1, and $Q_1(\mathbf{x})$ and $Q_2(\mathbf{x})$ are the near-tip displacement field [79]. For displacement in horizontal direction (with respect to local coordinate system) Q_1 and Q_2 are given as

$$Q_1(\mathbf{x}) = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\cos\left(\frac{\theta}{2}\right)\left[\kappa - 1 + 2\sin^2\left(\frac{\theta}{2}\right)\right] \tag{5.5}$$

$$Q_2(\mathbf{x}) = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\left(\frac{\theta}{2}\right)\left[\kappa + 1 - 2\cos^2\left(\frac{\theta}{2}\right)\right] , \tag{5.6}$$

and for displacement in vertical direction Q_1 and Q_2 are given as

$$Q_1(\mathbf{x}) = \frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\sin\left(\frac{\theta}{2}\right)\left[\kappa + 1 + 2\cos^2\left(\frac{\theta}{2}\right)\right] \tag{5.7}$$

$$Q_2(\mathbf{x}) = -\frac{1}{2\mu}\sqrt{\frac{r}{2\pi}}\cos\left(\frac{\theta}{2}\right)\left[\kappa - 1 - 2\sin^2\left(\frac{\theta}{2}\right)\right] \tag{5.8}$$

where μ is the shear modulus and κ is the Kolosov constant defined as

$$\kappa = \begin{cases} 3 - 4\nu & \text{plane strain} \\ (3-\nu)/(1+\nu) & \text{plane stress} \end{cases} . \tag{5.9}$$

The unknown coefficients $\mathbf{a}(\mathbf{x})$ can be solved by minimizing a weighted norm, a procedure in the MLS approximation similar to Equation (2.3) as

$$J = \sum_{I}^{n} \frac{1}{2}\mathbf{w}(\mathbf{x}-\mathbf{x}_I)\left[\mathbf{p}^T(\mathbf{x})\mathbf{a} + \sum_{j}^{n_c}[k_1^j Q_1^j(\mathbf{x}_I) + k_2^j Q_2^j(\mathbf{x}_I)] - u_I\right]^2 \tag{5.10}$$

where n is the number of nodes in support of $\mathbf{x}$. The stationarity of J with respect to $\mathbf{a}_\alpha(\mathbf{x})$ leads to

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \sum_{I}^{n}\mathbf{P}(\mathbf{x})\left\{u_I - \sum_{j}^{n_c}[k_1^j Q_1^j(\mathbf{x}_I) + k_2^j Q_2^j(\mathbf{x}_I)]\right\} \tag{5.11}$$

where

$$\mathbf{A}(\mathbf{x}) = \sum_{I}^{n} w(\mathbf{x}-\mathbf{x}_I)\mathbf{p}(\mathbf{x}_I)\mathbf{p}^T(\mathbf{x}_I) \tag{5.12}$$

and

$$\mathbf{P}(\mathbf{x}) = [w(\mathbf{x}-\mathbf{x}_1)\mathbf{p}(\mathbf{x}_1), w(\mathbf{x}-\mathbf{x}_2)\mathbf{p}(\mathbf{x}_2), \cdots, w(\mathbf{x}-\mathbf{x}_n)\mathbf{p}(\mathbf{x}_n)]\ . \tag{5.13}$$

By solving Equation (5.11), we have

$$\mathbf{a}(\mathbf{x}) = \sum_{I}^{n}\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x})\left\{u_I - \sum_{j}^{n_c}[k_1^j Q_1^j(\mathbf{x}_I) + k_2^j Q_2^j(\mathbf{x}_I)]\right\}\ . \tag{5.14}$$

Then the displacement field is approximated by

$$\begin{aligned} u^h(\mathbf{x}) = &\sum_{I}^{n}\mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x})\left\{u_I - \sum_{j}^{n_c}[k_1^j Q_1^j(\mathbf{x}_I) + k_2^j Q_2^j(\mathbf{x}_I)]\right\} \\ &+ \sum_{j}^{n_c}[k_1^j Q_1^j(\mathbf{x}_I) + k_2^j Q_2^j(\mathbf{x}_I)]\ . \end{aligned} \tag{5.15}$$

It seen that the first term on the *r.h.s.* of the above equation (5.15) has the classic EFGM shape functions

$$N_I(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}^{-1}\mathbf{P}(\mathbf{x})\ . \tag{5.16}$$

Thus Equation (5.15) can be rewritten as

$$u^h(\mathbf{x}) = \sum_{I}^{n} N_I(\mathbf{x})\tilde{u}_I + \sum_{j}^{n_c}[k_1^j Q_1^j(\mathbf{x}_I) + k_2^j Q_2^j(\mathbf{x}_I)] \tag{5.17}$$

where

$$\tilde{u}_I = u_I - \sum_{j}^{n_c}[k_1^j Q_1^j(\mathbf{x}_I) + k_2^j Q_2^j(\mathbf{x}_I)]\ . \tag{5.18}$$

Determination of the derivatives of $u^h(\mathbf{x})$ requires only the shape functions $N_I(\mathbf{x})$ to be derived. The enriched trial terms k_1 and k_2 are unknown constants like u_I thus there is no need to differentiate them with respect to the coordinates. The formulation of the approximation is similar to enrichment functions developed for the FEM [158], however the difference is that in the FEM k_1 and k_2 correspond to K_{I} and K_{II} which is part of the solution [158], while in the EFGM a postprocessing step is still needed to extract the SIFs [159]. The difference lies in the way that shape functions are formed in the two methods.

5.3 Intrinsic enrichment

The MLS approximation used in the EFGM allows the number of bases to be independent of the number of degrees of freedom. The field can therefore be enriched in an intrinsic way without introducing extra unknowns, which is desirable in terms of computational cost. The first paper to use the intrinsic basis in the EFGM for fracture modelling was by Fleming et al. [159] and the term "intrinsic" was coined for the EFGM in a general discussion under the concept of PU by Fries and Belytschko [148]. Using an intrinsic method in the EFGM there is also no problem of ill-conditioning in the global stiffness matrix. However, the convergence rate is said to be slower than the EFGM using extrinsic enrichment [154]. A high order basis such as quadratic ($\{1, x, y, x^2, xy, y^2\}$) can be regarded as a kind of intrinsic enrichment since the number of degrees of freedom appearing in the basis function are higher than the physical degrees of freedom at nodes.

5.3.1 Jump enrichment

The displacement jump can be introduced simply by modifying the nodal support via the weight function. A simple way to do this is directly to truncate the nodal support at a crack face, which is a line or curve in 2D. This is known as the visibility criterion, and is explained in Figure 5.4. The support of a node is restricted to areas of the domain visible from the node with the crack faces acting as an opaque barrier. If a line between a node and the point of interest intersects a crack, and if the crack

tip is inside the support of that node, the node will have no influence on that point, i.e. r_I between that point and the node is modified to infinity. It corresponds to the use of the Heaviside function in the enriched trial functions used in XFEM described in §5.2.1. An alternative to the visibility criterion is the diffraction method which works slightly differently as shown in Figure 5.5. Instead of a sudden truncation, the support is allowed to bend (or "diffract") around the crack tip, and the distance is defined as the diffracted path between a point and the node. While conceptually simple, the diffraction method becomes computationally complex for the case of non-planar cracks in 3D, or for multiple cracks in 2D and 3D. These two methods can be summarized as follows

$$\text{Visibility}: r_I = \infty\,; \quad \text{Diffraction}: r_I = \left(\frac{r_1 + r_2(\mathbf{x})}{r_0(\mathbf{x})}\right)^{\lambda} r_0(\mathbf{x}) \tag{5.19}$$

where $r_1 = \|\mathbf{x}_I - \mathbf{x}_c\|$ is independent of $\mathbf{x}$, $r_2 = \|\mathbf{x} - \mathbf{x}_c\|$ and $r_0 = \|\mathbf{x} - \mathbf{x}_I\|$ as shown in Figure 5.5, $\mathbf{x}$ is the point of interest, $\mathbf{x}_I$ are the nodal coordinates, $\mathbf{x}_c$ are the crack tip coordinates and λ is a dilatancy parameter which controls the nodal influence at the point of interest (typically $1.0 \leq \lambda \leq 1.5$; here we use $\lambda = 1.2$). The shape functions and their derivatives using the visibility criterion and diffraction method are plotted in Figure 5.7 and 5.8 respectively using the nodal arrangement in Figure 2.8(a). It can be seen that the weight function based on diffraction method affects a larger area than the visibility criterion which bends around the crack tip. A similar feature can be found in the shape functions and their derivatives, however both become more complicated. To capture the singular stress field, the enriched basis function in (5.3) can be added to the normal basis [143, 148, 160]. Alternatives are to use improved weight functions [161] and enriched weight functions [108, 162], both of which introduce no additional unknowns.

Neither of the procedures to deal with the displacement jump are, however, entirely satisfactory as they do not directly correspond to the physical behaviour of cracks. In particular, the definition of the path of diffraction between two points is arbitrary in 3D when the crack front is curved, or when there are multiple cracks close to each other [5] as shown in Figure 5.9. In these situations, it is not possible to design a general formulation to calculate the derivatives of the modified distance without a simplification of the crack geometry. For example, in [108, 113] triangular

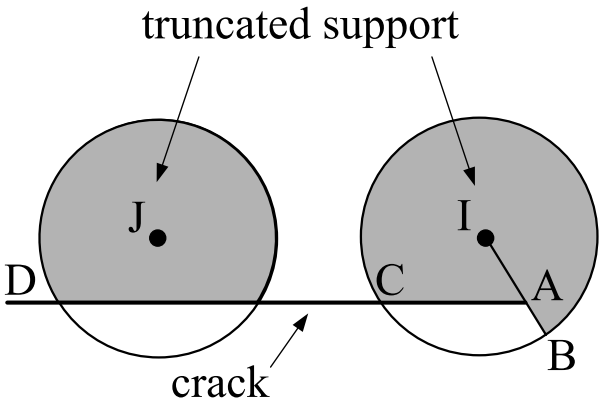


Figure 5.4: The visibility criterion.

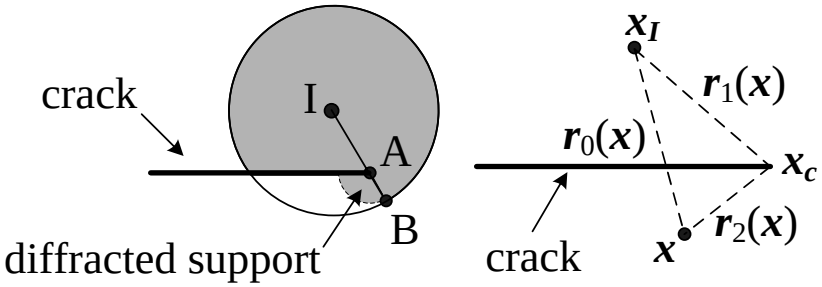
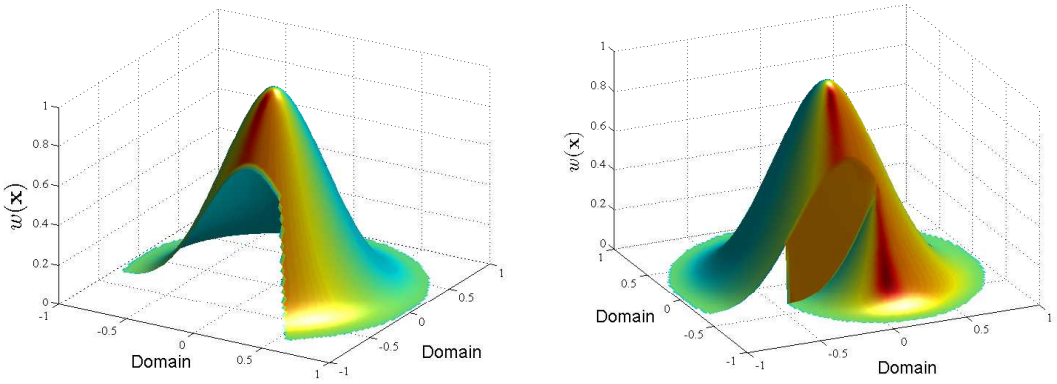
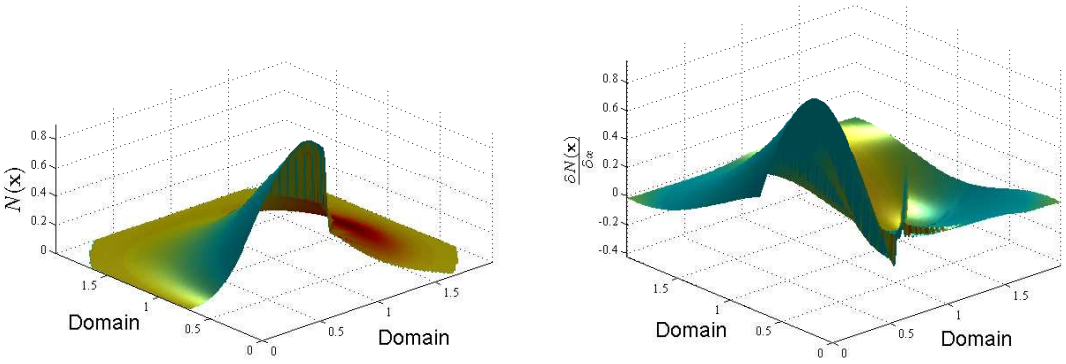


Figure 5.5: The diffraction method.



(a) Visibility criterion (b) Diffraction method

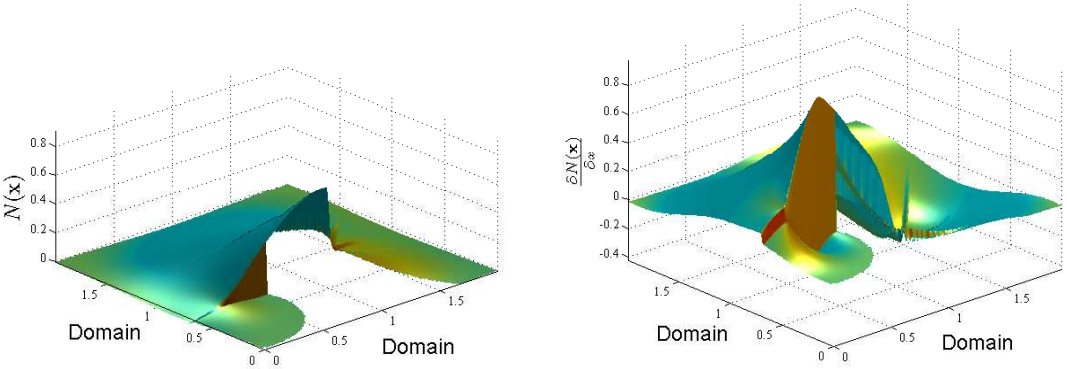
Figure 5.6: Weight functions changed by use of the visibility criterion and diffraction method over a 2D domain. (contour surfaces shown)



(a) Shape function

(b) First order derivative of the shape function

Figure 5.7: Plot of shape functions and its first order derivative over a 2D domain by the visibility criterion. (contour surfaces shown)



(a) Shape function

(b) First order derivative of the shape function

Figure 5.8: Plot of shape function and its first order derivative over a 2D domain by the diffraction method. (contour surfaces shown)

facets are used explicitly to represent the crack surface, and a polyline consisting of edges of these triangles is used to represent the crack front. However, the accuracy of the crack geometry is reduced by this method. For instance, a penny shaped crack, which has a smooth crack front, must be modelled as a collection of straight line segments which have C_0 continuity. This difficulty can be easily dealt with by the use of the level set method (LSM) where front curvature can be accurately captured as naturally smooth and continuous as will be demonstrated later.

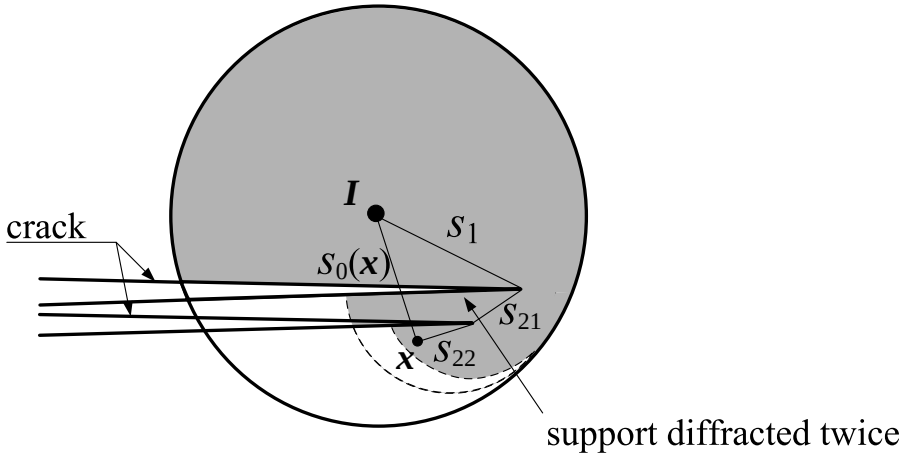


Figure 5.9: The diffraction path for multiple cracks (reproduced from [5]).

5.3.2 Asymptotic enrichment

In intrinsic enrichment methods, asymptotic bases for capturing singular stress are added to polynomial basis functions for describing normal displacements (without a crack). Therefore the basis function $\mathbf{p}^a(\mathbf{x})$ becomes

$$\mathbf{p}^a(\mathbf{x}) = \left\{1, x, y, \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta\right\} \tag{5.20}$$

where the superscript a indicates "augmented basis" comprising seven terms where the first three are from the linear basis. If a quadratic basis is used, there will be ten bases, six from quadratic and four from the asymptotic basis. The rest of the implementation to calculate the shape functions and derivatives will be the same as in the standard EFGM. An asymptotic basis in Equation 5.3 is termed as "full" enrichment since it forms the complete independent basis from Westergaard

expansion for describing the displacement [82]. A reduced version is the radial enrichment

$$\mathbf{p}^a(\mathbf{x}) = \left\{1, x, y, \sqrt{r}\right\} \ , \tag{5.21}$$

which is simpler however not as accurate [143]. The use of the intrinsic enrichment here does not involve additional unknowns, which is an advantage in the EFGM compared of the FEM or the XFEM. In the latter the order of approximation is associated with the element topology. However the requirement of nodal arrangement is more strict in the former concerning the condition number of the $\mathbf{A}$ matrix (refer to Equation (2.8)) in solving the shape functions [163, 164].

5.4 Fracture analysis

SIFs are the most important parameters for any fracture analysis as they are used to predict crack growth rate and direction. The direct calculation of SIFs based on the stress and displacement results is not accurate for general cases due to the varying crack geometries and complicated stress fields around the crack tip. Alternatively, various integrals which only require far field results from the crack tip have been proposed, namely the J integral of strain energy density [165], the M_1 integral based on conservation laws [166], a contour integral based on Betti's reciprocal theorem [167] and a domain integral in 2D or 3D [168–171]. They vary from each other in formulations but are based on an identical measurement of energy released rate around the crack tip. In this section, we describe some of the most useful methods which are performed as postprocessing steps which are independent of the method used for field approximation.

5.4.1 The J integral as energy release rate

The J integral was introduced by Rice [165] and the name was coined after the author's given name. In [165], a path independent integration around crack tip was proposed as a mean of determining SIFs. The integral provides a generalisation of the relation between the SIFs and energy release rate. In the following we refer to [6]

to derive the J integral expression from the point of view of total potential energy. Of course one may arrive at the same expression in some other ways such as the use of the Eshelby momentum tensor [172]. We consider an infinite elastic plate of unit thickness in 2D containing a crack as shown in Figure 5.10(a). A control volume A is taken around the crack tip bounded by Γ. Suppose the volume has an arbitrary surface traction $\mathbf{t}$ applied along Γ and if we assume the absence of body forces, then the total potential energy Π over the area A of this plate is given by

$$\Pi = \int_A W \, \mathrm{d}A - \int_\Gamma \mathbf{t} \cdot \mathbf{u} \, \mathrm{d}\Gamma \tag{5.22}$$

where $W = \boldsymbol{\sigma} : \boldsymbol{\varepsilon}$ is the strain energy density, $\mathbf{t}$ and $\mathbf{u}$ are vectors of tractions and displacement along Γ respectively. Suppose the crack is extends by length Δa as shown in Figure 5.10(b) then the total potential energy will change. Denoting the potential energy before crack extension as Π_1 and after extension as Π_2, then the potential energy release rate can be calculated as

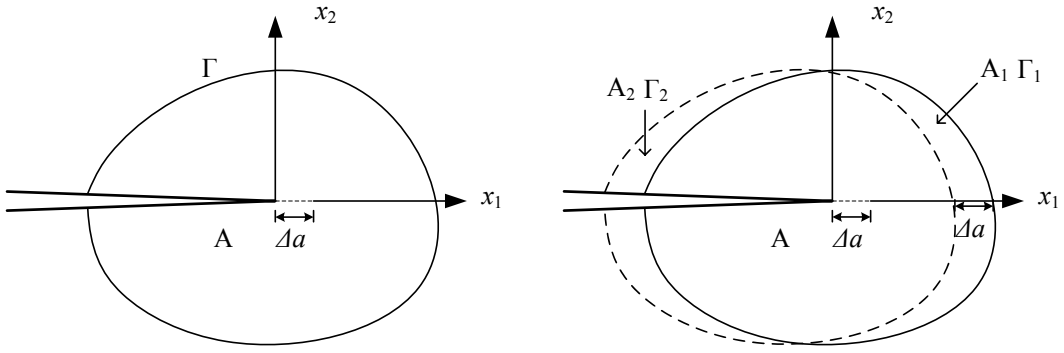
$$-\frac{\partial \Pi}{\partial a} = -\lim_{\Delta a \to 0} \left\{ \frac{\Pi_2 - \Pi_1}{\Delta a} \right\} . \tag{5.23}$$

The domains used to calculate Π_1 and Π_2 are the areas bounded by a closed curve and a closed dashed curve respectively as shown in Figure 5.10(b), and are marked with A_1 and A_2 respectively. Domain A_1 is the same as A in Figure 5.10(a) while A_2 is obtained by keeping the crack length fixed and shifting the original crack toward the left by the same amount of crack extension. For an infinite problem, this is equivalent to extending the original crack in the direction of Δa. From Equation (5.22), the potential energies Π_1 and Π_2 can be expressed using the strain energy density hence

$$-\frac{\partial \Pi}{\partial a} = -\lim_{\Delta a \to 0} \frac{1}{\Delta a} \left\{ \int_{A_1 - A_2} W \, \mathrm{d}A - \int_{\Gamma_1} \mathbf{t} \cdot \mathbf{u} \, \mathrm{d}\Gamma + \int_{\Gamma_2} \mathbf{t} \cdot \mathbf{u} \, \mathrm{d}\Gamma \right\} \tag{5.24}$$

In Equation (5.24), Δa is approaching zero thus it can be assumed that the displacement field between Γ_1 and Γ_2 varies linearly. Thus the stress and strain fields which are determined from the derivatives of the displacement field will be constant. Therefore in Equation (5.24) the second and third term on the *r.h.s.* containing $\mathbf{t}$ can be combined together and the equation becomes

$$-\frac{\partial \Pi}{\partial a} = -\lim_{\Delta a \to 0} \frac{1}{\Delta a} \left\{ \int_{A_1 - A_2} W \, \mathrm{d}A - \int_{\Gamma} \mathbf{t} \cdot (\mathbf{u}^1 - \mathbf{u}^2) \, \mathrm{d}\Gamma \right\} \tag{5.25}$$



(a) A crack in a plate and crack extension toward the right

(b) The area of calculating potential energy moving toward the left

Figure 5.10: A 2D plate containing a crack and crack extension used for calculating potential energy.

where $\mathbf{u}^1$ and $\mathbf{u}^2$ denotes the displacements along Γ_1 and Γ_2 respectively. From Figure 5.11, it can be seen that $\mathbf{u}^1$ and $\mathbf{u}^2$ are related by

$$\mathbf{u}^1 = \mathbf{u}^2 + \frac{\partial \mathbf{u}}{\partial x_1}\Delta a \tag{5.26}$$

Substituting this relation into (5.24) gives

$$-\frac{\partial \Pi}{\partial a} = -\lim_{\Delta a \to 0}\frac{1}{\Delta a}\left\{\int_{A_1 - A_2} W \,\mathrm{d}A - \int_{\Gamma} \mathbf{t}\cdot\frac{\partial \mathbf{u}}{\partial x_1}\Delta a \,\mathrm{d}\Gamma\right\} \tag{5.27}$$

It can be seen from Figure 5.11 that the area difference between the two boundaries is given by $\mathrm{d}A = \Delta a \cdot \mathrm{d}x_2$ and substitution into Equation (5.27) leads to

$$-\frac{\partial \Pi}{\partial a} = -\lim_{\Delta a \to 0}\frac{1}{\Delta a}\left\{\int_{\Gamma} W\,\Delta a \mathrm{d}x_2 - \int_{\Gamma} \mathbf{t}\cdot\frac{\partial \mathbf{u}}{\partial x_1}\Delta a \,\mathrm{d}\Gamma\right\} \ . \tag{5.28}$$

Note that $\mathrm{d}x_2 = n_1 \mathrm{d}\Gamma$ with n_1 as the component of outer normal along Γ in x_1 direction as shown in Figure 5.11. Then Equation (5.28) gives the J integral as an expression of energy release rate

$$J = \int_{\Gamma}\left(W n_1 - \mathbf{t}\cdot\frac{\partial \mathbf{u}}{\partial x_1}\right)\mathrm{d}\Gamma \ . \tag{5.29}$$

It has been shown here that the J integral therefore provides a fracture parameter for measuring the potential energy release rate near crack tip as an alternative to the Griffith theory.

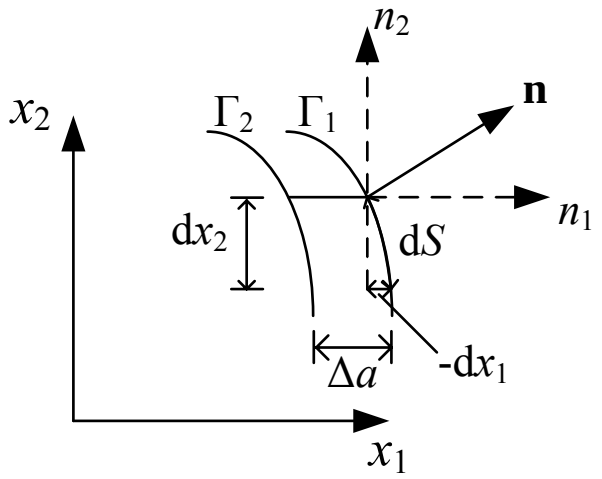
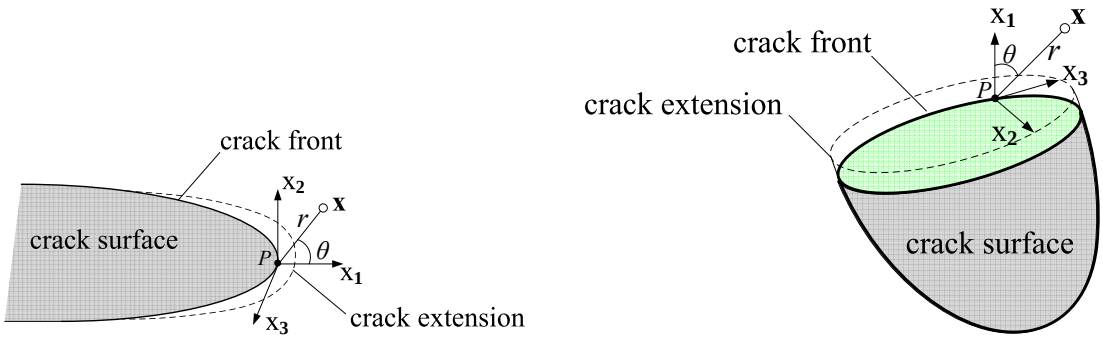


Figure 5.11: Two boundaries separated by a horizontal distance Δa (reproduced from [6]).



(a) planar crack surface in 3D

(b) non-planar crack surface in 3D

Figure 5.12: The local coordinate system defined along the crack front for crack surfaces in 3D

5.4.2 Extraction of stress intensity factor

In the previous section, we derived the J integral as an expression of potential energy release rate. It can be regarded as a special case of the more general expression

$$J_k = \int_{\Gamma_\varepsilon} \left(W n_k - \sigma_{ij} \frac{\partial u_i}{\partial x_k} n_j \right) \mathrm{d}\Gamma \tag{5.30}$$

where Γ_ε as a contour in a counter-clockwise direction with vanishing radius ε. When $k = 1$ Equation (5.30) becomes the J integral (also called the J_1 integral in many papers) and the contour is path independent for an arbitrary contour Γ close to the crack tip that

$$J_1 = \int_{\Gamma} \left(W n_1 - \sigma_{ij} \frac{\partial u_i}{\partial x_1} n_j \right) \mathrm{d}\Gamma \ , \tag{5.31}$$

which is essentially the same as Equation (5.29). In 3D a crack tip becomes a (curved) line (a crack front) and the local coordinates can vary as a point moves along the crack front, as shown in Figure 5.12. Then an integral calculation near a point P lying on the front will be influenced by other points on the front close to P as shown in Figure 5.12. In this case, the contribution of stress and strain in x_3 direction cannot be neglected and path independence of the J_k integral in 3D holds true only when ε tends to zero. For an arbitrary contour Γ which is not approaching zero, an area term comprising stress and strain terms with respect to x_3 needs to be added [173] so Equation (5.30) becomes

$$J_k = \int_{\Gamma} \left(W n_k - \sigma_{ij} \frac{\partial u_i}{\partial x_k} n_j \right) \mathrm{d}\Gamma - \int_{\Omega} \frac{\partial}{\partial x_3} \left(\sigma_{i3} \frac{\partial u_i}{\partial x_k} \right) \mathrm{d}\Omega \tag{5.32}$$

where Ω is the area bounded by Γ. Equation (5.32) is path-area independent integration in 3D and different from the path independent integration in 2D. Cherepanov [174] showed that for each of the three modes (refer to §1.3.4) there is a corresponding J integral, namely J^{I}, J^{II} and J^{III} and they are related to J_k as follows

$$J_1 = J^{\mathrm{I}} + J^{\mathrm{II}} + J^{\mathrm{III}} \tag{5.33}$$

$$J_2 = -2\sqrt{J^{\mathrm{I}} J^{\mathrm{II}}} \tag{5.34}$$

where J^{III} is often noted as in G_{III} [168] and J^{M} for each mode are defined as

$$J^{\mathrm{M}} = \int_{\Gamma_\varepsilon} \left(W^M n_1 - \sigma_{ij}^M \frac{\partial u_i^M}{\partial x_1} n_j \right) \mathrm{d}\Gamma, \quad M = \mathrm{I}, \mathrm{II}, \mathrm{III} \ . \tag{5.35}$$

If we substitute the analytical solutions of stresses and displacements near a crack tip (refer to §C.1) into Equation (5.35), the relation between SIFs and J_1 and J_2 can be found as

$$J_1 = \frac{1}{E^*}(K_{\mathrm{I}}^2 + K_{\mathrm{II}}^2) + \frac{1}{2G}K_{\mathrm{III}}^2 \tag{5.36}$$

$$J_2 = -\frac{2}{E^*}K_{\mathrm{I}}K_{\mathrm{II}} \ . \tag{5.37}$$

While K_{I} and K_{II} are both linked to J_1 and J_2, there is an independent relation between energy release rate due to a mode III crack denoted as G_{III} and K_{III} as

$$G_{\mathrm{III}} = \frac{K_{\mathrm{III}}^2}{2\mu} \tag{5.38}$$

where μ is the shear modulus of the material

$$\mu = \frac{E}{2(1-\nu^2)} \tag{5.39}$$

often noted as G in the literature, E^* is a modified Young's modulus determined by E and ν as

$$E^* = \begin{cases} \dfrac{E}{1-v^2} & \text{plane strain} \\ E & \text{plane stress} \end{cases} \tag{5.40}$$

For the general 3D case Cherepanov [175] defines E^* as

$$E^* = \frac{E}{1-\nu^2} + \frac{\nu}{1+\nu}\frac{\varepsilon_{33}}{\varepsilon_{11}+\varepsilon_{22}} \tag{5.41}$$

where all the strain terms of ε are defined locally in the vicinity of the crack front. For a plane strain problem, where $\varepsilon_{33} = 0$, and a plane stress problem, where $\sigma_{33} = 0$, Equation (5.41) simply becomes (5.40), and for any intermediate state it provides an interpolation. From Equations (5.33) to (5.38), the SIFs can be found as

$$K_{\mathrm{I}} = \frac{1}{2}\sqrt{E^*}\left(\sqrt{J_1 - J_2 - G_{\mathrm{III}}} + \sqrt{J_1 - J_2 - G_{\mathrm{III}}}\right) \tag{5.42}$$

$$K_{\mathrm{II}} = \frac{1}{2}\sqrt{E^*}\left(\sqrt{J_1 - J_2 - G_{\mathrm{III}}} - \sqrt{J_1 - J_2 - G_{\mathrm{III}}}\right) \tag{5.43}$$

$$K_{\mathrm{III}} = \sqrt{2\mu G_{\mathrm{III}}} \tag{5.44}$$

However, the J_2 integral includes singular terms as n_2 is non zero when approaching the tip. For a mixed mode crack problem, currently the most popular method

used in fracture mechanics is the decomposition of the mixed mode J_1 integral originally proposed in 2D by Kitagawa et al. [176] and later used in the dual BEM by Portela et al. [177]. These formulations in 3D fracture mechanics have been discussed by Rigby and Aliabadi [173] and later some errors in stress decomposition were corrected by Huber et al. [7]. In the test examples presented in this thesis, we use this method to calculate SIFs. The decomposition method takes advantage of the symmetric and antisymmetric mechanical behavior in the three crack modes to extract the stress and strain which are uniquely influenced by a certain mode

$$\sigma_{ij} = \sigma_{ij}^{\mathrm{S}} + \sigma_{ij}^{\mathrm{AS}} \tag{5.45}$$

where the superscript S denotes symmetric part and AS denotes antisymmetric part and σ_{ij} is the Cauchy stress at a point defined in the local coordinates system of crack as shown in Figure 5.13. Since the symmetric stress components are due to mode I crack while the antisymmetric components are from modes II and III, Equation (5.45) can be written as

$$\sigma_{ij} = \underbrace{\sigma_{ij}^{\mathrm{I}}}_{\sigma_{ij}^{\mathrm{S}}} + \underbrace{\sigma_{ij}^{\mathrm{II}} + \sigma_{ij}^{\mathrm{III}}}_{\sigma_{ij}^{\mathrm{AS}}} \tag{5.46}$$

where σ_{ij}^{I}, $\sigma_{ij}^{\mathrm{II}}$ and $\sigma_{ij}^{\mathrm{III}}$ are stresses caused by mode I, II and III cracks respectively. To extract the independent contribution from each mode, we can use the relations in stress states at a pair of points, e.g. P and P' symmetric to crack surfaces as shown in Figure 5.13. Figure 5.13(a) shows the stress state, which is symmetric, under mode I and Figure 5.13(b) shows the stress state, which is antisymmetric, from both mode II and III. It can seen that the mode I stress can be directly obtained as

$$\begin{Bmatrix} \sigma_{11}^{\mathrm{I}} \\ \sigma_{22}^{\mathrm{I}} \\ \sigma_{33}^{\mathrm{I}} \\ \sigma_{12}^{\mathrm{I}} \\ \sigma_{23}^{\mathrm{I}} \\ \sigma_{13}^{\mathrm{I}} \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} \sigma_{11} + \sigma'_{11} \\ \sigma_{22} + \sigma'_{22} \\ \sigma_{33} + \sigma'_{33} \\ \sigma_{12} - \sigma'_{12} \\ \sigma_{23} - \sigma'_{23} \\ \sigma_{13} - \sigma'_{13} \end{Bmatrix} \tag{5.47}$$

Since the stress components in mode II and III do not cross over (refer to the analytical solution for the near tip stress field Appendix C.1), i.e. $\sigma_{ij}^{\mathrm{II}}\sigma_{ij}^{\mathrm{III}} = 0$, they

can be obtained directly using the stress relation shown in Figure 5.13(b) as

$$\begin{Bmatrix} \sigma_{11}^{\mathrm{II}} \\ \sigma_{22}^{\mathrm{II}} \\ \sigma_{33}^{\mathrm{II}} \\ \sigma_{12}^{\mathrm{II}} \\ \sigma_{23}^{\mathrm{II}} \\ \sigma_{13}^{\mathrm{II}} \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} \sigma_{11} - \sigma'_{11} \\ \sigma_{22} - \sigma'_{22} \\ 0 \\ \sigma_{12} + \sigma'_{12} \\ 0 \\ 0 \end{Bmatrix} \tag{5.48}$$

and

$$\begin{Bmatrix} \sigma_{11}^{\mathrm{III}} \\ \sigma_{22}^{\mathrm{III}} \\ \sigma_{33}^{\mathrm{III}} \\ \sigma_{12}^{\mathrm{III}} \\ \sigma_{23}^{\mathrm{III}} \\ \sigma_{13}^{\mathrm{III}} \end{Bmatrix} = \frac{1}{2} \begin{Bmatrix} 0 \\ 0 \\ \sigma_{33} - \sigma'_{33} \\ 0 \\ \sigma_{23} + \sigma'_{23} \\ \sigma_{13} - \sigma'_{13} \end{Bmatrix} \tag{5.49}$$

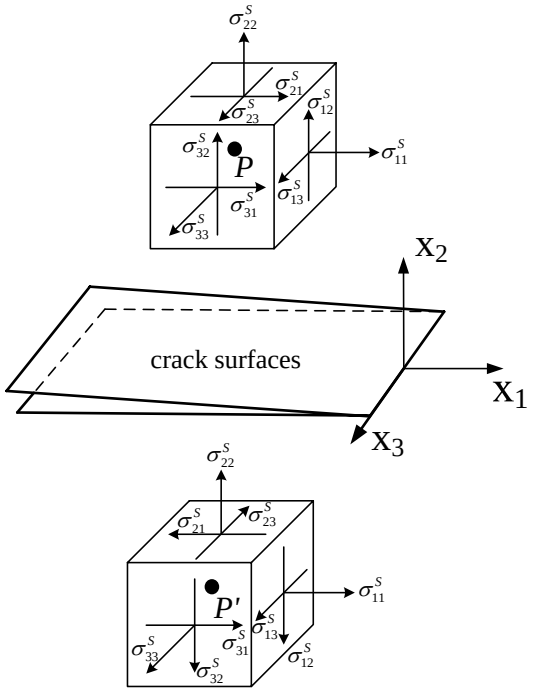
respectively. Similarly, strain components can also be decomposed into independent contributions from three crack modes [7]. With the stress and strain in each mode obtained, the J^M integral corresponding to each mode can be performed using Equation (5.35). From the relation between J^{M} and K_{M} in Equation (5.36) and (5.33), K_{I} and K_{II} can be found as

$$K_{\mathrm{I}} = \sqrt{J^{\mathrm{I}} E^*}, \quad K_{\mathrm{II}} = \sqrt{J^{\mathrm{II}} E^*}\ , \tag{5.50}$$

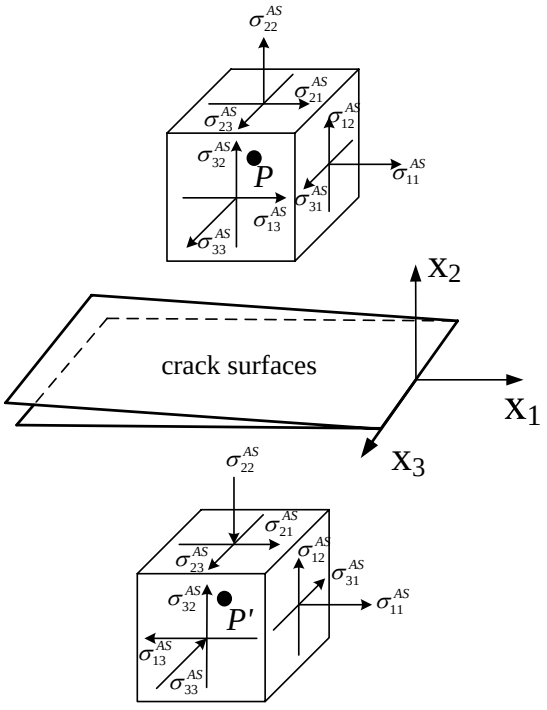
and K_{III} is found by Equation (5.38).

5.4.3 Contour integral method

In previous sections, we reviewed the J integral for the calculation of SIFs. For mixed mode problems, a decomposition of displacement and stress fields is needed. In this section, we discuss an alternative method, the contour integral method, proposed by Huber et al. [7] which can be used to calculate the mixed mode SIFs independently and does not involve the decomposition process. The method follows Betti's reciprocal theorem for static elastic fracture problems in 2D. It applies auxiliary stress and displacement fields over the real stress and displacement fields to



(a) Stress components symmetric to crack surface



(b) Stress components unsymmetric to crack surface

Figure 5.13: Stress components being decomposed to symmetric and antisymmetric parts at a pair of points P and P' symmetric to crack surfaces in 3D (referred from [7]).

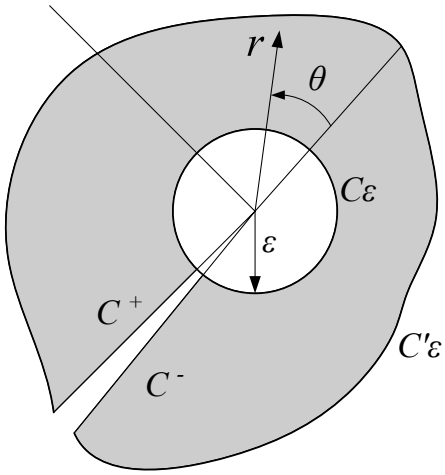


Figure 5.14: The crack tip polar coordinate system and contours used to calculate the work.

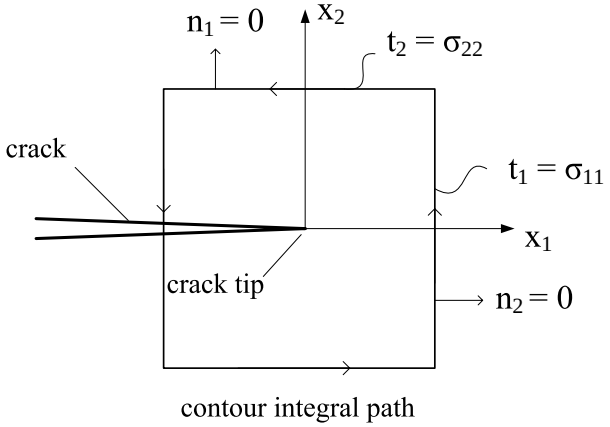


Figure 5.15: The contour integral in Cartesian (local) coordinates which has a simple implementation.

determine the SIFs. To apply the reciprocal theorem, two integral paths, namely C_ε and C'_ε are considered as shown in Figure 5.14. The stress and displacement fields all refer to the local polar coordinate system around the crack tip as shown in the figure. Taking a small piece of the domain around the crack tip, which is shown as the shaded area in Figure 5.14, the reciprocal theorem gives the work done along the boundary of the domain as

$$\int_C (\mathbf{u} \cdot \hat{\mathbf{t}} - \hat{\mathbf{u}} \cdot \mathbf{t}) \mathrm{d}s = 0 \tag{5.51}$$

where C is a path comprising C^+, C^-, C_ε and C'_ε as shown in the figure, $\hat{u}$ and $\hat{t}$ are

the displacement and tractions along C corresponding to another state (auxiliary fields) and $\mathbf{u}$ and $\mathbf{t}$ are solutions of the real problem. The crack surfaces are assumed traction free (same as in the J integral), so Equation (5.51) becomes

$$I_\varepsilon = -\int_{C_\varepsilon} (\mathbf{u}\cdot\hat{\mathbf{t}} - \hat{\mathbf{u}}\cdot\mathbf{t})\mathrm{d}s = \int_{C'_\varepsilon} (\mathbf{u}\cdot\hat{\mathbf{t}} - \hat{\mathbf{u}}\cdot\mathbf{t})\mathrm{d}s \ . \tag{5.52}$$

where I_ε is termed the interaction integral around the crack tip. This equation shows the path independence of the interaction work along an arbitrary path around and close to a crack tip. Huber et al. [7] derived suitable auxiliary fields using the complex functions (see Appendix §C.2). Substituting these into Equation (5.52) results in a very neat expression with some constants and SIFs as ε approach zero

$$I_\varepsilon = c_1 K_\mathrm{I} + c_2 K_\mathrm{II} \tag{5.53}$$

where c_1 and c_2 are constants from complex functional analysis. Therefore SIFs for each crack mode can be obtained respectively by comparing corresponding terms as

$$c_1 K_\mathrm{I} + c_2 K_\mathrm{II} = \int_{C'_\varepsilon} (\mathbf{u}\cdot\hat{\mathbf{t}} - \hat{\mathbf{u}}\cdot\mathbf{t})\mathrm{d}s \ . \tag{5.54}$$

Compared with the J integral decomposition in previous sections, the features of the contour integral approach are as follows.

- Freedom from the need to decompose the symmetric and antisymmetric terms of displacement and stress.
- K_I and K_II can be calculated independently and directly.
- Only displacement and tractions along the integral path are needed.

Some classic 2D crack examples were tested in Huber et al. [7] showing the robustness and accuracy based on this approach. The above formulation is based on polar coordinates and was later reformulated in Cartesian coordinates by Wen and Aliabadi [178]. The implementation can be simplified if a rectangle integral path is used so that $n_2 = 0$ and $\mathbf{t} = \sigma_{11}$ along the right-side edge as shown in Figure 5.14. The idea of using the auxiliary fields is extended to 3D in [169] which will be dealt with in the following section.

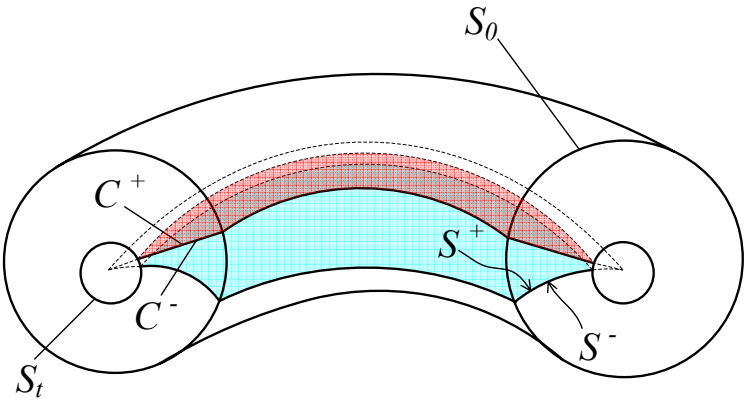


Figure 5.16: The crack tip polar coordinate system and contours used to calculate the work.

5.4.4 Domain integral method

By applying the divergence theorem, the J integral can be converted to a domain form, i.e. over an area in 2D or a volume in 3D rather then a closed curved in 2D or closed surface in 3D as is detailed in [168]. A domain integral approach [170, 179] provides a general form of integral near the crack front applicable to both linear and non-linear problems. With properly defined local curvilinear coordinates, the method is effective for non-planar cracks. Moran and Shih [170] adopted the idea of applying auxiliary field over the problem field [167] and developed the 3D interaction domain integral so that the SIF of each mode can be obtained independently [169]. In Appendix A.2, the interaction domain integral method is described in detail. Consider a part of a crack surface and a tubular domain denoted as Ω going along the crack front as shown in Figure 5.16. The interaction integral $\bar{I}$ for calculating the energy release over the tubular domain is given as

$$\bar{I} = \int_{\Omega} (P_{lj,j} q_l + P_{lj} q_{l,j}) \mathrm{d}\Omega + \int_{C^+ + C^-} P_{lj} q_l n_j \mathrm{d}S + \int_{S^+ + S^-} P_{lj} q_l n_j \mathrm{d}S \ , \tag{5.55}$$

where Ω is the volume bounded by S_0 and the crack surface S and a plane C which is an auxiliary surface tangent to S, n_j is the unit outer normal of a surface, S^+ and S^- are the upper and lower surfaces of S respectively, C^+ and C^- are the upper and lower surfaces of C respectively, and q_l is a test function given by

$$q_l = \begin{cases} \Delta a c_l(s) & \text{on} \quad S_t \\ 0 & \text{on} \quad S_0 + C^+ + C^- \end{cases} \ , \tag{5.56}$$

and P_{lj} is the interaction form of the Eshelby momentum tensor [172] given by

$$P_{lj} = \left(\sigma_{ik}\varepsilon_{ik}^{\text{aux}}\delta_{ij} - u_{i,l}^{\text{aux}}\sigma_{ij} - u_{i,l}\sigma_{ij}^{\text{aux}}\right) \ . \tag{5.57}$$

Since $\bar{I}$ is obtained through an integration over the domain of a small length along the crack front as shown in Figure 5.16, the average energy release rate $I(s)$ is calculated by

$$I(s) = \frac{\bar{I}}{\int_{L_c} \Delta a(s)\mathrm{d}s} \tag{5.58}$$

where L_c is the length of the tubular domain. The SIFs are found through the relation

$$I(s) = \frac{2(1-v^2)}{E}(K_{\text{I}}K_{\text{I}}^{\text{aux}} + K_{\text{II}}K_{\text{II}}^{\text{aux}}) + \frac{1}{\mu}K_{\text{III}}K_{\text{III}}^{\text{aux}} \tag{5.59}$$

where $K_{\text{I}}^{\text{aux}}$, $K_{\text{II}}^{\text{aux}}$ and $K_{\text{III}}^{\text{aux}}$ are the stress intensity factors associated with the auxiliary fields. When we evaluate SIFs for each mode, for example K_{I}, we can let the $K_{\text{I}}^{\text{aux}} = 1$, $K_{\text{II}}^{\text{aux}} = 0$ and $K_{\text{III}}^{\text{aux}} = 0$, thus K_{I} can be found from Equation (5.59) as

$$K_{\text{I}} = \frac{2(1-v^2)}{E}I(s) \ . \tag{5.60}$$

5.4.5 The J integral for non-planar crack

For non-planar crack surfaces, the assumption of symmetric and antisymmetric fields along the crack surface no longer holds true. Therefore, the decomposition of the J integral into different modes is not valid. If it is used, it can introduce considerable error, as shown by the examples in [180] of about 1% for K_{I} and 5% for K_{II} using fine FE meshes. To address this problem, Chang and Wu [180] proposed a curve fitting procedure for the J integral so it can be used for non-planar cracks with curvature on the crack surfaces. J_k is calculated by the following equation

$$\begin{aligned} J_k = &\int_{\Gamma}\left[Wn_k - \sigma_{ij}n_j\left(\frac{\partial u_i}{\partial x_k}\right)\right]\mathrm{d}\Gamma + \int_{\Omega}\left[Wn_k - \sigma_{ij}n_j\left(\frac{\partial u_i}{\partial x_k}\right)\right]\mathrm{d}\Omega \\ &+ \underbrace{\int_{(S^+-R)+(S^--R)} Wn_k\mathrm{d}S + \Lambda n_k R^{1/2}}_{\text{non-planar contribution}}, \end{aligned} \tag{5.61}$$

where S^+ and S^- are the upper and lower crack surfaces and R is a small distance to the crack front and Λ is a constant coefficient to be determined. The crack surface is divided into two parts, a part distant from the crack front namely $S^+ - R$ and $S^- - R$ and the residual part which is close to the crack front. In Equation (5.61) the first term on the *r.h.s* is the same as in Equation (5.30). Terms along crack surfaces are added since the traction cannot be assumed zero on a non-planar crack surface. The last term on the *r.h.s* shows the contribution as the surface integration approach the crack front. As R vanishes, n_1 approaches zero. So when calculating J_1, the last term on the *r.h.s* can be committed if R is small enough. For J_2 integral, n_2 approaches 1 as R vanishes. Then the contribution from the last term on the *r.h.s* needs to be considered which is of $O(\sqrt{r})$. In this case, the choice of R and Λ are determined by a curve fitting procedure by testing different values of R and Λ [180]. The above described method requires a number of trials to fit the $\sqrt{r}$ curve for every problem. To the author's knowledge, the domain integral method is the only method available at the moment which provides an exact and general formulation for arbitrary crack surfaces considering the curvature along the crack surface. However the computational cost of the domain integral method is very expensive in 3D and requires a lots of integrations points.

5.5 Crack propagation criterion

Once the SIFs are obtained, they can be used to predict crack propagation, i.e. the rate and direction of crack growth. Various rules have been developed depending on the stress state indicated by the SIFs, material properties and fracture type, e.g. brittle, fatigue, cohesive or corrosive crack.

5.5.1 Crack growth angle

The direction of crack growth with respect to the current crack front geometry is determined by the ratio of SIFs between different modes. For a pure mode I crack, the crack front is only driven by the opening force lying in the plane orthogonal to the crack front. The crack growth in this case does not change direction with

respect to the original crack front and the increment of the crack surface shown as the dashed area in Figure 5.17(a) stays in the plane tangential to x_1, the outer normal to the original crack front. For a crack subject to mixed modes, the crack growth will change direction with respect to the original crack surface due to the existence of the sliding force of mode II, as shown in Figure 5.17(b). In this case, the crack opens and slides at the same time and the increment of the crack surface is inclined to x_1 as shown in the figure. The crack growth angle θ_p at a point on a crack front is only determined by K_{I} and K_{II}. There are many criteria to determine θ_p, such as the maximum principal stress and maximum strain energy density. Here we use the maximum principal stress. According to this criterion, the crack growth direction is perpendicular to the maximum tangential (hoop) stress $\sigma_{\theta,\max}$. The crack bending angle can be obtained by

$$\theta_p = 2 \arctan \frac{1}{4} \left(\frac{K_{\mathrm{I}}}{K_{\mathrm{II}}} - \mathrm{sign}(K_{\mathrm{II}}) \sqrt{\left(\frac{K_{\mathrm{I}}}{K_{\mathrm{II}}} \right)^2 + 8} \right) \tag{5.62}$$

where θ_p is measured with respect to x_1 in the local coordinates at a point as shown in Figure 5.17, K_{I} and K_{II} are the mode I and mode II SIFs, respectively. The general dominance of mode I crack through kink or tilt of mixed mode cracks are observed in repeatable results from laboratory experiments [181] the main conclusion of which is that the mixed mode crack will eventually become a mode I crack.

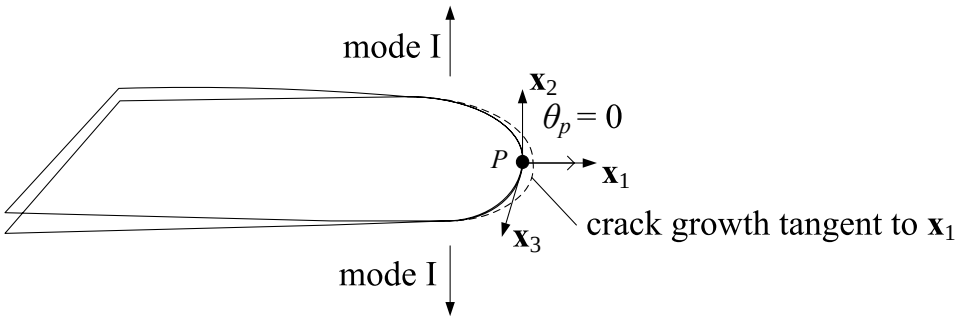
5.5.2 Fatigue crack growth

Fatigue crack growth refers to the slow process of crack propagation under cyclic loading. The Paris crack growth law [182] is commonly used which gives the rate of crack growth in mode I in terms of load cycles N as

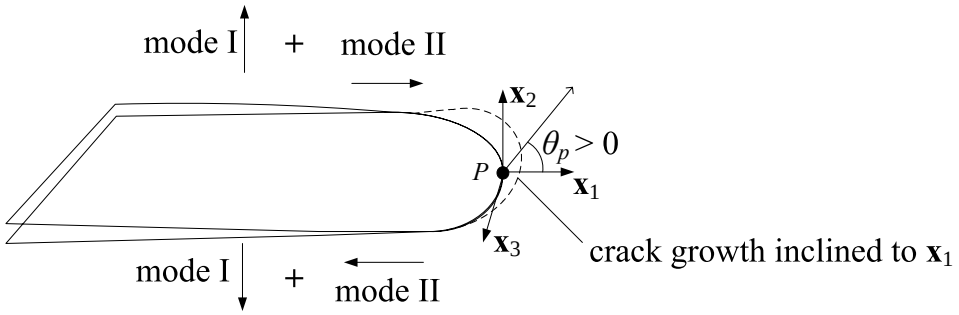
$$\frac{\mathrm{d}a}{\mathrm{d}N} = C(\Delta \mathrm{K})^m \tag{5.63}$$

where a is the length of crack growth, C and m are material constants fit to experimental results, N is the number of loading cycles and ΔK is the range of SIF, i.e. the difference between maximum and minimum SIF at maximum and minimum loading.

$$\Delta \mathrm{K} = \mathrm{K}_{\max} - \mathrm{K}_{\min} \tag{5.64}$$



(a) The crack growth tangent to x_1 in pure mode I



(b) The crack growth inclined to x_1 in mixed mode I and II

Figure 5.17: The crack growth in different modes with respect to the original crack front.

where $\mathrm{K}_{\max}$ and $\mathrm{K}_{\min}$ are the maximum and minimum SIFs in a loading cycle respectively. For a mixed mode crack, K is calculated by an equivalent $\mathrm{K}_{\mathrm{Ieq}}$

$$\mathrm{K}_{\mathrm{Ieq}} = \sqrt{\left(K_{\mathrm{I}} \cos^3 \frac{\theta_p}{2} - 3K_{\mathrm{II}} \cos^2 \frac{\theta_p}{2} \sin \frac{\theta_p}{2}\right)^2 + \frac{E}{2(1-\nu^2)\mu} K_{\mathrm{III}}^2} \ . \tag{5.65}$$

Theoretically, accurate prediction of fatigue crack growth by Equation (5.63) requires accurate SIFs through each cycle. In numerical simulation, this is approximated by

$$\Delta a = C(\Delta \mathrm{K}_{\mathrm{Ieq}})^m \Delta N \ . \tag{5.66}$$

where Δa is set to a small given value and the number of predicted life cycles can be given by integration over a numerically.

5.5.3 Quasi-static crack propagation

Quasi-static fracture refers to crack propagation at or above the fracture toughness. That is the material is stable when

$$\mathrm{K}_{\mathrm{Ieq}} \leq \mathrm{K}_{\mathrm{Ic}} \tag{5.67}$$

where K_{Ic} is the fracture toughness of the material, which is a critical value of the SIF corresponding to G_c (see Chapter 1), and $\mathrm{K}_{\mathrm{Ieq}}$, the equivalent mode I SIF, is calculated according to Equation (5.65). Since it is observed in experiments that if the component is thick enough to resist shear then K_{II} and K_{III} are not critical, K_{Ic} is tested for materials under tensile loading [79]. Materials with large values of fracture toughness tend to undergo ductile fracture while brittle fracture is characteristic of materials with a low fracture toughness [183]. However, there is no generally accepted rule on the rate of crack growth and the crack is assumed to propagate as long as K_{Ieq} is above the threshold. The EFGM has been used in [143] for quasi-static fracture modelling in 2D and the results show fairly good agreement of the predicted crack path with the experiment results.

5.6 Summary

In this chapter, additional components developed in the EFGM for fracture modelling method have been described. Methods for field approximation are summarised into extrinsic and intrinsic enrichments. Test examples in the following chapters in this thesis use the intrinsic enrichment which keeps the number of unknowns unchanged and does not have problems of ill conditioning in the global stiffness matrix. It has been shown that extrinsic enrichment in the EFGM is similar to that used in the XFEM while intrinsic enrichment based on nodal support is however distinctive to the meshless method. These methods, both extrinsic and intrinsic, developed for EFGM are applicable to many other meshless methods based on nodal support. Other components needed for fracture modelling have been described including SIF calculation and crack propagation. The various advantages and limitations are discussed. Field approximation of modelling the stresses and displacements near a crack followed by fracture analysis determining the crack state comprise the two main steps in fracture modelling. In the following chapters, the focus will be given on new methods which simplify further some of these modelling features using level sets.

Chapter 6

Using level sets with the EFGM for fracture modelling

6.1 Introduction

This chapter presents new techniques for fracture modelling combining level sets with the EFGM. The LSM has been previously coupled with the XFEM for 3D fracture modelling in [116] but here we explore its use with the EFGM for 3D crack modelling and address the issues that arise. Level sets build the local curvilinear coordinate systems around the crack front which facilitates the stress analysis as well as the computation of fracture parameters. New formulations to introduce displacement jumps in 2D and 3D for cracks of arbitrary geometry are proposed. Complete flowcharts for updating and advancing the level sets to capture the crack geometry evolution are detailed in this chapter. A number of 2D and 3D crack problems with various geometries are tested showing the feasibility and accuracy of using level sets in this ways for fracture modelling.

6.2 Level sets description of a crack

In fracture modelling, two level sets, namely $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ are used for each crack to describe the crack geometry [116]. The former measures the signed distance normal to the crack surface and the latter measures the signed distance normal to the crack

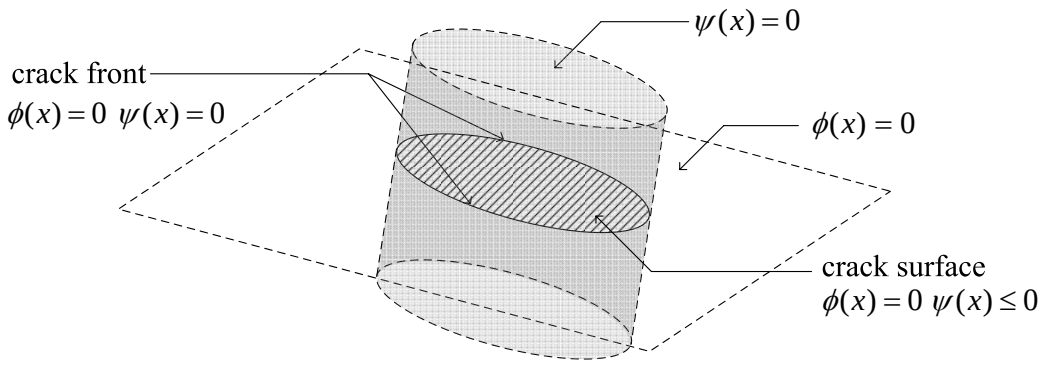


Figure 6.1: Illustration of the level sets description of an elliptic crack surface a in 3D.

front for an arbitrary point of interest $\mathbf{x}$ in the level set domain. The two level sets collect points at the same distance normal to the crack surface and tangential to the crack front respectively as shown in Figure 6.1. Specifically, the zero normal level set $\phi(\mathbf{x}) = 0$ describes the surface containing the crack and the zero tangent level set $\psi(\mathbf{x}) = 0$ gives the plane containing the crack front which is orthogonal to the crack surface as shown in the figure. Similarly, $\phi(\mathbf{x}) = 1$ is the surface with unit distance above of the crack surface, $\phi(\mathbf{x}) = -1$ is the surface with unit distance beneath the crack surface, $\psi(\mathbf{x}) = 1$ is unit distance ahead of the crack front and $\psi(\mathbf{x}) < 0$ are sets of surfaces behind the crack front. The inversion of a certain level set gives the collection of points lying on the contour of a certain level. For example $S^0_\phi(\mathbf{x}) = \{\phi^{-1}(0)\}$ are all the points lying on the crack surface and the extension of the crack surface. Similarly, $S^0_\psi(\mathbf{x}) = \{\psi^{-1}(0)\}$ are all the points lying on the surface which contains the crack front perpendicular to the crack surface. In this section, we describe the use of the level sets with the EFGM for 3D crack modelling.

Unlike the LSM applied in CFD or image processing, the interface in fracture modelling, i.e. the crack surface, is an open curve in 2D or an open surface in 3D. Thus the normal level set $\phi(\mathbf{x})$ is not only defined behind the crack front but also beyond the crack front as shown in Figure 6.1. The zero level set $\phi(\mathbf{x}) = 0$ represents the extended surface containing the crack surface. The part of $\phi(\mathbf{x})$ ahead of the crack front is tangential to the plane where crack front is located as shown in Figure 6.1. It can be seen from the figure that the crack front is the intersection of the two

level sets. Thus the geometry of the crack surface is represented by

$$\begin{aligned} \phi(\mathbf{x}) = 0 \quad \psi(\mathbf{x}) \leq 0 \quad \text{crack surface} \\ \phi(\mathbf{x}) = 0 \quad \psi(\mathbf{x}) = 0 \quad \text{crack front} \end{aligned} \tag{6.1}$$

The two level sets are constructed orthogonal to each other that

$$\nabla\phi(\mathbf{x}) \cdot \nabla\psi(\mathbf{x}) = 0 \tag{6.2}$$

and they work together to form a curvilinear coordinate system defined at the crack front.

6.2.1 Initialization

In the LSM, the initial levels values are assigned to each grid point measuring the signed distance from the grid point to the interface as the starting point for the level sets computation. This procedure is known as initialization. If the analytical solution of the signed distance function of the interface is known, then we will be able construct a zero level set $\phi(\mathbf{x}) = 0$ which exactly represents the interface intended to be described. This can be achieved only for certain types of crack geometry having parameterized description as will be shown in §6.6. The common situation is that initial values are obtained by measuring the distance from a grid point to a sampling point which is located on the interface and is closest to the grid point as

$$\phi(\mathbf{x}, t = 0) = \text{sign}(n_{\mathbf{x}_\Gamma} \cdot (\mathbf{x} - \mathbf{x}_\Gamma)) \min \|\mathbf{x} - \mathbf{x}_\Gamma\| \tag{6.3}$$

where $\mathbf{x}$ is an arbitrary point inside domain, $\mathbf{x}_\Gamma \in \Gamma$, $n_{\mathbf{x}_\Gamma}$ is the outer normal at $\mathbf{x}_\Gamma$ and Γ is the interface which is intended to be described. It can be easily seen that the accuracy of the level sets description is determined by the number and arrangement of the sampling points. Thus the initial values can be "crude", which may produce a 'blurry image' of the exact interface. They are not necessarily the exact signed distance functions to the interface and need to be regularized through a procedure known as "reinitialization". The reinitialization process imposes the signed distance condition on the grid point and requires the magnitude of the gradient of the level sets be unit, i.e. $\|\nabla\phi\| = 1$ throughout the domain. This will be described in more detail in §6.3.6.

6.3 Level sets updating

In the LSM, level sets are updated according to their velocity [184]. When the level sets are used in CFD, the velocity is determined by the velocity of flow which is a real physical quantity. When applying level sets to fracture modelling, level sets are updated according to the rate of crack growth to capture the new crack surface and front as the crack evolves. In this section, the complete "recipe" for the level sets updating in fracture modelling will be described, including the extension of the velocity field from the crack front to the whole domain, the advancement, reinitialization and reorthogonalization of the level sets, and the overall flowchart linking these constituent steps together. These are not new and have been used in the XFEM for fracture modelling [115, 116]. For convenience of later development, they are here reviewed in a systematic way. In the following, the term "time step" is used for solving HJ-PDEs in level sets. Two kinds of time steps are used, namely pseudo time τ and unit time step t.

6.3.1 Computation of SIFs using the level set coordinates

The decomposition of the mixed mode J integral is used in the examples in this thesis since it is simple to implement and accurate providing there are no kinked faces inside the integral path. For convenience of latter derivation, we rewrite the J integral in Equation 5.31 in the form

$$J_k = \int_\Gamma W n_k - t_j^L u_{j,k}^L \mathrm{d}\Gamma \qquad j,k = 1,2 \tag{6.4}$$

where the superscript L indicates all the tensor values are also defined in the local coordinates as shown in Figure 5.12, W is the strain energy density calculated as $W = \frac{1}{2}\boldsymbol{\sigma}^L : \boldsymbol{\varepsilon}^L$ where $\boldsymbol{\sigma}^L$ is the stress tensor, $\boldsymbol{\varepsilon}^L$ is the strain tensor, t_j^L is the traction along Γ calculated by $t_j^L = \sigma_{ij}^L n_j$ and $u_{j,k}^L$ is the derivative of the jth term of the displacement vector with respect to the kth axis. Note here the subscripts j and k indicate the local coordinates defined as in Figure 5.3. When using the J integral for mixed-mode crack problems, J_1 can be be decomposed into symmetric

and anti-symmetric parts as described in [177]

$$J_1 = J_1^{\mathrm{I}} + J_1^{\mathrm{II}}, \tag{6.5}$$

and the SIFs for each mode are given by

$$K_{\mathrm{I}} = \sqrt{\frac{J_1^{\mathrm{I}}}{E^*}} \quad K_{\mathrm{II}} = \sqrt{\frac{J_1^{\mathrm{II}}}{E^*}} \tag{6.6}$$

where E^* is the equivalent Young's modulus calculated by Equation (5.40). Note that the components in Equation (6.4) are based on the local coordinates, hence displacements, stresses and strains need to be mapped to the local coordinates. The use of the LSC in modelling the crack geometry makes this step straightforward. This is even more important in 3D where the domain integral (as opposed to the J integral) is used to determine SIFs [168, 171] so that curvature along the crack surface and front is accounted for (refer to §5.4.4). For the 2D case, the unit base vector of the local coordinates is given by

$$\mathrm{e}_1 = \frac{\nabla\psi}{\|\nabla\psi\|} \tag{6.7}$$

$$\mathrm{e}_2 = \frac{\nabla\phi}{\|\nabla\phi\|}. \tag{6.8}$$

Thus we find the transformation that relates the global tensor with the local tensor as $G = \{G_{mn}\} = [\mathrm{e}_1\ \mathrm{e}_2]^T$ [185]. Since the transformation from global to local e^L is

$$\mathbf{L} = \{L_{mn}\} = \{G_{nm}\} = \left[\mathrm{e}_1^L\ \mathrm{e}_2^L\right]^T \quad m, n = 1, 2. \tag{6.9}$$

Thus the displacement vector in local coordinates can be calculated from the global displacement vector by

$$\mathbf{u}^L = u_i \mathrm{e}_i^L \quad i = 1, 2 \tag{6.10}$$

and similarly the local strain and stress tensors can be calculated as

$$\boldsymbol{\varepsilon}^L = \varepsilon_{ij} \mathrm{e}_i^L \otimes \mathrm{e}_j^L \tag{6.11}$$

$$\boldsymbol{\sigma}^L = \sigma_{ij} \mathrm{e}_i^L \otimes \mathrm{e}_j^L \quad i, j = 1, 2. \tag{6.12}$$

where "$\otimes$" denotes the dyadic product of two vectors to form a second order tensor. We note here the components on the *r.h.s.* do not necessarily correspond to the

corresponding terms on the *l.h.s.* since the global unit based vectors can cross over with each other in the local coordinates. In this situation the component on the *r.h.s.* is the sum of non-zero projection terms on the *l.h.s.*. Thus the new local tensors are the products of the global tensors. It can be seen that the LSC play the role of defining the local coordinates system around the crack. For a curved crack in 2D or 3D, the LSC is actually the curvilinear coordinates system defined at the crack tip.

6.3.2 Computing front velocity from SIFs

The crack growth rate, i.e. the magnitude of crack propagation in a unit time step, is governed by the SIFs along the crack front. Since the crack geometry is described by level sets and each level set is updated according to the crack growth, the magnitude of level set velocity can be determined by crack growth rate

$$\|\mathbf{V}\| = \frac{\Delta a}{\Delta N} \tag{6.13}$$

where Δa is the length of crack growth and which can vary along a crack front, and N is a pseudo-time parameter. In fatigue crack propagation, N corresponds to the number of loading cycles, as in Equation (5.63)

$$\frac{\Delta a}{\Delta N} = C\left(\Delta K_{\mathrm{Ieq}}\right)^m \ . \tag{6.14}$$

where C and m are material parameters. For quasi-static crack propagation, N is a parameter to scale the length of crack growth in each propagation step. However there is no generally accepted criterion such as the Paris law in fatigue crack propagation. In both cases, Δa is affected by the equivalent mode I SIF K_{Ieq} indicating the magnitude of K_{I}, K_{II} and K_{II} as shown by Equation (5.65). It can be seen that crack growth rate increases with K_{Ieq}. There is a threshold value of SIF K_{Ic} (same as that in Equation (5.67)) for each type of material so that when K_{Ieq} reaches K_{Ic}, the material becomes unstable indicating failure. Correspondingly in fracture modelling, the calculation stops when the obtained K_{Ieq} reaches K_{Ic} and the number loading cycles at this point is the maximum allowable loading cycles. As shown in Equation (6.14), with material parameters C and m known, the crack

growth rate is at a point on crack front is determined by K_{Ieq}. There is threshold value of SIFs for each type of material that when the threshold value is reached, the crack becomes unstable. In fracture modelling, the calculation stops and the obtained number loading cycle given are the maximum allowable loading cycles. A note is added after Eq. 6.15 to clarify this point. In 3D fracture modelling K_{Ieq} can vary along a crack front, therefore the crack growth also varies. To determine the size of crack growth along a crack front, firstly the point lying on the crack front having the maximum K_{Ieq} is found, and the crack growth at that point is assigned with increment denoted as $\Delta a_{\max}$ (specified by the user regarding the requirement of the resolution of crack path). Then for all the other points having K_{Ieq} less then $K_{\text{Ieq}}^{\max}$, the following power relation is used to calculate Δa

$$\Delta a = \left(\frac{K_{\text{I}}}{K_{\text{Ieq}}^{\max}} \right)^{m} \Delta a_{\max} \ . \tag{6.15}$$

The direction of crack growth, or crack growth angle θ_p is uniquely determined by the ratio between K_{I} and K_{II} as shown by Equation (5.62). Mode III SIF does change crack growth angle, however it also affects K_{Ieq} and hence growth rate. With the magnitude and angle θ_p of velocity given, the front velocity can be projected to the level set coordinates as

$$\mathbf{V} = \{V_\phi,\, V_\psi\} = \frac{\Delta a}{\Delta N} \left\{ \cos\theta_p \mathbf{n}_\psi, \sin\theta_p \mathbf{n}_\phi \right\} \ . \tag{6.16}$$

In implementation, a number of of points are sampled along the crack front to calculate SIFs, after which V_ϕ and V_ψ are determined at the crack front.

6.3.3 Extension of the velocity field

In §6.3.2 the velocity along the crack front is defined based on the SIFs. However this 1D field along the crack front provides insufficient information to since the advancement requires the velocity defined throughout the domain. Thus the velocity along the front needs to be extended to the entire level set domain. In [116], a velocity extension method is proposed following the ideas developed in [186, 187] for capturing two evolving curves in 3D. In this method V_ψ is firstly extended to neighbouring points of the interface, i.e. the grid points as shown in Figure. Then

the velocity field is extended from the vicinity of the crack to the whole domain by solving the following equations for V_ψ with respect to τ to steady state

$$\frac{\partial V_\psi}{\partial \tau} + \text{sign}(\phi)\frac{\nabla \phi}{\|\nabla \phi\|} \cdot \nabla V_\psi = 0 \tag{6.17}$$

$$\frac{\partial V_\psi}{\partial \tau} + \text{sign}(\psi)\frac{\nabla \psi}{\|\nabla \psi\|} \cdot \nabla V_\psi = 0 \tag{6.18}$$

where τ is a time-like parameter for calculation not the real time, and the size of τ in each iteration is determined by the CFL condition (named after the three mathematicians Courant, Friedrichs and Lewy who contribute to form this theory) [188] which states in solving a time-space PDE (normally a hyperbolic PDE) an upwind scheme be convergent only if the following condition is satisfied as

$$\frac{V^i \Delta t}{\Delta x^i} \leq d_L \quad (d_L \leq 1) \ , \tag{6.19}$$

where V^i is the velocity component in ith dimension, Δt is the size of a time integration step and Δx^i is the grid size in i th dimension and d_L is a constant parameter depending on the approximation type and integration scheme which is normally set unit in FDM. Test examples in this thesis use the following criterion to determine the size of time step Δt [112]

$$\Delta t = \max\left(\sum_{i=1}^{\dim} \frac{|V_j^i|}{\Delta x_j^i}\right)^{-1} \quad \forall \ x_j^i \in \text{grid} \ , \tag{6.20}$$

where dim is the number of dimensions and dim = 3 for 3D problems. Note V^i is a general symbol indicating a velocity field for the variable to be solved e.g. in the advancement of ϕ, V^i corresponds to ith component of V_ϕ while in solving Equation (6.17) for V_ϕ, it becomes ith component of the vector $\text{sign}(\phi)\frac{\nabla \phi}{\|\nabla \phi\|}$. The steady state here refers to a situation where the change of V_ψ between successive steps is less than a tolerance ϵ which is set by the user. In this work $\epsilon = 10^{-3}$ is used. In solving each of these equations, an explicit time integration scheme, the third order accurate total variation diminishing Runge-Kutta (TVD RK) scheme [188], is used to solve for V_ψ with respect to τ (refer to Appendix B.3 for details). The spatial derivatives for V_ψ are found using the weighted essentially non-oscillatory schemes (WENO) (refer to Appendix B.2 for details). At steady state V_ψ is close to constant, it is therefore orthogonal to $\phi(\mathbf{x})$ as shown by the second term on the

r.h.s in Equation (6.17) and also orthogonal to $\psi(\mathbf{x})$ as shown by the second term on the *r.h.s* in Equation (6.18). Following a similar procedure, V_ϕ is extended to the entire domain by solving the following equations to steady state, i.e. V_ϕ converges with respect to τ so that the change of V_ϕ with the increment of τ is less than a given tolerance.

$$\frac{\partial V_\phi}{\partial \tau} + \text{sign}(\phi)\frac{\nabla \phi}{\|\nabla \phi\|} \cdot \nabla V_\phi = 0 \tag{6.21}$$

$$\frac{\partial V_\phi}{\partial \tau} + \text{sign}(\psi)\frac{\nabla \psi}{\|\nabla \psi\|} \cdot \nabla V_\phi = 0 \tag{6.22}$$

where τ is again a time-like parameter. The above described processes result in the velocity fields V_ϕ orthogonal to both ϕ and ψ, and V_ψ orthogonal to both ϕ and ψ. This is illustrated in Figure 6.3 showing that the extension of each velocity field along the ortho-lines of two level sets. Thus when the velocity field is used to advance the level set, it is along the gradient of each level set which fits in the framework of the LSM.

6.3.4 Advancing the level sets

The level sets are advanced according to the velocity along the gradient of the level sets. This yields a time dependent evolution equation [184]

$$\frac{\partial \phi(\mathbf{x})}{\partial t} + V_\phi \|\nabla \phi(\mathbf{x})\| = 0 \tag{6.23}$$

where V_ϕ is the velocity field defined throughout the entire domain, and at any point inside the domain the magnitude of the change of ϕ at that point is equal to the value of V_ϕ at that point and is directed along the gradient of ϕ. This means that V_ϕ determines the change of ϕ along $\|\nabla \phi(\mathbf{x})\|$. With the velocity field known throughout the domain, the level sets ϕ and ψ are advanced by solving the following equations [116]

$$\frac{\partial \phi(\mathbf{x})}{\partial t} + V_\phi \|\nabla \phi(\mathbf{x})\| = 0 \tag{6.24}$$

$$\frac{\partial \psi(\mathbf{x})}{\partial t} + V_\psi \|\nabla \psi(\mathbf{x})\| = 0 \tag{6.25}$$

where t is a unit time step different from τ defined in Equation (6.22). The unit step t is divided into smaller substeps and the number of divisions can be determined by

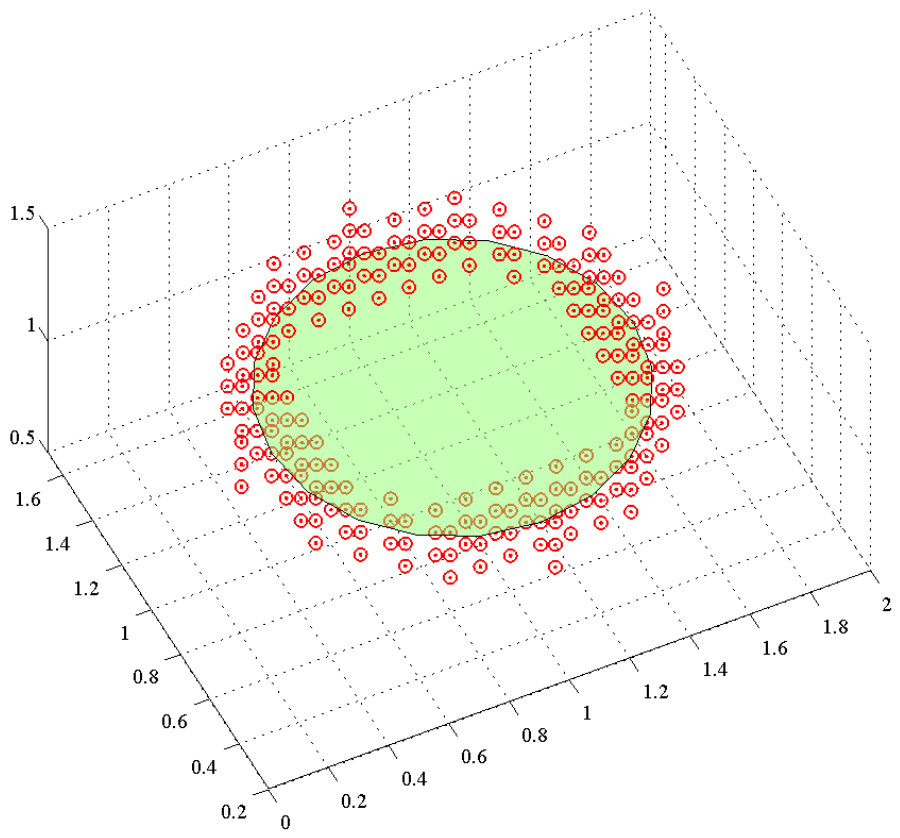
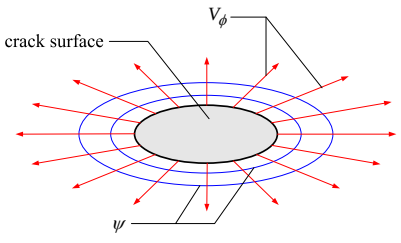
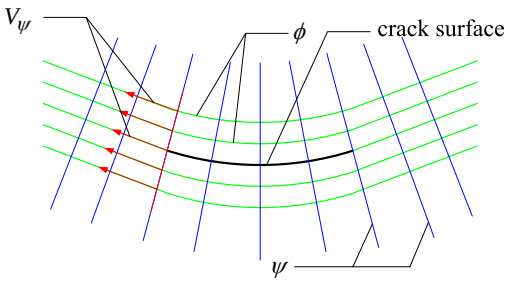


Figure 6.2: The grid points which are chosen near the crack front to start velocity extension.



(a) top view of the extension of velocity



(b) side view of the extension of velocity

Figure 6.3: The extension of V_ϕ and V_ψ along the orthogonal directions of ϕ and ψ.

the CFL condition as previously discussed in Equation (6.20). We note the above equations are in the form of standard level set convection equations. However as it is pointed out in [189] Equation (6.24) overestimates the orthogonality condition between the velocity field V_ϕ and gradient of the level set $\|\nabla\phi(\mathbf{x})\|$. The consequence is that the increment of crack growth will drift away from the correct answer. Thus it is proposed that the orthogonality condition be imposed to the original gradient of ϕ at $t = 0$ instead of having ϕ updated in each substep as follows

$$\frac{\partial\phi(\mathbf{x})}{\partial\tau} + V_\phi \nabla\phi_0 \cdot \nabla\phi(\mathbf{x}) = 0 \ . \tag{6.26}$$

In this thesis, this corrected equation is used to advance the ϕ level set.

It has been shown that the time step used to extend V_ϕ or V_ψ is a pseudo-time noted as τ and that used to advance the level sets are unit time noted as t. The value of pseudo time τ is not known at the beginning of calculation but grows gradually with iterations until the calculation converges while t is set to be equal to one.

6.3.5 Reorthogonalization

At the starting point of the level set computation, where no velocity fields are imposed, the two level sets are initialized as being orthogonal to each other. After the advancement of the level sets, the orthogonality condition may not be preserved. Thus at some point of the calculation, normally after each step of level sets advancing, we stop and enforce the two level sets to be orthogonal. This process is known as the reorthogonalization. The prefix "re" in the word "reorthogonalize" refers to recover the orthogonality condition as it is in the beginning. We refer to the method in [190] to reorthogonalize the level sets by solving the following equation to steady state

$$\frac{\partial\psi}{\partial\tau} + \text{sign}(\phi_0)\frac{\nabla\phi}{\|\nabla\phi\|}\nabla\psi = 0 \tag{6.27}$$

where ϕ_0 is value of ϕ at $\tau = 0$. At steady state when ψ almost does not change, $\nabla\phi \cdot \nabla\psi = 0$ meaning ϕ and ψ orthogonal. The reorthogonalization converges quickly after a number of iterations. An example demonstrates reorthogonalization will be presented in §6.6.

6.3.6 Reinitialization

Reinitialization refers to the termination of the level set computation after advancement and the reconstruction of level sets, both of ϕ and ψ here, signed distance functions [188]. The reason for doing this is that the level sets may be stretched or distorted after evolution so the condition of a signed distance function is not maintained. We refer to [191, 192] to reinitialize the level sets by solving the following HJ-PDE to steady state

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi_0)(\|\nabla \phi\| - 1) = 0 \tag{6.28}$$

It can be seen from Equation (6.28) that at steady state $\phi(\mathbf{x})$ will satisfy the distance condition, i.e. that $\|\nabla \phi\| = 1$. The tangent level sets $\psi(\mathbf{x})$ can also be initialized through a similar procedure

$$\frac{\partial \psi}{\partial \tau} + \text{sign}(\psi_0)(\|\nabla \psi\| - 1) = 0 \tag{6.29}$$

It should be noted that the reinitialization is performed after each step of evolution of ϕ and ψ to ensure that the orthogonality condition of the two level sets is maintained. It is an expensive but necessary step in level sets advancement.

6.3.7 The level sets updating process

The previous sections describe the ingredients for updating. Now we summarise the complete process in the following:

1. Initialize the two level sets throughout the level set domain. This can be done either numerically, using a set of discretized points on the interface known *a priori*, or analytically if the interface has an analytical expression of geometry, i.e. having a parametric function of geometry.

2. Reinitialize the two level sets ϕ and ψ respectively using the procedure described in §6.2.1 so that each level set becomes a signed distance functions.

3. Enforce the orthogonality condition between ϕ and ψ according to the procedure described in §6.3.5.

4. Advance the level sets by solving the advancement equations in §6.3.4.

5. After advancing the level sets, reinitialize ϕ, again according to §6.3.6.

6. Reorthogonalize ψ with respect to ϕ according to §6.3.5 and reinitialize ψ according to §6.3.6.

6.4 The jump term in the EFGM based on the LSC

Here we look at an issue that arises when the LSM is used for introducing the displacement jump across a crack surface in 3D with the EFGM, and highlight differences compared to its use in the XFEM. When the LSM is applied to the XFEM, the Heaviside step function can be uniquely determined by the sign value of the normal function, i.e. $H(\mathbf{x}) = \text{sign}(\phi(\mathbf{x}))$ where $\phi(\mathbf{x})$ is essentially the same as the normal level set described previously. However this rule does not work in the EFGM because the weight function associated with a node should be bounded by the crack front while the Heaviside step function is unbounded throughout the domain. In this section, we proposed a new formulation based on the LSC for the two commonly used methods of introducing the jump terms in the EFGM, namely the visibility criterion and the diffraction method (refer to §5.3.1 for the definition of these two methods).

6.4.1 Visibility criterion based on the LSC in 2D

We now describe a simple formulation to introduce the displacement jump term using the visibility criterion based on level sets. The formulation is effective for single or multiple crack problems.

Here we propose a simple procedure for using the visibility criterion based on the LSC. Consider two intersecting cracks, A and B as shown in Figure 6.4. Suppose the point of interest $\mathbf{x}$ is located near the cracks with four nearby nodes (marked $1 \ldots 4$) lying in different zones delineated by the zero level sets of ϕ_A and ϕ_B. Each of the four nodes has $\mathbf{x}$ inside its support and is cut by both A and B. For node 1,

the signs of ϕ_A and ϕ_B are both opposite to those of $\mathbf{x}$ and this node is therefore invisible to $\mathbf{x}$. Node 2 lies on the same side of $\mathbf{x}$ with respect to B but the other side for A. Thus for A, node 2 is invisible to $\mathbf{x}$ but not for B. Similarly for node 4 reversing A and B. Neither node 2 or 4 will influence the approximation at $\mathbf{x}$ as they are invisible to $\mathbf{x}$ for one of the cracks, thus the link between them is severed. Node 3, always lies on the same side of $\mathbf{x}$ with respect to either A or B and is therefore used in the approximation at $\mathbf{x}$. This simple example demonstrates the simplicity of the visibility criterion even with multiple cracks. If a node is separated from a point by one crack, then the node will not be used to approximate the field at the point. To contrast, in the diffraction method a node's support may bend around more than one crack, becoming "partially" visible to the point of interest, as shown in Figure 5.9. The rule demonstrated using Figure 6.4 is effective for nodal supports completely cut by cracks but not valid for nodal supports only partially cut. In this case we need to carefully judge whether the point $\mathbf{x}$ is separated from the node using an additional check.

Consider Figure 6.5 with a a node I having its support partially cut by the crack. It can be seen that if $\mathbf{x}$ falls inside the shaded region it is visible to node I, or formally

$$\begin{aligned} \psi(\mathbf{x}) &\geq -\psi_I \left| \frac{\phi(\mathbf{x})}{\phi_I} \right| \quad \text{visible} \\ \psi(\mathbf{x}) &< -\psi_I \left| \frac{\phi(\mathbf{x})}{\phi_I} \right| \quad \text{invisible.} \end{aligned} \tag{6.30}$$

Therefore the procedure using the visibility criterion based on the LSC can be summarized as follows:

1. Calculate the level set values $\phi^i(\mathbf{x})$ at $\mathbf{x}$ and $\{\phi_I^i\}$ $(i = 1, 2, ...N_K)$ for all the nodes in support of $\mathbf{x}$ without the presence of cracks. Here N_K is the total number of cracks and definition of I is same as in Equation (7.4).

2. `Loop` over the $\{\phi_I^i\}$

   `if` $\forall\, i, \quad \phi_I^i \phi^i(\mathbf{x}) > 0$

   node I is used in the approximation at $\mathbf{x}$.

   `else`

   judge the visibility according to Equation (6.30)

3. If node I is found invisible for any crack, then eliminate I from the nodes in support.

The loop in step 2 ensures that Equation (6.30) is applied only to cases where ϕ values have opposite signs. The above formulation is valid for 2D problems where the crack front is a point. We also note the differences between the present formulation from the existing published approaches. For instance in [5] the diffraction method is extended for multiple crack interaction for the EFGM however it is not based on level sets and requires complicated geometric operations. Formulations for dealing with conjunctions and intersections between multiple cracks is proposed for the extended EFGM using piecewise triangular facets in [113]. The underlying ideas in these examples, for dealing with multiple cracks, are somewhat similar however the formulations are different from that of the present method. The above formulation is valid for 2D problems where the crack front becomes a point. An extension to 3D, where the crack front is a line, is straightforward using the LSC and is described in [193] and will be detailed in the following section.

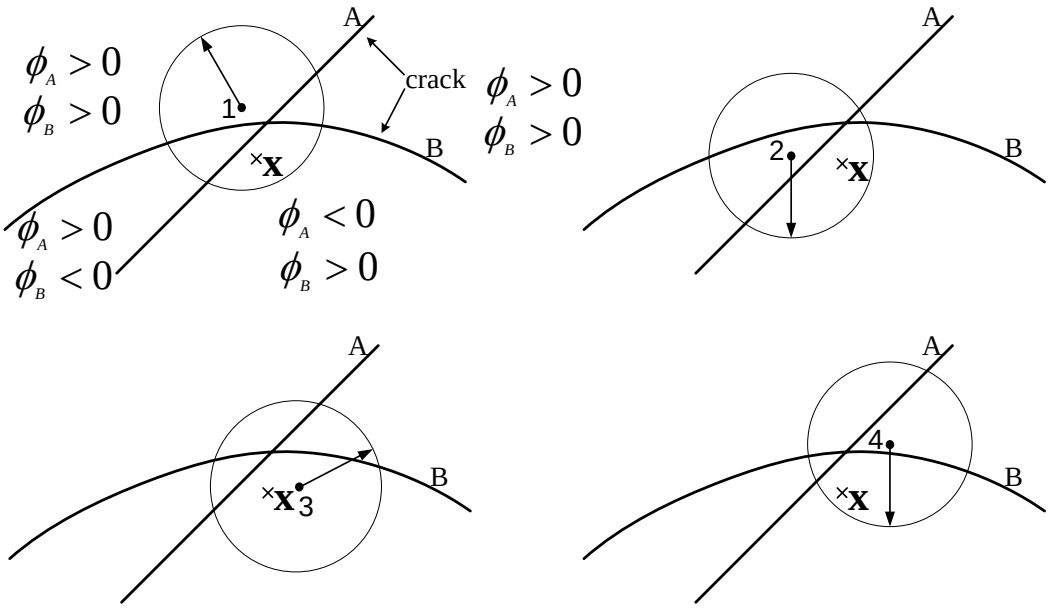


Figure 6.4: The visibility criterion for multiple cracks.

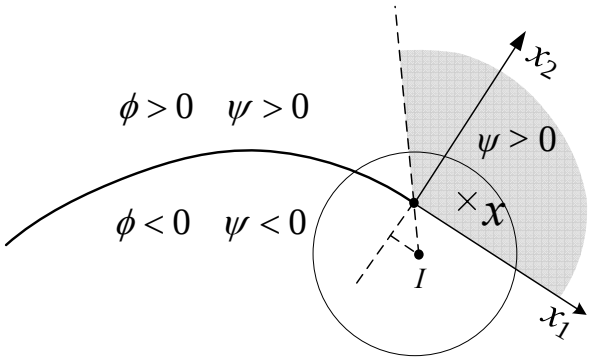
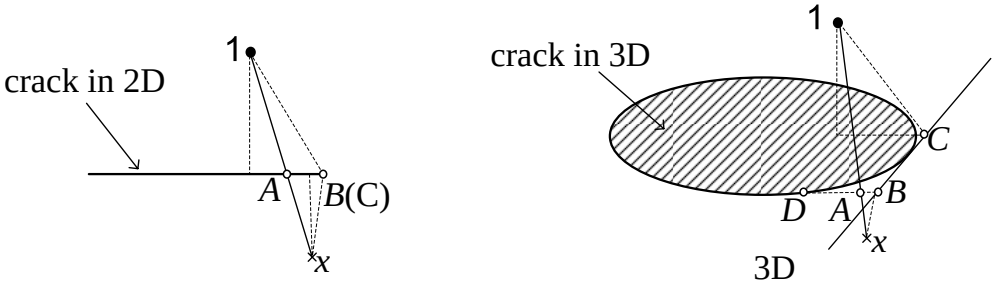


Figure 6.5: Nodal support partially cut by the crack.



(a) Crack front as a tip in 2D.

(b) Curved crack front in 3D.

Figure 6.6: The differences applying the visibility criterion to a crack in 2D and a crack with a curved front in 3D.

6.4.2 Visibility criterion based on the LSC in 3D

The formulation of the visibility criterion based on the LSC for a single crack is proposed in [108], and in [193] the formulation for multiple cracks is proposed. Both the methods are effective in 2D fracture modelling, however neither of them is valid in 3D for curved crack fronts. We show this by comparing a 3D penny-shaped crack with a 2D straight crack. Consider a penny-shaped crack as shown in Figure 6.6(b) which has a curved crack front. Suppose the point of interest $\mathbf{x}$ is located close to the crack and is included in the support of node 1 before considering the crack. Then we use the visibility criterion to judge if the line linking node 1 and $\mathbf{x}$ intersects the crack and if so node 1 will be excluded from the approximation at $\mathbf{x}$. Since the crack front is curved, we find that the nearest point on the crack front to $\mathbf{x}$ and to node 1 can be two different points noted as B and C respectively if crack front is assumed straight. From the figure, we can see point C is on the crack front while B is outside the crack surface. Although the line linking node 1 and $\mathbf{x}$ intersects the plane containing the crack surface, the intersecting point A is outside the crack surface since the crack front is curved. In contrast consider a 2D straight crack which is can be viewed as a section through penny-shaped crack as shown in Figure 6.6(a). Since the crack front becomes a point, then B and C coincide. In this case, using the visibility criterion requires only determination if the line linking 1 and $\mathbf{x}$ intersects the crack line (exactly what is described in [108] and [193]). The major difference in applying the visibility criterion between 2D and 3D is clear. Below is a formulation effective for dealing with a 3D crack surface with an arbitrary curved front.

1. Consider a point of interest $\mathbf{x}$ and a set of N nodes in support of $\mathbf{x}$.

2. Loop over each node $I\,(I = 1, 2, \cdots, N)$ for $\mathbf{x}$ to find the point $\mathbf{x}_c$ which satisfies the following conditions

$$\begin{cases} \mathbf{x}_c &= t(\mathbf{x} - \mathbf{x}_I) \quad (0 < t < 1) \\ \phi(\mathbf{x}_c) &= 0 \end{cases} \tag{6.31}$$

   Thus the point $\mathbf{x}_c$ found is the intersecting point of the line linking $\mathbf{x}$ and node I and the crack surface. To determine $\mathbf{x}_c$, we use the following simple iteration

$\mathbf{x}_c = \mathbf{x}_I$

`while` $|\phi(\mathbf{x})| > \mathrm{e}_{\mathrm{tol}}$

$$t = \frac{\phi(\mathbf{x}_c)}{\phi(\mathbf{x}_c) - \phi(\mathbf{x})}$$

$$\tilde{\mathbf{x}}_c = \mathbf{x} - t(\mathbf{x}_c - \mathbf{x})$$

$$\mathbf{x}_c = \tilde{\mathbf{x}}_c$$

`end`

where $\mathbf{x}_I$ is the Cartesian coordinates of node I, $\tilde{\mathbf{x}}_c$ is the predicted intersecting point in an iteration and $\mathrm{e}_{\mathrm{tol}}$ is the tolerance to stop the iteration and $\mathrm{e}_{\mathrm{tol}} = 0.001$ is used in the test. Since the node I and point of intersect is in the vicinity of crack, the iteration converges within one or two times.

3. If $\psi(\mathbf{x}_c) > 0$, i.e. point $\mathbf{x}_c$ falls outside the crack surface, then node I is excluded from the approximation at $\mathbf{x}$, otherwise node I included.

6.4.3 The diffraction method based on the LSC

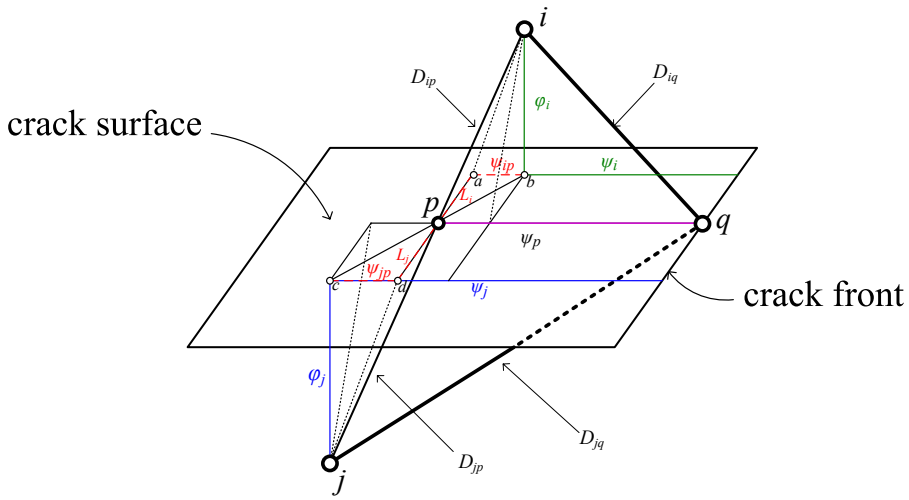


Figure 6.7: The diffraction method based on the level sets in 3D.

The previous section presented a formulation for the the visibility criterion based on the LSC. In this section, we derive the diffraction method based on the LSC for completeness. Consider a part of a crack surface as shown in Figure 6.7 and a pair of points i and j lying on either side of the crack. The diffraction method requires determination of the shortest path bending around the crack front from i to j, the

path is not being allowed intersect the plane. Here we assume point i and j are very close to the crack surface, and the curvatures of the crack surface and front are zero. Suppose the line linking i and j intersects the crack surface at q and the shortest path from i to j is iqj. The purpose of the following derivation is to find out the diffraction distance

$$d_{ij} = D_{iq} + D_{jq} \tag{6.32}$$

and its derivatives

$$d'_{ij} = D'_{iq} + D'_{jq} \tag{6.33}$$

based on the LSC as the distance d_{ij} is essentially d_{mI} between a node and a point of interest and is needed in calculating the weight functions and their derivatives to obtain the EFGM shape functions (refer to §2.2.1 and §2.2.2). From Figure 6.7, it can be easily found that

$$D_{iq} = \sqrt{\phi_i^2 + \psi_i^2 + L_i^2} \tag{6.34}$$

$$D_{jq} = \sqrt{\phi_j^2 + \psi_j^2 + L_j^2} \ . \tag{6.35}$$

Since ϕ_i, ϕ_j, ψ_i and ψ_j are known from the LSC, we need only find L_i and L_j. Firstly, the distance between i and j is given as

$$D_{ij} = D_{ip} + D_{jp} \ . \tag{6.36}$$

By examining the geometry in Figure 6.7 we find is the proportional relation exists

$$\frac{L_j}{L_i} = \frac{D_{jp}}{D_{ip}} = \frac{\phi_j}{\phi_i} = k \ . \tag{6.37}$$

thus D_{ip} can be expressed in terms of D_{ij} as

$$D_{ip}^2 = \frac{1}{(1+k)^2} D_{ij}^2 \ . \tag{6.38}$$

Substitute Equation (6.38) into (6.37) to solve for L_i

$$L_i = \sqrt{\frac{1}{(1+k)^2} D_{ij}^2 - \psi_{ip}^2 - \phi_i^2} \tag{6.39}$$

and $L_j = kL_i$, $D_{jp} = kD_{ip}$. Alternatively we can solve for L_i and D_{jq} as

$$L_j = \sqrt{\frac{k^2}{(1+k)^2} D_{ij}^2 - \psi_{jp}^2 - \phi_j^2} \tag{6.40}$$

$$D_{jq} = \sqrt{\phi_j^2 + \psi_j^2 + L_j^2} \ . \tag{6.41}$$

Substitute L_i and L_j into D_{iq} and D_{jq} respectively to obtain

$$D_{iq} = \sqrt{\frac{1}{(1+k)^2} D_{ij}^2 - \psi_p^2 + 2\psi_i\psi_p} \tag{6.42}$$

$$D_{jq} = \sqrt{\frac{k^2}{(1+k)^2} D_{ij}^2 - \psi_p^2 + 2\psi_j\psi_p} \tag{6.43}$$

In the 2D case, $L_i = 0$ and $L_j = 0$, thus

$$D_{iq} + D_{jq} = \sqrt{\phi_i^2 + \psi_i^2} + \sqrt{\phi_j^2 + \psi_j^2} \ . \tag{6.44}$$

Denote the first term on the $r.h.s$ of D_{iq} and D_{jq} as

$$F_i = \frac{1}{(1+k)^2} D_{ij}^2 \tag{6.45}$$

$$F_j = \frac{k^2}{(1+k)^2} D_{ij}^2 \ . \tag{6.46}$$

Applying the chain rule to Equation (6.45), we get the derivatives of F_i and F_j

$$F_i' = \frac{-2k'}{(1+k)^3} D_{ij}^2 + \frac{2}{(1+k)^2} D_{ij} D_{ij}' \tag{6.47}$$

$$F_j' = \frac{2kk'}{(1+k)^3} D_{ij}^2 + \frac{2k^2}{(1+k)^2} D_{ij} D_{ij}' \ . \tag{6.48}$$

Substitute F_i' and F_j' into Equation (6.43), then the derivatives of D_{iq} and D_{jq} are

$$D_{iq}' = \frac{1}{2D_{iq}} (F_i' - 2\psi_p\psi_p' + 2\psi_i\psi_p') \tag{6.49}$$

$$D_{jq}' = \frac{1}{2D_{jq}} (F_j' - 2\psi_p\psi_p' + 2\psi_j'\psi_p + 2\psi_j\psi_p'). \tag{6.50}$$

The derivatives in Equation (6.47) and (6.48) requires us to find k' and ψ_p'. We start with the relation in Equation (6.57)

$$k = \left| \frac{\phi_j}{\phi_i} \right| \tag{6.51}$$

and then the derivative of k is found as

$$k' = \frac{\phi'_j}{k}\frac{\phi_j}{\phi_i^2} \ . \tag{6.52}$$

To find out ψ'_p, we start with the proportional relation that

$$k = \frac{\psi_{ip}}{\psi_{jp}} \tag{6.53}$$

where

$$\psi_{ip} = \psi_p - \psi_i \tag{6.54}$$

$$\psi_{jp} = \psi_j - \psi_p \ . \tag{6.55}$$

Therefore ψ_p can be expressed as

$$\psi_p = \frac{k\psi_i + \psi_j}{1+k} \tag{6.56}$$

and its derivatives can be found using the chain rule as

$$\psi'_p = \frac{(k'\psi_i + \psi'_j)(1+k) - k'(k\psi_i + \psi_j)}{(1+k)^2} \tag{6.57}$$

Once we have obtained k' and ψ'_p, we substitute them into Equation (6.50) and Equation (6.33) to finally find the derivatives of d'_{ij}. The present formulation gives a new and general formulation of the diffraction method based on the LSC. However it requires the nodal support to be small so that within each nodal support the curvature along the crack front is almost zero. Besides in our test, we find the computational cost is very expensive using the above described formulation. For these reasons the visibility criterion approach is adopted in the examples to be presented later in this thesis.

6.5 Asymptotic enrichment based on the LSC

One of the key issues in fracture modelling for arbitrary crack geometry is determination of the local coordinate system along the crack front. It is needed both for the asymptotic enrichment described in §5.3.2 and in the J integral or domain integral

calculations for the SIFs (refer to §5.4). With the help of level sets, this can be done simply. The local angle θ is given by (refer to §5.4.2 for the definitions of r and θ)

$$\theta = \tan^{-1}\left(\frac{\phi(\mathbf{x})}{\psi(\mathbf{x})}\right) \tag{6.58}$$

and the local radius r at a point $\mathbf{x}$, which corresponding to the shortest distance from $\mathbf{x}$ to the crack front, is given by

$$r = \sqrt{\phi(\mathbf{x})^2 + \psi(\mathbf{x})^2} \tag{6.59}$$

Determining r and θ using explicit methods will require some effort in two-dimensions; and in three dimensions, considerable care must be taken to take into account the curvature along the crack surface and front. In contrast, the use of level sets greatly reduces the complexity of building of the local coordinate system, and has a general formulation for arbitrary curved cracks in 3D.

6.6 Numerical examples

In this section, we demonstrate the use of level sets in representing crack geometries with a number of typical 2D and 3D crack examples. These examples have various geometries in 2D or 3D and each is used to demonstrate various issues in level set descriptions of cracks. Some of the examples will be used in the later chapters for crack modelling in 2D and 3D. Here we do not apply any mechanical analysis to any of these examples. The purpose is to demonstrate the procedures described above. To estimate the error of the level sets description of the geometry from the exact geometry, we define the following

$$\mathbf{x}_{\text{lsc}} = \{\phi^{-1}(0)\} \cap \{\psi^{-1}(0)\} \tag{6.60}$$

$$\mathrm{e}_{\text{lsc}} = \frac{\|\mathbf{x}_{\text{lsc}} - \mathbf{x}_{\text{exact}}\|}{\|\mathbf{x}_{\text{exact}}\|} \tag{6.61}$$

where $\|\cdot\|$ is the L_2 norm of a vector (refer to Equation (1.21)), $\mathbf{x}_{\text{lsc}}$ is the crack geometry given by the intersection of the two level sets and e_{lsc} is the relative error of crack geometry between the level sets and exact solutions. In implementation, a set of discretized points are sampled to calculate Equations (6.60) and (6.61).

6.6.1 Edge crack initialization

Firstly we look at the level sets representation of a 2D straight edge crack as shown in Figure 6.8. We start with this simplest case in order to show the basic concepts of the level sets initialization and orthogonalization. Firstly, the two level sets describing an edge crack are found as

$$\phi(\mathbf{x}) = y - y_c \tag{6.62}$$

$$\psi(\mathbf{x}) = x - x_c \tag{6.63}$$

where $\mathbf{x} = \{x, y\}$ are the coordinates of a point, and $\{x_c, y_c\}$ the coordinates of the crack tip. In this example, the two ends of crack line are positioned at $\{-1, 0\}$ and $\{0, 0\}$. The level set domain is $[-1, 1] \times [-1, 1]$ and the structured grid points are set at 0.1 intervals in both dimensions. The plot shows the different values of the level sets, the crack line as the intersection of the zero levels of the two level sets and the crack tip shown as the asterisk.

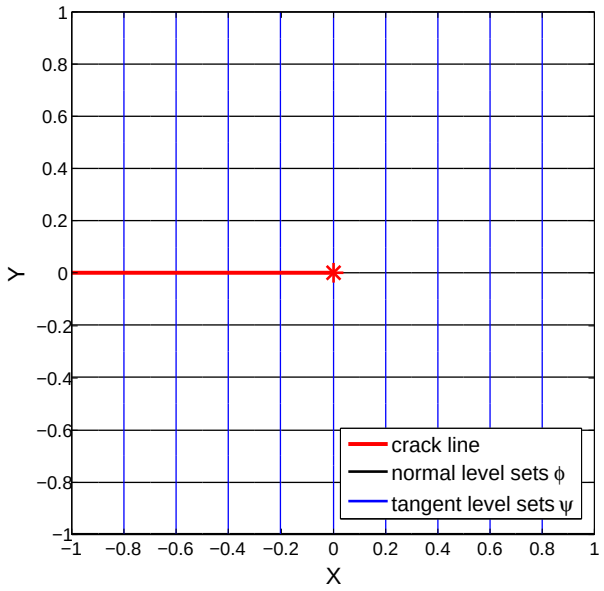


Figure 6.8: The level sets representation of a 2D edge crack.

6.6.2 Edge crack reorthogonalization

Now consider a situation when ψ is no longer orthogonal to ϕ after advancement in level set calculation. We assume ϕ aligns exactly with the crack line and ψ was

drifted from being orthogonal to ϕ and needs to be reorthogonalized, as shown in Figure 6.9. We stop the level set computation to reorthogonalize the two level sets using the procedure described in §6.3.5. The two level sets after reorthogonalization are plotted in Figure 6.10 which shows that ψ is now orthogonal to ϕ however it is no longer a signed distance function. The ψ values along the grid points from $\{x = -1 : 0.1 : 1\}$ are $\{-0.707 : 0.0707 : 0.707\}$ while the exact solution is $\{-1 : 0.1 : 1\}$ meaning the tangent level set ψ is spuriously stretched in the x direction. This is reflected by the contour plot in Figure 6.10 of radius r which is stretched in the same direction as ψ. We then reinitialize ψ according to §6.3.6 and the final results are plotted in Figure 6.11. It can be seen that after reinitialization ψ is a signed distance function and the contour plot of r becomes a set of circles centered on the crack tip. In Table 6.1 the ψ values are listed and for this problem the ψ values with grid points in same column. It shows that the procedure for reorthogonalization and reinitialization leads to less than 0.003% error as estimated by e_{lsc}. Table 6.2 shows the convergence of the error in reorthogonalization which converges quickly to less than 10^{-6} after 2.5 time steps.

x	-1.0	-0.9	-0.8	-0.7	-0.6	-0.5	-0.4
$\psi(\mathbf{x})$	-0.99992	-0.89999	-0.80000	-0.70000	-0.60000	-0.50000	-0.40000
x	-0.3	-0.2	-0.1	0	0.1	0.2	0.3
$\psi(\mathbf{x})$	-0.30000	-0.20000	-0.10000	9.0763e-20	0.10000	0.20000	0.30000
x	0.4	0.5	0.6	0.7	0.8	0.9	1
$\psi(\mathbf{x})$	0.40000	0.50000	0.60000	0.70000	0.8000	0.89999	0.99993

Table 6.1: ψ after reorthogonalization and reinitialization.

Pseudo time τ	0.5	1	1.5	2	2.5
Relative error (%)	52.0313	26.6547	2.5419	0.0437	0.0003

Table 6.2: Convergence of the error of the reorthogonalization.

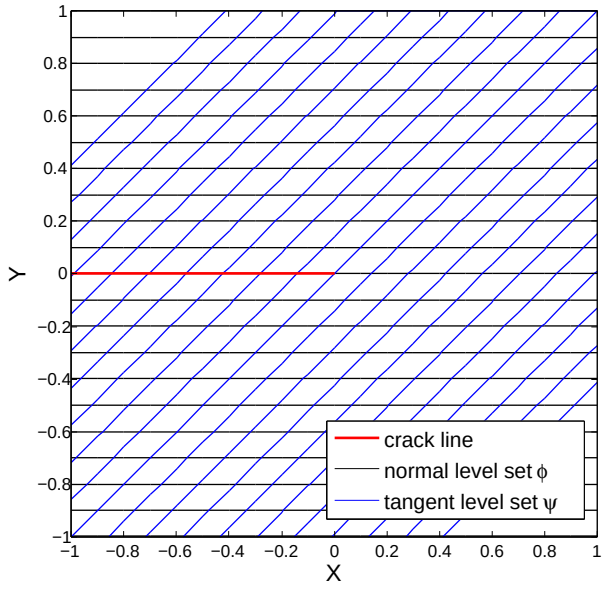


Figure 6.9: Level sets before reorthogonalization.

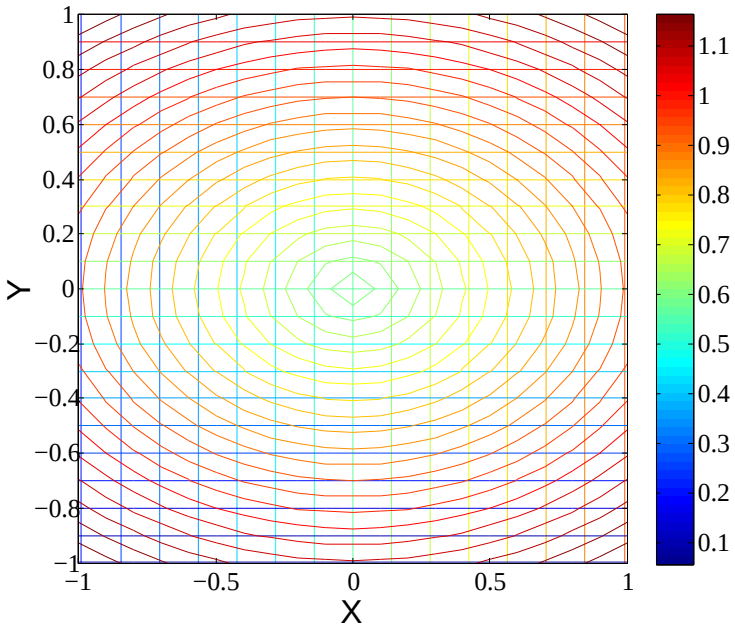


Figure 6.10: Level sets after reorthogonalization and before reinitialization.

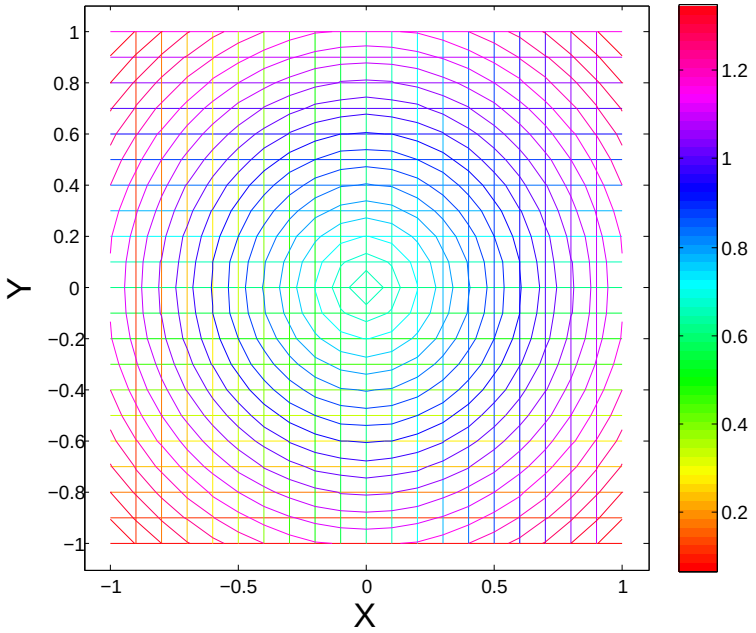


Figure 6.11: Level sets after reorthogonalization and reinitialization.

6.6.3 Penny-shaped crack initialization

The penny-shaped crack is a typical 3D crack surface and is a close description of material flaws found naturally. It is a 3D version of a 2D straight crack rotated through a full circle. The level set description of the geometry of the penny-shaped crack is simple to describe. Suppose a penny-shaped crack with radius r is aligned in the plane of $Z = h$ with the centre point as (x_c, y_c, h). Then the level sets are initialized as

$$\phi(\mathbf{x}) = z - h \tag{6.64}$$

$$\psi(\mathbf{x}) = \sqrt{(x - x_c)^2 + (y - y_c)^2} - r \tag{6.65}$$

In this example, we set $r = 0.3$ and the centre point of the penny-shape crack is located at $(0, 0, 1)$. The level set domain is $[0, 2] \times [0, 2] \times [0, 2]$ and the structured grid points are set at 0.1 intervals in all dimensions. The zero isosurfaces of the two level sets are plotted in Figure 6.13 and the intersection of the isosurfaces is the geometry of the penny-shaped crack as shown in 6.14. Two different levels of $\psi(\mathbf{x}) = 0$ and $\psi(\mathbf{x}) = 0.1$ are shown in Figure 6.15. To study the accuracy of the level sets description, a set of discretized points belonging to the intersection of the two level sets of the crack surface are sampled as shown in Figure 6.12. The relative

error norm e_{lsc} are calculated Equation (6.61) is about 0.3%. The unit outer normal along the crack front, which is required in the J integral, is listed in Table 6.3. It can be seen from the table that the LSC are at least to second digit accurate after the decimal point and some are up to third or fourth digit accurate. The relative error norm using Equation (6.61) is 0.25%.

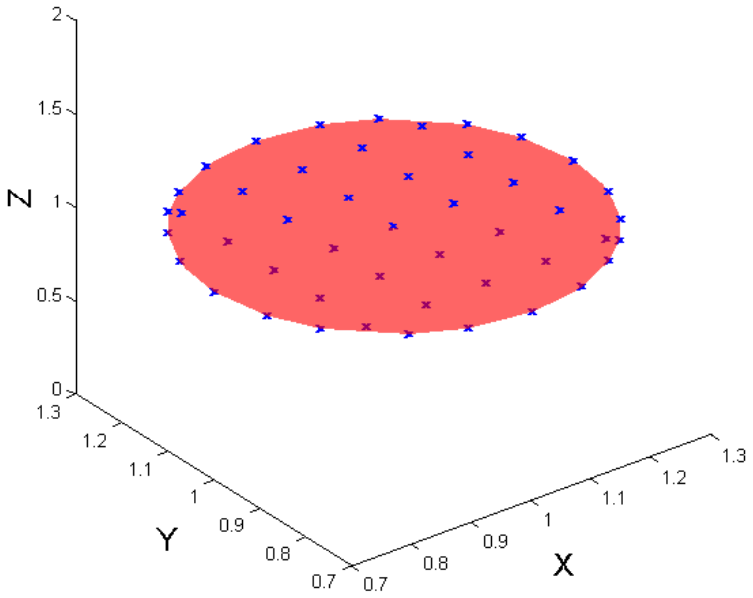


Figure 6.12: Points lying on the crack surface given by the level sets for calculating the relative error.

6.6.4 Lens-shaped crack initialization

Now we look at how to initialize the level sets for a lens-shaped crack. Understanding the geometry of the lens-shaped crack is a first step of initialization. Consider a lens in 3D as shown in Figure 6.16, which can be regarded as a portion of a spherical surface cut off by a plane. It is also known as the spherical cap in geometry. The portion can be measured by the half central angle θ shown in Figure 6.16. If $\theta = \frac{\pi}{2}$ then the lens becomes a semi-spherical surface, and the extreme case is $\theta = \pi$ and lens becomes a sphere which is a closed surface. In Cartesian coordinates, the

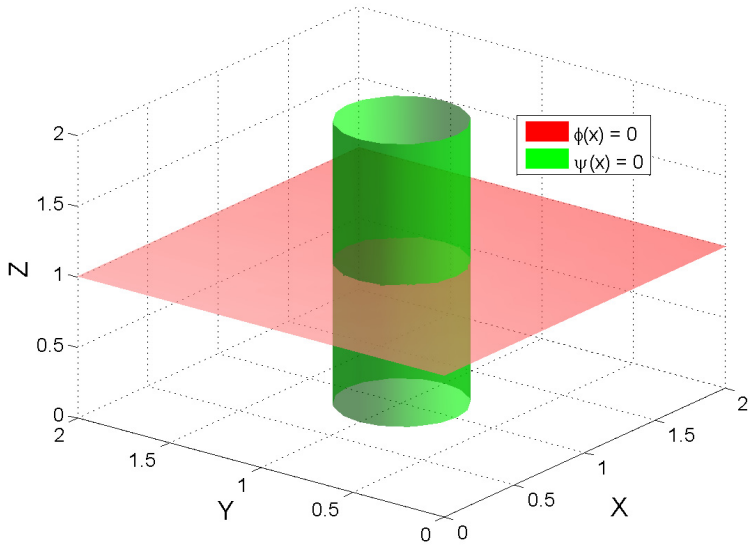


Figure 6.13: Plot of zero level sets of a penny-shaped crack.

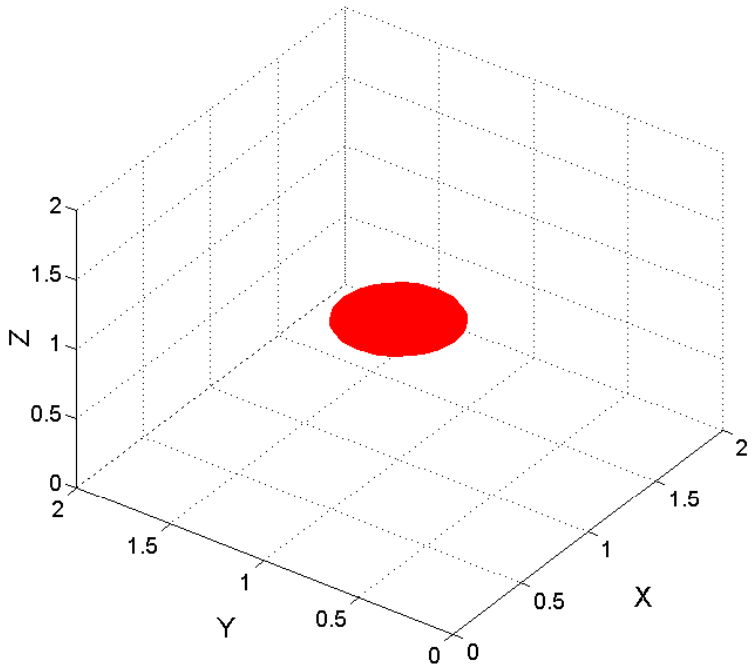


Figure 6.14: The penny-shaped crack described by the intersection of the two level sets.

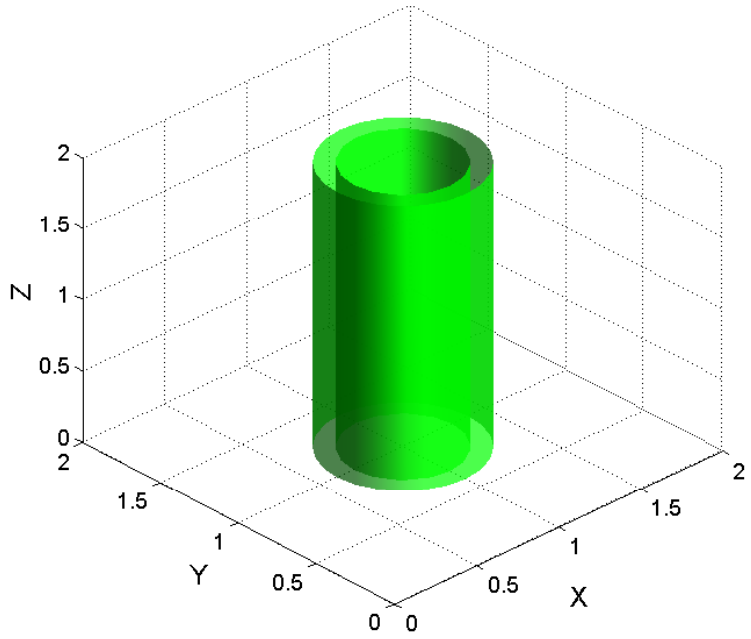


Figure 6.15: Plot of the isosurfaces $\psi = 0$, and 0.1 (from inner to outer) of the penny-shaped crack.

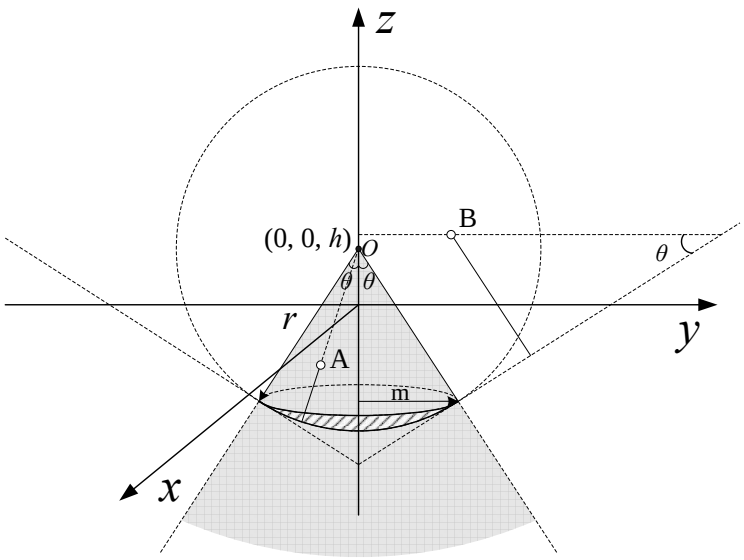


Figure 6.16: A 3D lens-shaped crack.

Exact			LSC		
n_x	n_y	n_z	n_x	n_y	n_z
-0.83133	-0.55578	0.00000	-0.83303	-0.55323	0.00000
-0.98704	-0.16046	0.00000	-0.98738	-0.15834	0.00000
-0.94849	0.31681	0.00000	-0.94871	0.31614	0.00000
-0.83133	0.55578	0.00000	-0.83303	0.55323	0.00000
-0.55578	-0.83133	0.00000	-0.55323	-0.83303	0.00000
-0.55578	0.83133	0.00000	-0.55323	0.83303	0.00000
-0.16046	-0.98704	0.00000	-0.15834	-0.98738	0.00000
-0.10681	0.99428	0.00000	-0.10418	0.99456	0.00000
0.31681	-0.94849	0.00000	0.31614	-0.94871	0.00000
0.31681	0.94849	0.00000	0.31614	0.94871	0.00000
0.55578	-0.83133	0.00000	0.55323	-0.83303	0.00000
0.55578	0.83133	0.00000	0.55323	0.83303	0.00000
0.83133	-0.55578	0.00000	0.83303	-0.55323	0.00000
0.99428	-0.10681	0.00000	0.99456	-0.10418	0.00000
0.94849	0.31681	0.00000	0.94871	0.31614	0.00000
0.83133	0.55578	0.00000	0.83303	0.55323	0.00000

Table 6.3: Comparison of the unit outer normal of along the crack front given by the LSC and the exact solution for the penny-shaped crack.

geometry of a lens can be parameterized as

$$\begin{cases} x^2 + y^2 + (z-h)^2 = r^2 \\ z \leq r(1-\cos\theta) \end{cases} \tag{6.66}$$

where $\{x, y, z\}^T$ is the coordinate vector of a point on the surface, r is the radius of the sphere from which the surface is a part and h is the z coordinate of the centre as shown in Figure 6.16. The lens is a bounded surface while the level sets should extend beyond the lens-shaped surface to the entire domain. Thus we will need to define the level set values inside and outside the lens respectively. If a point falls inside the cone formed by the spherical cap, shown as the shaded area in Figure 6.16

(for example A), the distance normal to the lens is simply the the radius r minus the distance from A to O. While for a point outside the cone (for example B) the distance required is from B to its projection on the extension of the lens surface. The extension of the lens is the tangential plane to the rim of the lens. To this end, it is clear that the 3D space around the crack needs to be divided into two parts. The part inside the cone is described by points for which

$$f(x,y,z) = \begin{cases} x^2+y^2 & \leq \quad (h-z)\tan\alpha \\ z \leq h & \end{cases} \tag{6.67}$$

where $f(x,y,z)$ is a parametric function describing the geometry of the cone. Through geometric analysis the normal level set $\phi(\mathbf{x})$ is given as follows:

$$\phi(\mathbf{x}) = \begin{cases} \left((z-m)\cot\theta - \sqrt{x^2+y^2}\right)\sin\theta - r\sin\theta & \text{Outer part} \\ \sqrt{x^2+y^2+(z-h)^2} - r & \text{Inner part} \end{cases} \tag{6.68}$$

where $m = (h-z)\tan\alpha$ is the radius of the osculating circle, i.e. the circle comprising the lens front. It is difficult to give an analytical expression for $\psi(\mathbf{x})$ since the lens has curvature along the crack surface unlike the penny-shaped crack. We assign the initial values of ψ_0 as

$$\psi_0 = \sqrt{(x^2+y^2)} - r\sin\theta \tag{6.69}$$

and then reorthogonalize ψ with respect to ϕ using the procedure described in §6.3.5. An algorithm implementing the initialization of the lens-shaped crack is given in Appendix B.1. In this example, we use $r = 0.4$ with the centre point of the sphere that the lens belongs to located at $(0,0,0.3)$. The level set domain is $[-1,1]\times[-1,1]\times[-1,1]$ and the structured grid points are set at 0.1 intervals in all dimensions. In Figure 6.17 the isosurfaces of ψ are plotted for various levels. It can be seen each isosurface of ψ is an open conical surface however the top of the cone is not a point but is connect to a slant tube. This is due to the curvature of the surface that make it impossible to have ψ perfectly orthogonal to ϕ when it is close to the centre of the lens. The osculating circle of the front of the most outer isosurface $\psi = 0.2$ exceeds the size of the level set domain and is therefore is shown jagged along the front in this figure.

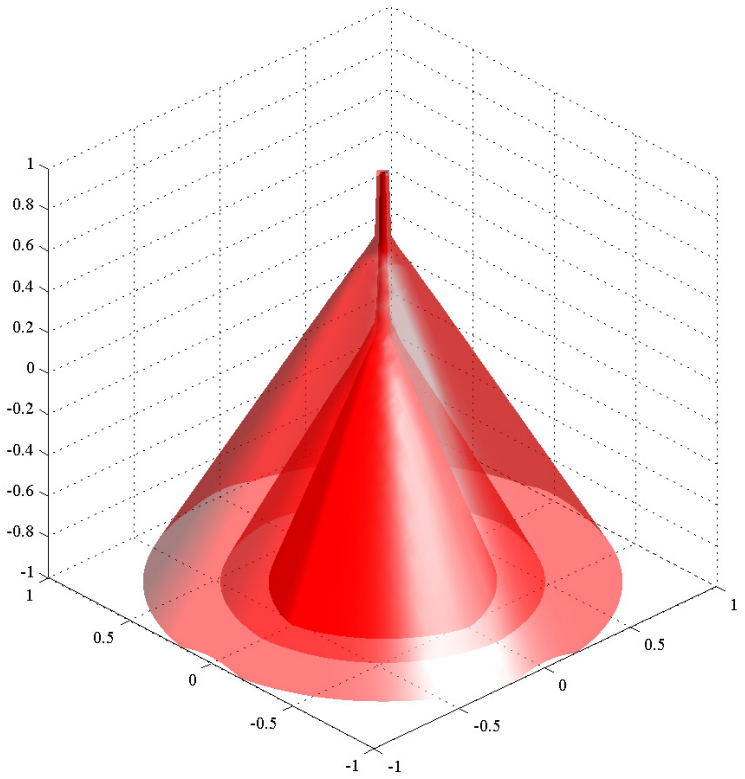


Figure 6.17: Plot of the isosurfaces $\psi = -0.05$, $\psi = 0$, and $\psi = 0.2$ (from inner to outer) of the lens-shaped crack.

Figure 6.18: The lens-shaped crack described as the intersection of the two zeros level sets.

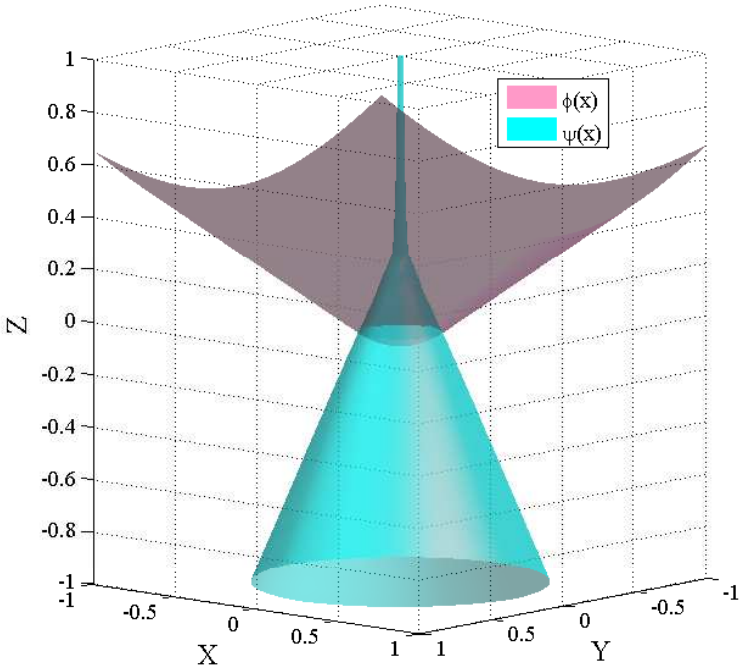


Figure 6.19: Plot of the zero isosurfaces of two level sets.

6.6.5 Elliptic crack initialization

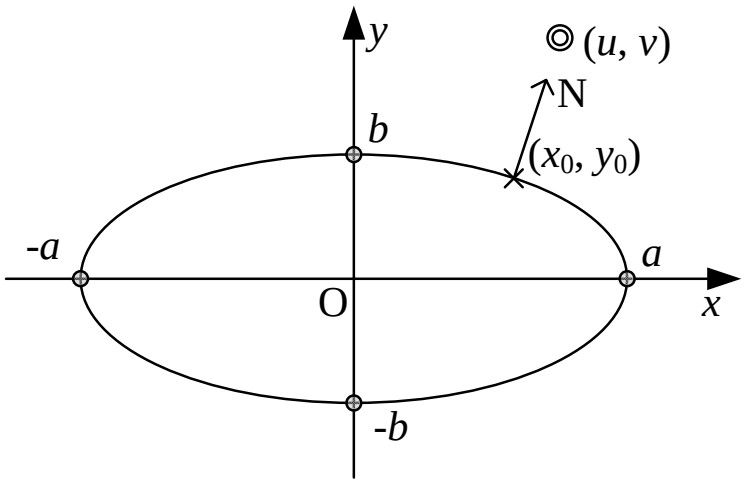


Figure 6.20: The geometry of an ellipse (a and b are respectively the lengths of long axis and short axis).

In this section, we use the level sets to describe an elliptic crack surface. The geometry of an ellipse in 2D can be described by the following equation

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 0 \tag{6.70}$$

where a and b are respectively the half lengths of the long and short axes (assuming $a > b$) of the ellipse as shown in Figure 6.20. An ellipse can be regarded as a circle being "squashed" and the severity of squashing can be measured by the eccentricity defined as

$$e = \sqrt{1 - \frac{b^2}{a^2}} \ . \tag{6.71}$$

Two extreme cases are firstly $a = b$, $e = 0$ and the ellipse is a circle, and secondly $b \to 0$ and $e = 1$ and the ellipse is squashed into a "closed lips" which becomes a straight crack. Interestingly, the variation in the eccentricity e is linked to that of the the stress concentration factor described in §1.3.2.

The ellipse in 2D corresponds to an elliptic crack surface in 3D. Again we will use two level sets to describe the geometry. The initialization of ψ requires finding the shortest distance to an ellipse. Surprisingly, there is no analytical solution for this problem though the geometry of an ellipse is quite simple. Two numerical approaches to this problem are described in [194], namely reducing the problem to solving a rational function or reducing the problem to a quartic polynomial solution. The latter approach, which was originally proposed in [195], is the most widely used at present. The quartic polynomial equation for finding the closest point is

$$P(t) = a^2U^2(t + b^2)^2 + b^2v^2(t + a^2)^2 - (t + a^2)^2(t + b^2)^2 = 0 \tag{6.72}$$

where (u, v) is the coordinates of the point to which we wish to find the shortest distance to the ellipse and t is the unknown parameter to be solved for. By solving for t, the closest point can be found as $(x_0, y_0) = (u, v) - t\mathbf{N}$ where $\mathbf{N}$ is the outer normal at (x_0, y_0) and its components are calculated as $\mathbf{N} = \left\{\frac{x_0}{a^2}, \frac{y_0}{b^2}\right\}$. In this example, we set $a = 0.6$, $b = 0.3$ and the centre point of the ellipse is located at $\{0, 0, 1\}$. The level set domain is $[0, 2] \times [0, 2] \times [0, 2]$. The elliptic surface described by the level sets is plotted in Figure 6.21 which is the intersection of the two level sets shown in Figure 6.22. A number of isosurfaces of ψ are plotted in Figure 6.23. It should be noted that for the elliptic crack the shortest distances obtained as described above do not lead to an accurate signed distance function [184]. Thus reinitialization needs to be applied to ψ. The level set domain is $[0, 2] \times [0, 2] \times [0, 2]$ and the structured grid points are set at 0.1 interval in all dimensions. In Figure

6.24 a number of isosurfaces of ψ are plotted. In Figure 6.23 we compare the ψ before and after reinitialization for $\psi = -0.2, -0.1, 0, 0.1, 0.2, 0.3, 0.4, 0.5$. where it can be seen that they are almost identical except the level inside the zero contour at $\psi = -0.1$. A number of sampling points along the crack front are compared between their level sets as shown in Figure 6.25. The relative error of the crack front is approximately 0.35% calculated by Equation (6.61) .

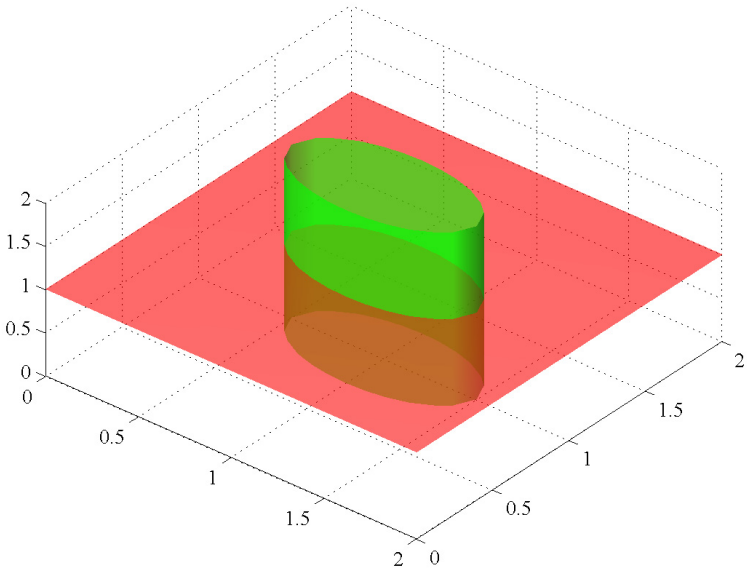


Figure 6.21: Plot of the isosurfaces of zero level sets of elliptic crack.

6.6.6 Visibility criterion for the penny-shaped crack

We use the same example of a penny-shaped crack in §6.6.3 with $r = 0.5$ to test the proposed visibility criterion based on the LSC. The meshless nodal arrangement is arranged at approximate intervals of 0.1 in the three directions of the coordinates. Here we do not perform any stress analysis and just show the use of the LSC in introducing the jump term. Figure 6.26 shows one quarter of the penny-shaped crack with intersection points between a node on one side and a quadrature point on the other. The marks, i.e. circle or asterisk in the figure shows the intersection between crack surface and lines linking a node on one side of the crack surface and an integration point on the other side of the surface. Red circles indicate points given by the proposed new method for the visibility criterion while the black points are those

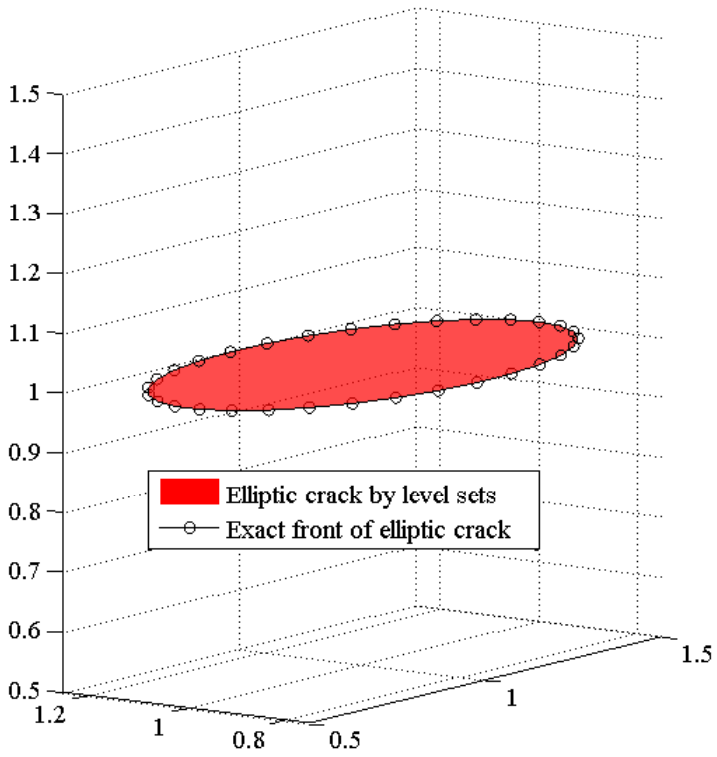


Figure 6.22: The elliptic crack described by the intersection of the two level sets.

from a standard Eulerian method using triangular facets to represent the crack surface as shown in Figure 6.27. Applying the visibility criterion requires determination if the line linking the nodes and the integration point on the other side intersects a triangular facets, the representation of the crack surface. In an Eulerian method this is achieved by solving a set of simultaneous parameter equations consisting of the functions describing the plane containing the triangular facet and the line, and then assessing if the intersecting point falls inside or outside the triangular facet. It can be seen from Figure 6.26 that some points along the crack front captured by the present method (indicated by the blue arrows) are missed by the Eulerian method.

6.6.7 Level sets advancement for the penny-shaped crack

Here we demonstrate the level sets advancement procedure described in §6.3.4 with the penny-shaped crack problem. The crack model and level set domain used are the same as in §6.6.3. The initial size of the penny-shaped crack is set at $r = 0.3$ and we apply a virtual constant velocity of $V_\psi = 0.1$ and $V_\phi = 0$ and advance the

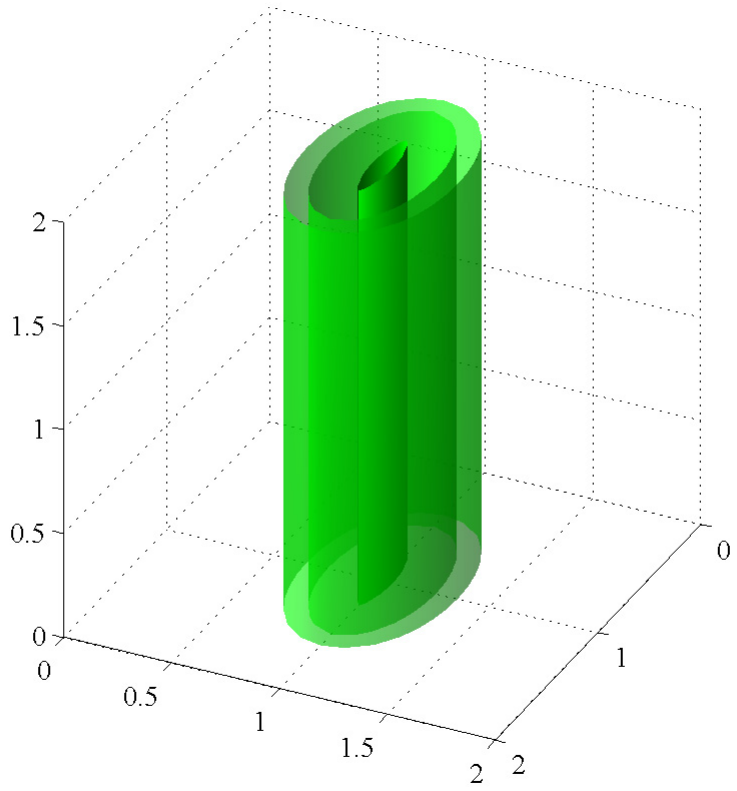


Figure 6.23: Plot of the isosurfaces $\psi = -0.2$, $\psi = 0$, and $\psi = 0.1$ (from inner to outer) of the elliptic crack.

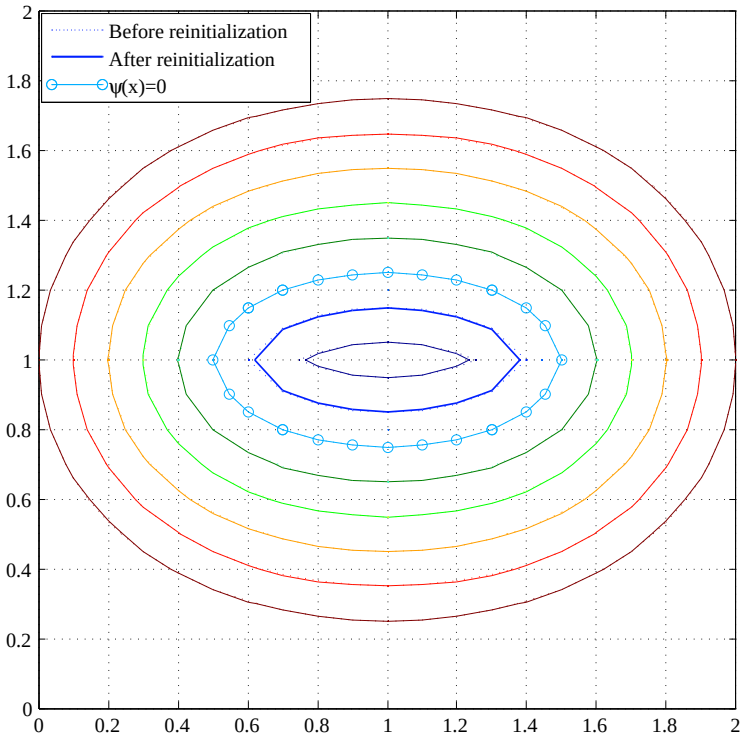


Figure 6.24: Contour slices of $\psi = -0.2,\ -0.1,\ 0,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5.$ (Z view) for the elliptic crack before and after reinitialization.

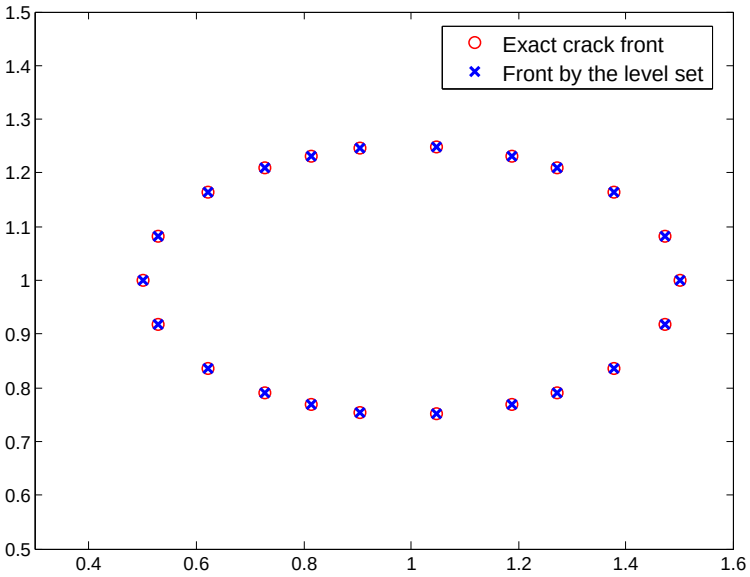


Figure 6.25: Sampling points along the crack front given by the LSC and the exact solution for the elliptic crack problem.

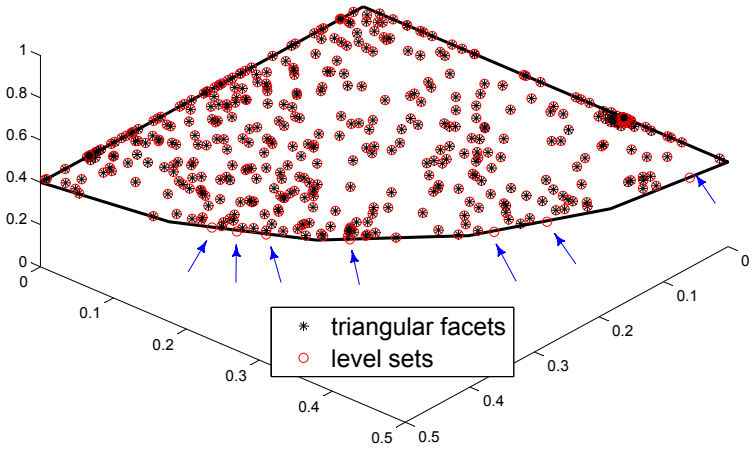


Figure 6.26: Curvature along crack front is missed by the piecewise triangular facets and is captured by the level sets method.

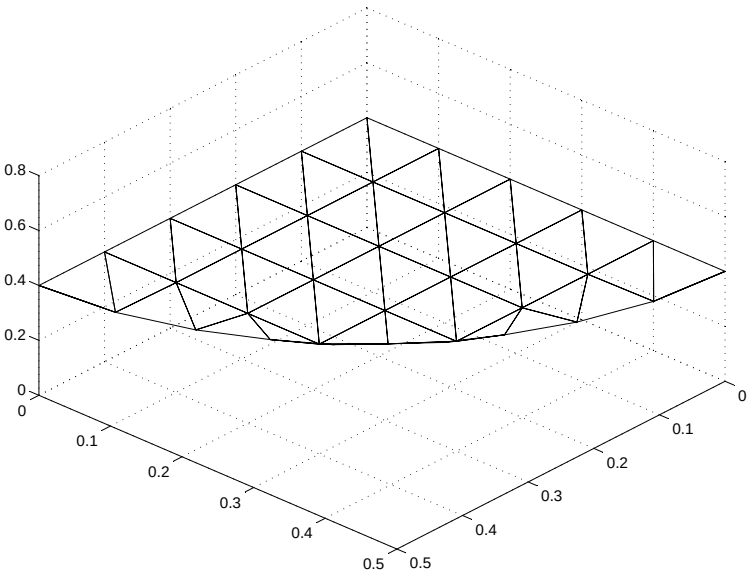


Figure 6.27: The triangular facets used in Eulerian method for the visibility criterion.

level sets for six unit time steps. Since the velocity along the normal level set is zero thus the penny-shaped crack will grow uniformly in all directions, i.e. the crack surface maintain a perfect circular shape and increases in radius of 0.1 at each step of advancement. The crack surfaces given by the level sets at t from 1.0 to 6.0 are shown in Figure 6.29. In these figures the crack surfaces are seen to remain circular surfaces. However if we plot the crack fronts given by the LSC and compare them with the exact solution in Figure 6.28 (where the dashed lines shows the crack front given by the level sets and the solid line shows where crack front is expected to be located), it can be seen that the crack fronts gradually deviate from the correct fronts with an increase in t. In Table 6.4, the relative error e_{lsc} with respect to the expected positions is listed. After five steps of advancement, the error becomes about 1.3%.

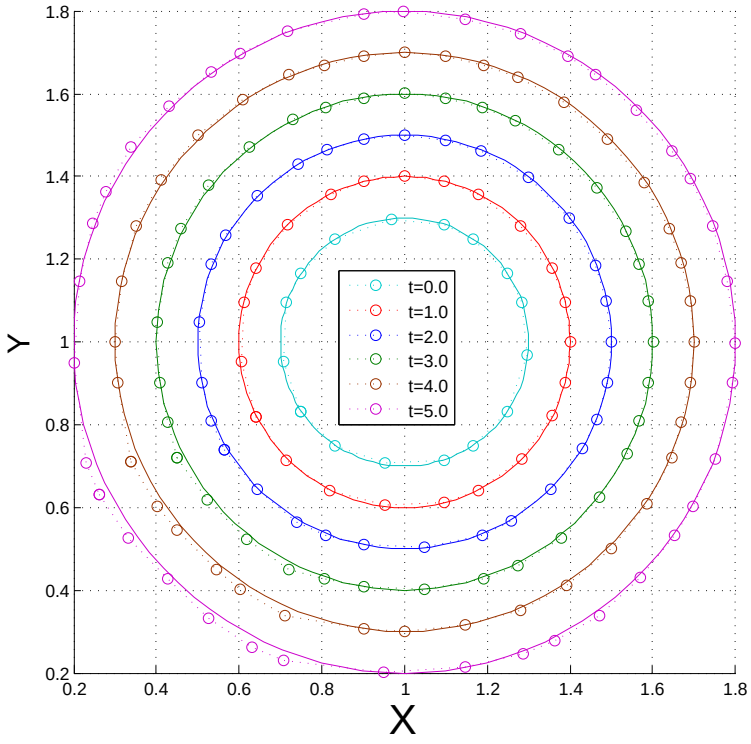
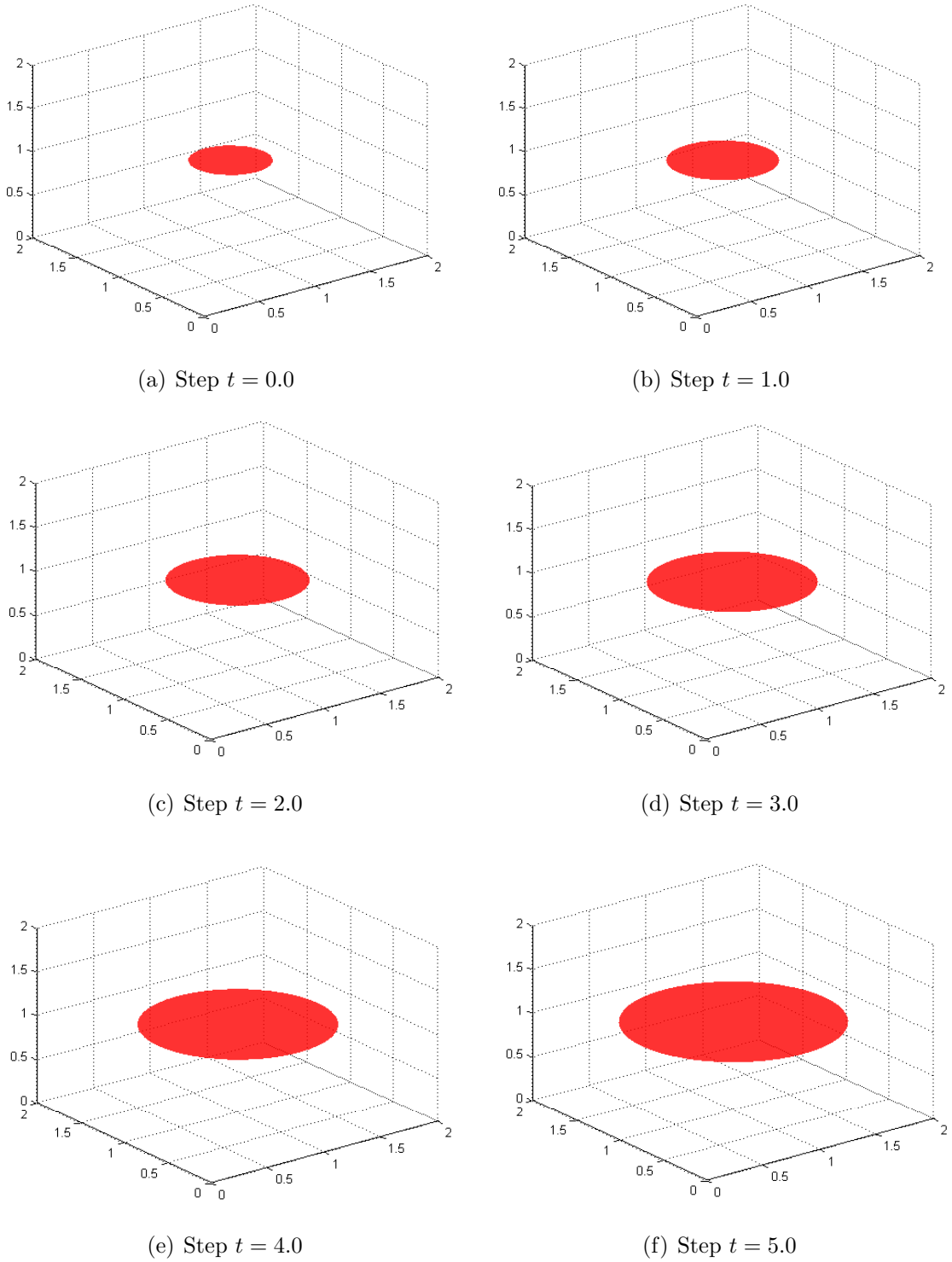


Figure 6.28: The crack fronts given by level sets after each step of advancement against the expected crack fronts.



(a) Step $t = 0.0$

(b) Step $t = 1.0$

(c) Step $t = 2.0$

(d) Step $t = 3.0$

(e) Step $t = 4.0$

(f) Step $t = 5.0$

Figure 6.29: Level sets advancing of the penny-shaped crack at different time steps.

Unit time step t	1.0	2.0	3.0	4.0	5.0
Relative error $e_{lsc}(\%)$	0.33	0.58	0.97	1.10	1.28

Table 6.4: The error e_{lsc} in level sets description or the penny-shaped crack after each step of advancement.

6.7 Discussion

This chapter has described the use of level sets in fracture modelling with the EFGM, showing them to be of wide utility in the overall system for fracture modelling including the post-processing steps. The difference from the coupling of the level sets with the XFEM has been shown and jump terms based on level sets are proposed for both the visibility criterion and the diffraction method used in the EFGM. The level sets initialization, reinitialization, reorthogonalization and advancing procedures are described and some issues in the level sets reinitialization and advancing are discussed. A number of 2D and 3D crack problems with varying geometries are presented using the procedures described to address one or several issues in the level set representation of the crack surfaces. They show the feasibility of using the level sets in 3D crack modelling and as stated above the visibility criterion is adopted from here onwards. On problem, however remains in the use of this method instead of the diffraction method and that is discussed and solved in the next chapter.

Chapter 7

A tying procedure for accurate crack modelling using the visibility criterion

7.1 Introduction

The visibility criterion for introducing displacement jump in the EFGM is simple to implement and efficient to compute, especially with the help of level set coordinates. However, spurious discontinuities have been reported around crack tips using the visibility criterion, while implementing the diffraction method in 3D is much more complicated than the visibility criterion. In this chapter, a tying procedure is proposed to remove the difficulty with the visibility criterion so that crack tip closure can be ensured while the advantages of the visibility criterion can be preserved. The formulation is based on the use of level set coordinates and the element free Galerkin method, and is generally applicable for single or multiple crack problems. The chapter explains the formulation and provides verification of the method against a number of 2D crack problems.

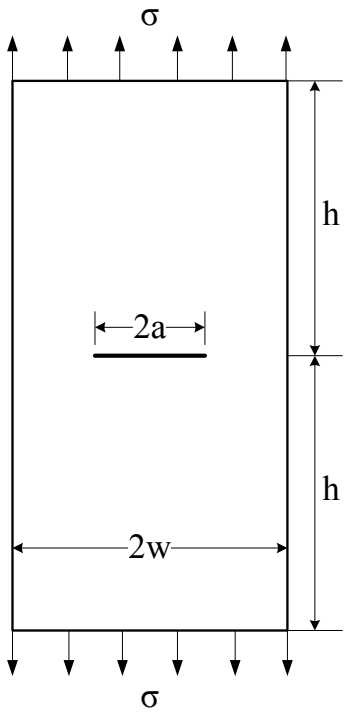
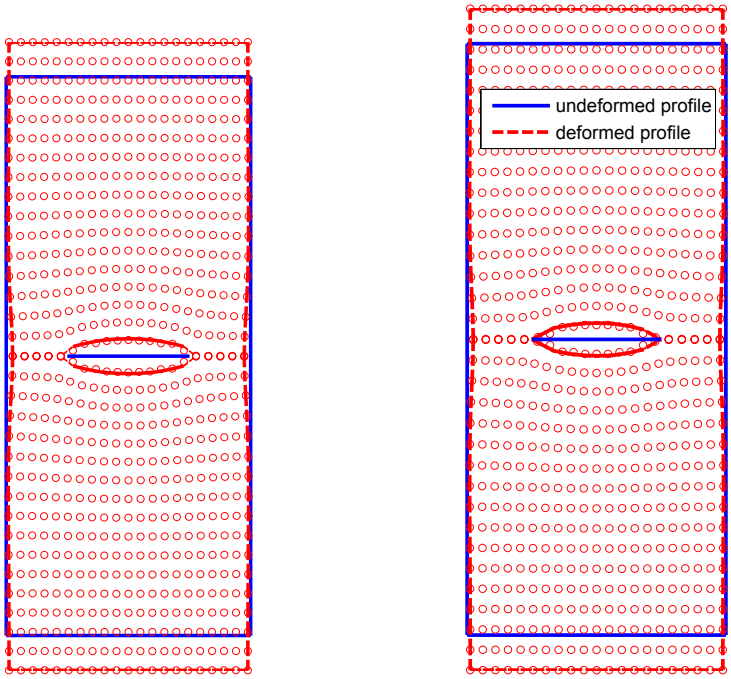


Figure 7.1: Geometry of the centre crack problem.

7.2 Dilemma between visibility and diffraction

It has been reported in [154] that significant errors occur when using the visibility criterion to introduce the displacement jump. The errors are due to spurious crack extension ahead of the crack tip so that the displacements and stresses calculated do not correspond to the original crack tip position, leading to errors in the SIFs. This is now demonstrated by modelling a centre crack in an infinite plate using the standard visibility criterion. A finite plate is modelled as shown in Figure 7.1 where the dimensions of the plate are width $w = 7$, height $h = 16$ and crack length $2a = 3.5$ (i.e. $h/w > 2.285$). (It is shown in [196] that the dimensional effects of using a finite plate can be omitted when $h/w > 1.5$ so that for this problem the SIF is only dependent on a and w). A unit tensile stress $\sigma = 1$ is applied at both ends of the plate and the modelling uses the EFGM with 714 nodes. For this pure mode I problem (i.e. $K_{II} = 0$) the reference value for the normalized mode I stress intensity factor is $F_I = 1.1867$. Therefore the exact SIF should be $K_I = F_I\sqrt{\pi a} = 2.7825$ however the K_I given using the EFGM with the visibility criterion is 2.970, an error of 7%. Nodal displacements are plotted in Figure 7.2 showing that the crack



(a) spurious crack extension by visibility (b) crack tip remain closed by diffraction

Figure 7.2: Centre crack example: comparisons of deformations using the visibility criterion and the diffraction method.

tips spuriously extend ahead of the defined locations. The SIF cannot therefore be correctly calculated as r and θ are not measured correctly. If the diffraction method is used, however, the crack tip remains closed and there is no problem of spurious extension. K_I obtained is then 2.6889, which is an error of 3%, satisfactorily accurate without enrichment. Both SIFs (either by visibility or diffraction) are required with respect to the intended crack tip location. The purpose of this test is to show that while SIFs can be obtained using the visibility criterion the results are useless if there is spurious tip extension. It is clear then that the diffraction method fixes the problem met with the visibility criterion, but brings raised computational complexity if moving to 3D or multiple cracks, or both. The spurious crack extension problem is associated the approximation in the EFGM as shown in the Figure 5.1(c). In the EFGM, the discontinuity along the crack geometry is approximated and is unlike the case in the FEM where the crack geometry is exactly represented as shown in Figure 5.1(a). Take node I for example as shown in Figure 7.3. The nodal support of I is shown by the truncated grey circle. It can be seen that node I will be used in the

approximation within the shaded area, part of which is ahead of the crack tip. Thus the dashed line, which is the extension of the crack, will be influenced by u_I since $\phi_I(\mathbf{x}) \neq 0$. This clearly shows the difficulty of defining an exact boundary to stop crack extension. Zi *et al.* [105] fixed this problem by modifying nodal support (shown as dashed circles in Figure 5.1(c)) so that the nodal support is either completely cut by the crack or does not intersect the crack. This method excludes the nodes which are partially cut by the crack in the approximation at the crack tip. The idea is similar to that used in the XFEM [197] where an element edge is aligned with the crack tip to stop spurious crack extension inside the element. However, in the XFEM the shape functions take nodal values so that crack tip can be exactly closed when it is aligned with element edge. While in the EFGM, shape functions generated by the MLS approximation lack the delta property. Thus a rigorous proof of crack tip closure using the modified nodal support requires further examination. Another question is if one needs to recover the displacement at the crack tip then there will be no nodes in support according to the modified nodal supports. It is clearly shown

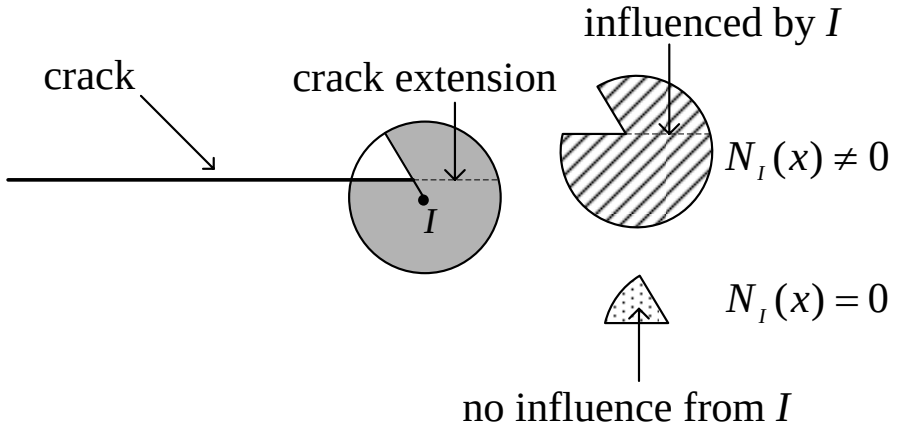


Figure 7.3: An illustration of spurious extensions ahead of the crack tip due to the difficulty of blending approximations in the EFGM.

in the above that the diffraction method fixes the problem met with the visibility criterion, but brings raised computational complexity if moving to multiple cracks or 3D , or both. So, the dilemma is that one would prefer to use the visibility criterion in 3D for economy but cannot guarantee great accuracy due to spurious extension. In the following, a new approach will be developed for fracture modelling

using the EFGM where the visibility criterion can still be used for accurate fracture modelling.

7.3 A new approach avoiding spurious crack extension using ties

In this section we propose a modifications to the standard EFGM for crack modelling as described so far which involves "tying" the crack tip to prevent spurious crack extension. The procedure produce accurate solutions using the (cheap) visibility criterion, a removal of the spurious crack extension problem described in the previous section and a robust means of modelling and tracking crack fronts and faces in 3D. It is simple and cheap to implement.

The idea is illustrated in Figure 7.4 for an edge crack problem in 2D and is based on the tying elements in [198] originally developed for the FEM. In practice no elements are created as such (thus maintaining the meshless nature of the modelling). The only additional computational cost required is to calculate a small additional stiffness matrix and to assemble it into the global stiffness matrix. It will be shown later that in the context of meshless methods the additional cost is equivalent to the addition of two integration points for each crack tip, which is trivial with respect to the total number of integration points. The tying element is a line type spring with stiffness β located at the crack tip connecting a pair of nodes located very close to the crack tip but on either side. We assume no contact between the upper and lower crack surfaces and the crack surfaces remain traction free, thus the work done along the surface due to the contact is zero without any tying element. The total potential energy of the system is given as

$$\Pi = \Pi_s - W_f. \tag{7.1}$$

where Π is the total potential energy, Π_s is the strain energy and W_f is the external work as were used in Equation 2.50. As the crack tip should not extend spuriously (this being an elastic analysis) and this is not ensured by the EFGM plus visibility alone, the total potential energy of the system should include a tying term and

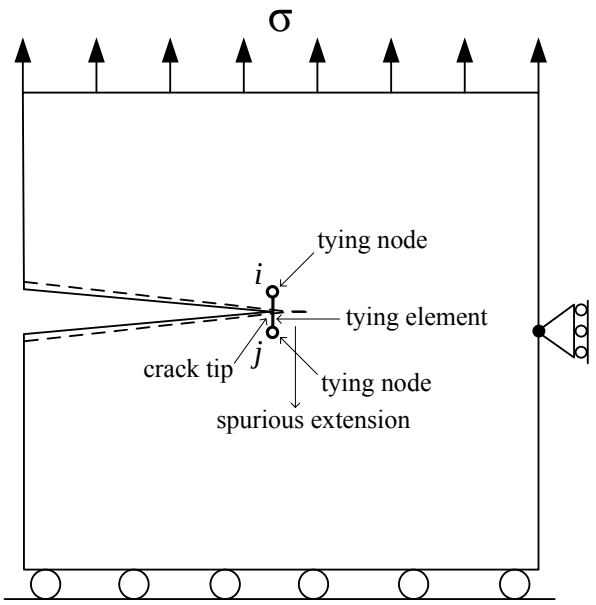


Figure 7.4: The tying scheme applied to a 2D edge crack problem.

should be expressed as

$$\Pi = \Pi_s - W_f + W_{\text{tying}} \tag{7.2}$$

where W_{tying} is calculated from

$$W_{\text{tying}} = \frac{1}{2}\beta(\mathbf{u}^i - \mathbf{u}^j) \cdot (\mathbf{u}^i - \mathbf{u}^j). \tag{7.3}$$

and $\mathbf{u}^i$ and $\mathbf{u}^j$ are displacements at the upper and lower nodes of the tying element as shown in Figure 7.4. From the variation

$$\delta\Pi = \delta\Pi_{\text{int}} - \delta W_{\text{ext}} + \delta W_{\text{tying}} = 0. \tag{7.4}$$

discrete equations leading to stiffness matrix terms can be determined in the standard way. In Equation (7.4) the final term is found by applying a variation to the nodal unknowns $\mathbf{u}^i$ and $\mathbf{u}^j$ in Equation (7.1), to obtain

$$\delta W_{\text{tying}} = \beta \left[(\delta\mathbf{u}^i)^T\mathbf{u}^i + (\delta\mathbf{u}^j)^T\mathbf{u}^j - (\delta\mathbf{u}^i)^T\mathbf{u}^j - (\delta\mathbf{u}^j)^T\mathbf{u}^i \right]. \tag{7.5}$$

It should be noted for implementation that the tying element should be offset a small value along the crack surface from the crack tip, as shown in Figure 7.5. The shaded areas Ω_i and Ω_j are the domains of definition of points coincident with nodes i and j respectively. Here, the "domain of definition" is defined at a point of interest

and it is the union of all the nodal supports of the nodes in support of the point as shown in Figure 7.5. Ω_c denotes the union of these domains truncated with the presence of the crack and represents the connection between the tying nodes. If the tying nodes are located exactly at the crack tip then the tying procedure can fail, or at least become unstable, due to the change in the region Ω_c which is then affected by the proximity of the crack tip to the line connecting the tying nodes.

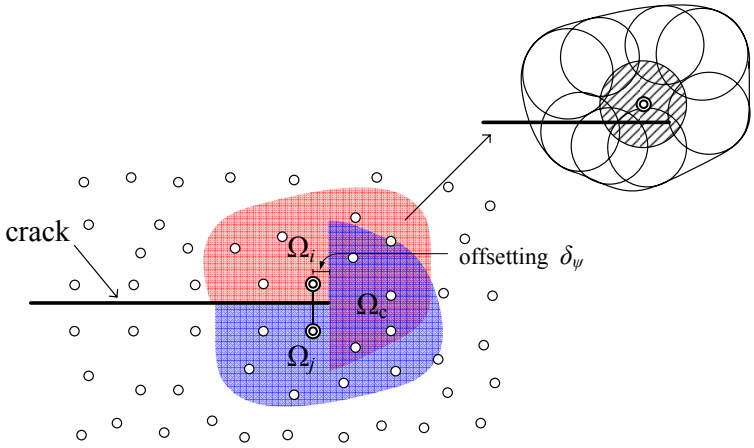


Figure 7.5: Nodes offset inward of a small value δ_ψ with respect to the crack tip.

7.4 Implementation issues

In this section we discuss the implementation issues concerning the discretisation of the proposed weak form and the addition of jump nodes along either side of a crack based on level sets.

7.4.1 Discretisation of tying procedure

In Equation (7.5), $\mathbf{u}^i$ and $\mathbf{u}^j$ are the approximations at $\mathbf{x}_i$ and $\mathbf{x}_j$ instead of "fictitious nodal values" worked with in MLS-based meshless methods such as the EFGM. Therefore when the discretized form is derived, we substitute for $\mathbf{u}^i$ and $\mathbf{u}^j$ the MLS approximations

$$\mathbf{u}^i(\mathbf{x}_i) = \sum_I^{n_i} N_I \mathbf{u}_I^i \qquad \mathbf{u}^j(\mathbf{x}_j) = \sum_J^{n_j} N_J \mathbf{u}_J^j \tag{7.6}$$

where I and J are the indices of nodes in support of $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively, and n_i and n_j are the total number of nodes in support of $\mathbf{x}_i$ and $\mathbf{x}_j$, also respectively. In Equation (7.4), $\delta\Pi_{\text{int}}$ and δW_{ext} are the normal terms in the weak form for an elastic problem. The parameter β is a large value constant representing the stiffness of the tying element and here we take $\beta = 1000 \times E$ where E is the Young's modulus of the material. Substituting Equation (7.6) into (7.5) we get the discretised form as

$$\begin{aligned}\delta W_{\text{tying}} = \beta\delta\mathbf{u}_I^i \sum_I \sum_J N_I N_J \mathbf{u}_J^i + \beta\delta\mathbf{u}_I^j \sum_I \sum_J N_I N_J \mathbf{u}_J^j \\ - \beta\delta\mathbf{u}_I^i \sum_I \sum_J N_I N_J \mathbf{u}_J^j - \beta\delta\mathbf{u}_I^j \sum_I \sum_J N_I N_J \mathbf{u}_J^i \ .\end{aligned} \tag{7.7}$$

Equation (7.7) is then substituted into to Equation (7.4) to obtain the global stiffness matrix and no additional unknowns introduced by the tying procedure. $\mathbf{u}_I^i$ and other similar terms appearing in Equation (7.7) will cancel out and the rest of the implementation follows that detailed in Chapter 2.

7.4.2 Nodes aligned with the crack

Improved accuracy can be obtained from the modified method by placing pairs of nodes along a crack face on either side. This agrees with findings elsewhere, e.g. the node splitting method proposed in [152] as well as other proposals [162, 199]. The displacement jump can be better captured by these nodes instead of via extrapolation from nodes distant from the crack face. Generation of these node pairs may appear like an onerous extra task in preprocessing however the LSC enable easy placement of these along a crack, and also serve to aid updating in the case of crack propagation (not dealt with here). In 2D this may not appear particularly more advantageous than using line segments to describe the crack but, once again, the benefits are more evident in 3D. The generation procedure for these points using the LSC is as follows:

1. Find the sets of LSC grid points $N_\phi^+ = \{\mathbf{x}|\phi(\mathbf{x}) = 0.01L_K\}$ and $N_\phi^- = \{\mathbf{x}|\phi(\mathbf{x}) = -0.01L_K\}$

2. Find a set of LSC grid points that $N_\psi = \{\mathbf{x}|\psi(\mathbf{x}) \leq \epsilon_\psi\}$, ϵ_ψ is a small value and here $\epsilon_\psi = 0.001L_K$ where L_K is the length of the crack.

3. Find the intersection of the sets $N_K^+ = N_\phi^+ \cap N_\psi$ and $N_K^- = N_\phi^- \cap N_\psi$.

4. Insert points within the set N_K^+ and N_K^- respectively at even spacings L_K/n_k, where n_k is the number of the pairs of nodes aligned with the crack. If the node overlaps or gets too close (within a specified tolerance) to an existing node, then the aligned node is not added.

An example is given in §7.5 to demonstrate the benefit these additional nodes bring in terms of accuracy.

7.5 Numerical tests

In this section, we test the proposed tying procedure with a number of 2D crack problems. The purpose of this section is to demonstrate that the procedure works satisfactorily.

As has been previously pointed out in [119, 121], a large number of integration points are required in the EFGM to avoid the underestimation of the global weak form. It is also suggested that in 2D the number of integration points should be at least $9 - 10$ times the number of nodes to ensure acceptable accuracy. If this requirement is not met we found in our tests that the SIF is oscillatory for varying size of integral path. The J integral used to calculate SIF should be path independent, i.e. the J value is not dependent on the shape and size of path and hence SIF. In all the examples, the integration for the weak form is performed over triangular background integration cells, where all nodes and background cells are generated using `gmsh` [142]. A structured nodal arrangement is used throughout for simplicity, and in some examples the background cells are refined near the crack. The additional tying and jump nodes near the crack are arranged according to §7.4.2. The J integral is calculated along a circular path around the crack tip which is divided into 40 smaller arc segments in which a four-point Gaussian integration rule is used.

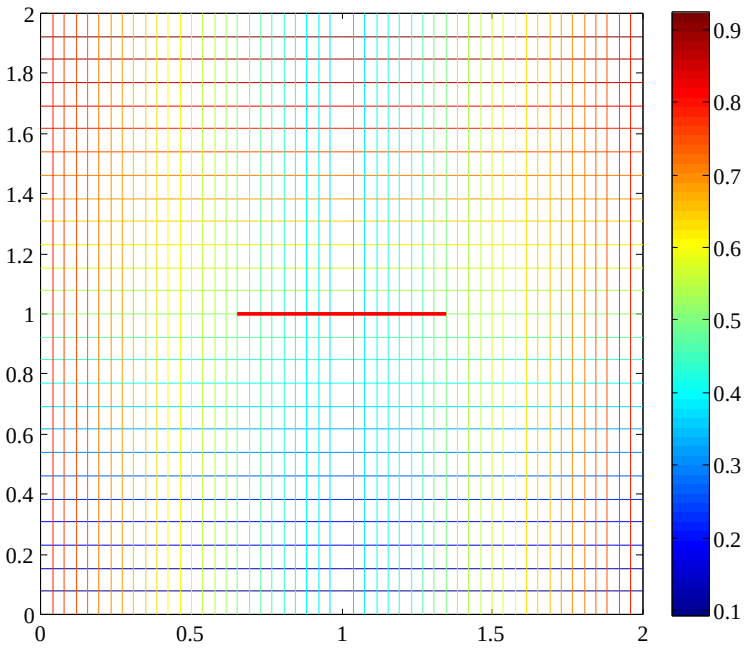


Figure 7.6: Contour plots of ϕ and ψ of the centre crack.

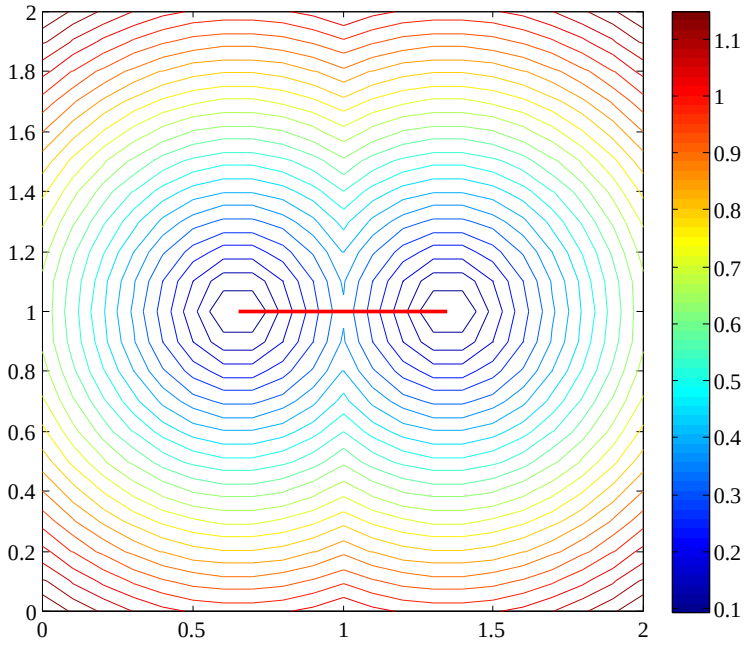


Figure 7.7: Contour plot of r of the centre crack.

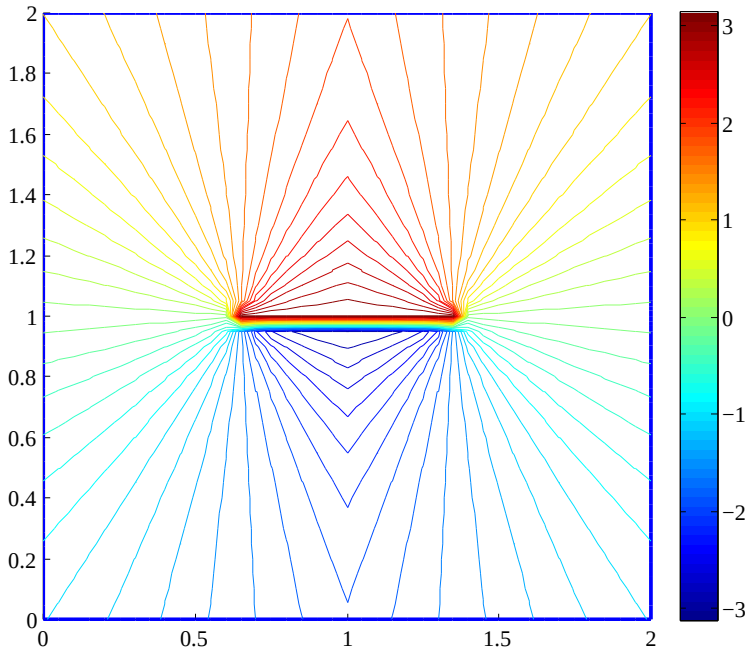


Figure 7.8: Contour plot of θ of the centre crack.

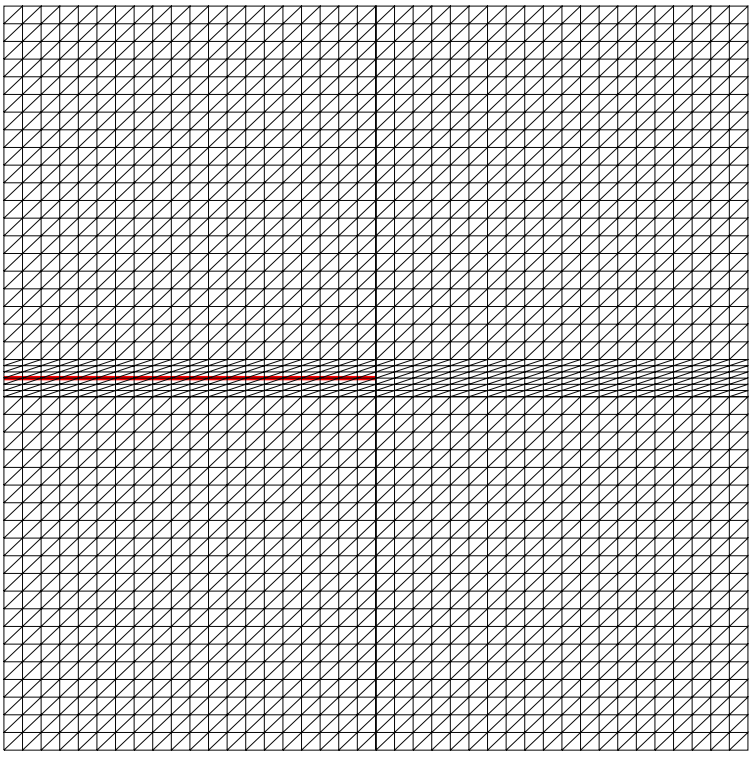


Figure 7.9: Background cells used for the centre crack problem.

7.5.1 Centre crack in a square plate

Predictions for SIFs for a finite square plate with a centre crack is the first example as shown in Figure 7.1. The dimensions of the plate are $w = h = 1$. Uniform tractions $\sigma = 1$ are applied at both ends of the plate. A structured nodal arrangement (21×21 nodes) is used in the test. The contour plot of the level sets representation for this problem is shown in 7.6. The distances r to the crack tips according to Equation (6.59) and the local angle θ according to Equation (6.58) are plotted in Figure 7.8 and 7.7 respectively. The triangular background mesh used for integration is shown in Figure 7.9 comprising 3680 triangles. The integration cells are refined around the crack tips to improve accuracy. Material parameters, Young's Modulus $E = 1e5$ and Poisson's ratio $\nu = 0.25$, are used. The half length of the crack a is changed in successive analyses from 0.1 to 0.7 at 0.1 increments. As this is a symmetric problem, the two crack tips have the same SIFs. In this example, we do not apply any criterion for crack propagation, and the purpose is to test the proposed method for varying crack lengths. We keep the nodal arrangement the same while jump nodes and tying nodes are added as described in §7.4.2 for the varying sizes of crack. In all tests with varying crack length there is no spurious discontinuity ahead of the crack by the present method. To demonstrate this point, we plot the crack opening profile for $a = 0.4$ as an example in Figure 7.10 in which the crack tip is located at $(0.4, 1.0)$. Plots are shown of vertical displacements along and ahead of the crack tip using the visibility criterion with and without tying. The difference between upper and lower face displacements is plotted as δU_y. It can be seen that spurious crack tip extension is halted using the tying scheme as required. The SIFs obtained by the new procedure are compared with those using the diffraction method in Table 7.1 and the reference solutions, and SIFs with enrichment are compared in Table 7.2. When the crack size is very small $a = 0.1$, poorer agreement is found as compared to the results with a large crack $a \geq 0.3$. This is because we keep the uniform nodal arrangement and integration cells for all sizes of crack length. The accuracy could be improved through an adaptive analysis [4, 41] by gradually refining the integration cells and inserting new nodes. However it is not within the scope of the present thesis. It can be seen from the Tables that when the two crack tips approach the

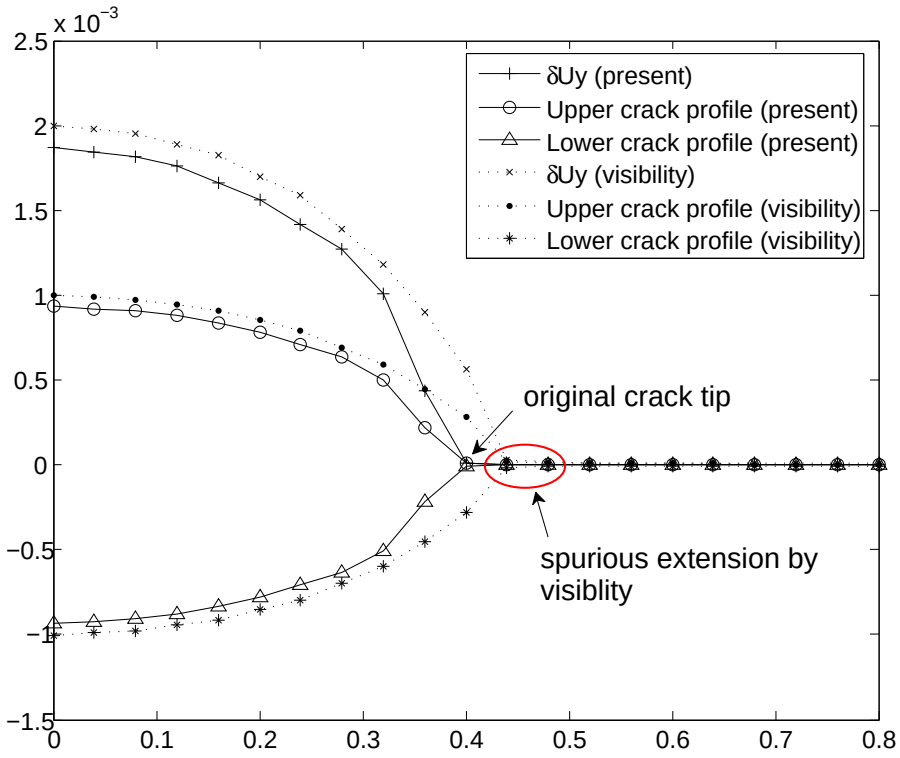


Figure 7.10: Crack opening profiles for the centre crack problem ($a = 0.4$) showing removal of spurious extension using the tying scheme. (without enrichement)

plate edges, for $a \geq 0.5$, the boundary effects also lead to a reduction in accuracy. The present method is comparable to the diffraction method and good agreement is obtained with the reference solution with enrichment.

7.5.2 Slanted edge crack

The second example is a single slanted edge crack as shown in Figure 7.11. The dimensions used in the test are $w = 7$, $L_1 = 7$, $L_2 = 10.5$, $a = \frac{1}{\sqrt{2}}w$ and $\theta = 45°$. A structured nodal arrangement (15×31 nodes) is used as shown in Figure 7.12. The background cell arrangement comprises 840 triangles formed by partitioning each of the 420 rectangles formed by the nodes. To initialize the level sets for slanted crack, the only difference from the centre crack example is that the local coordinate system is rotated from the global coordinate system one. Thus we can firstly map the coordinates of a grid point from global to local and then the rest are the same as in §6.6.1. The mapping can be generalized for 2D or 3D if we use the

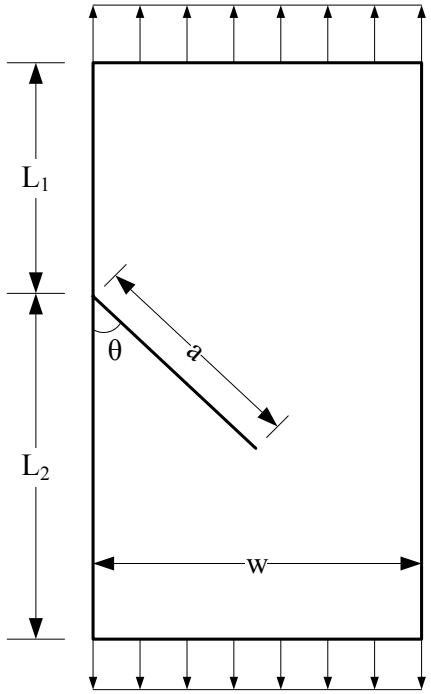


Figure 7.11: A slanted edge crack in a rectangle plate.

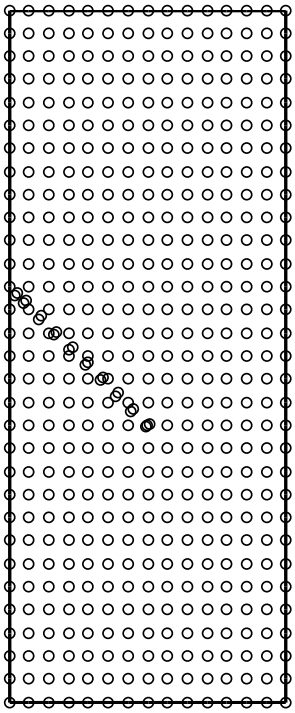


Figure 7.12: The nodal arrangement used for the slanted edge crack problem.

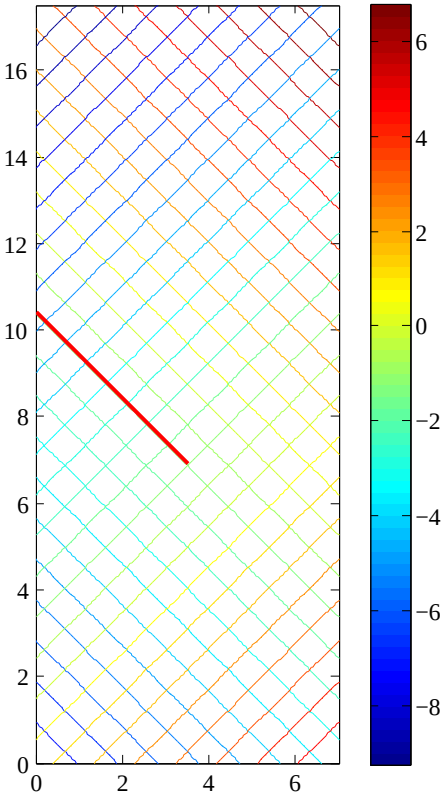


Figure 7.13: Contour plots of ϕ and ψ of the slanted edge crack.

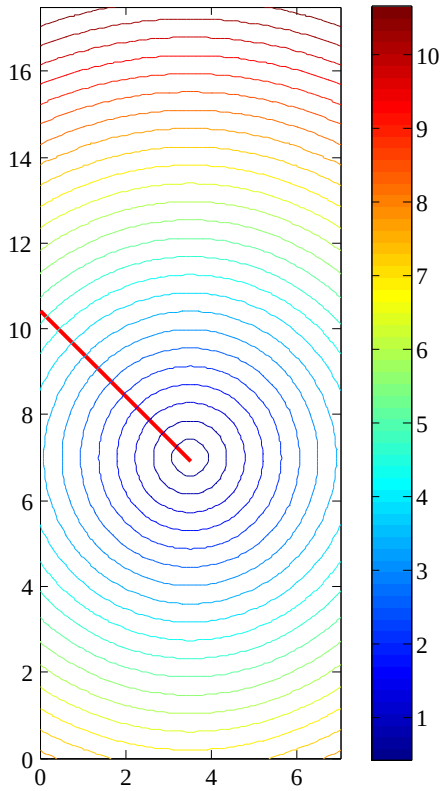


Figure 7.14: Contour plot of r of the slanted edge crack.

a	K_{I} Exact	Tying element		Diffraction	
		K_{I}	Error(%)	K_{I}	Error(%)
0.1	0.568	0.547	-3.773	0.535	-5.850
0.2	0.836	0.809	-3.308	0.806	-3.631
0.3	1.090	1.051	-3.589	1.056	-3.130
0.4	1.363	1.306	-4.221	1.318	-3.282
0.5	1.672	1.604	-4.080	1.615	-3.416
0.6	2.033	1.949	-4.162	1.960	-3.611
0.7	2.491	2.371	-4.830	2.322	-6.817

Table 7.1: SIFs for the centre crack problem (without enrichment).

curvilinear coordinate system. The two level sets obtained are plotted in Figure 7.13 and the contour plots of r and θ are shown in Figures 7.14 and 7.15. For this mixed mode problem, we do not refine the background cells as in the centre crack problem and consequently accuracy is reduced. The SIFs given by the present method are compared with those given by the reference solution in Table 7.3.

7.5.3 Slanted centre crack

In this problem, the dimensions of the plate are $h = 18$, $w = 9$ and the crack length is 1.8, as shown in Figure 7.19. The inclined angle θ of the crack is varied from 0° to 90° at 15° intervals. Nine jump nodes are added along the crack on either side and are offset by 1% of the crack length. There are two ties added at each crack tip as shown in Figure 7.19. The SIF results obtained are plotted in Figure 7.20 compared against the analytical solution [200] showing the validity of the procedure. In all tests with varying inclined angle there is no spurious discontinuity ahead of the crack by the present method. It should be noted the tests are performed without enrichment thus the SIFs are slightly beneath the exact answers, which is reasonable. We also observe that with more nodes added along the crack, $\mathrm{K_I}$ converges monotonically while $\mathrm{K_{II}}$ has oscillations in convergence as shown in Table 7.4. The position of the tying elements and all the nodes aligned with crack can be easily determined

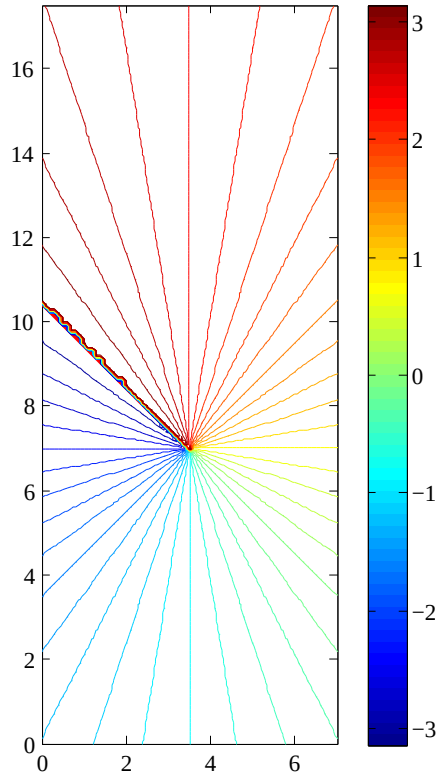


Figure 7.15: Contour plot of θ of the slanted edge crack.

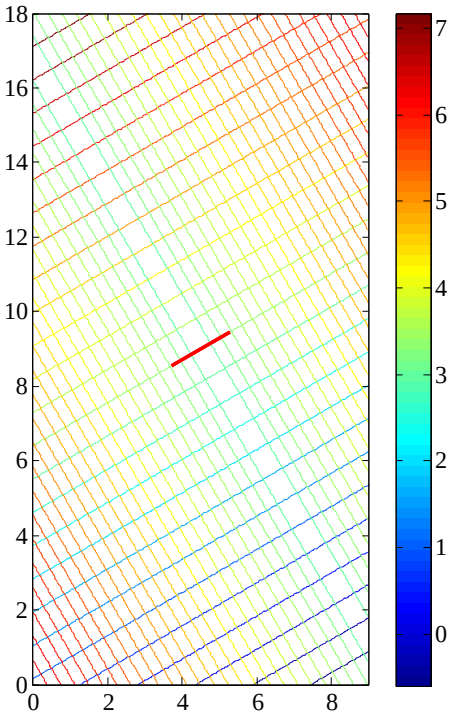


Figure 7.16: Contour plots of ϕ and ψ of the slanted centre crack.

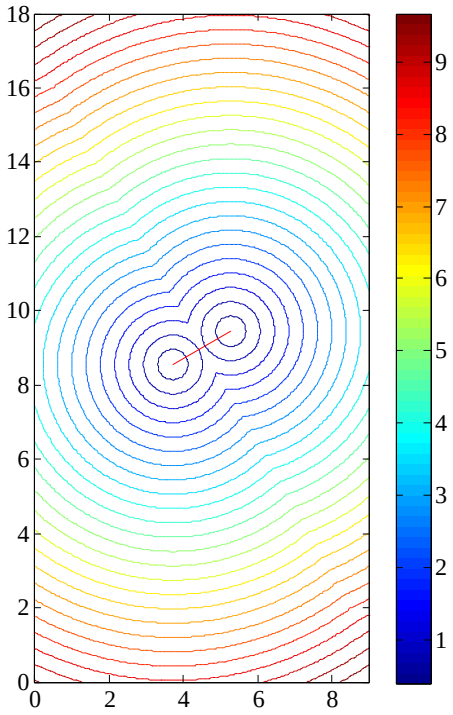


Figure 7.17: Contour plot of r of the slanted centre crack.

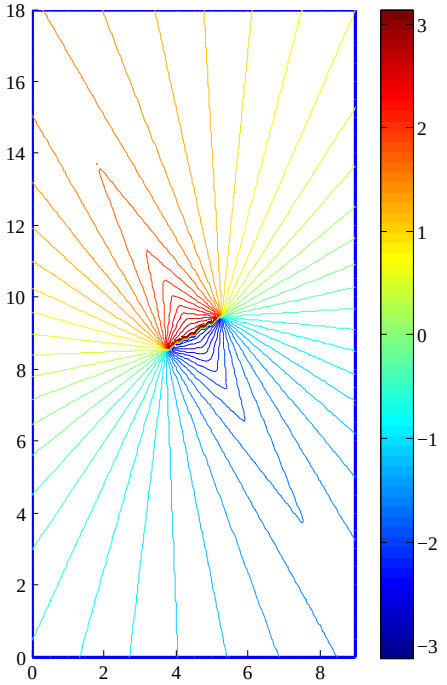


Figure 7.18: Contour plot of θ of the slanted centre crack.

a	K_{I} Exact	Tying element		Diffraction	
		K_I Enriched	Error(%)	K_{I} Enriched	Error(%)
0.1	0.568	0.568	-0.114	0.562	-1.169
0.2	0.836	0.834	-0.235	0.831	-0.689
0.3	1.090	1.087	-0.314	1.088	-0.213
0.4	1.363	1.355	-0.575	1.373	0.694
0.5	1.672	1.661	-0.683	1.687	0.890
0.6	2.033	2.025	-0.434	2.046	0.643
0.7	2.491	2.481	-0.419	2.519	1.110

Table 7.2: SIFs for the centre crack problem (with enrichment).

	K_{I}	Error (%)	K_{II}	Error (%)
Tying element	7.010	-4.235	-3.152	-4.182
Diffraction	6.952	-5.030	-3.135	-4.705
Collocation	7.32	–	-3.29	–

Table 7.3: SIFs given by different methods for the slanted edge crack problem.

with the help of the level set coordinates. In Figure 7.16, the normal level sets ϕ and ψ are plotted as two sets of orthogonal parallel lines and in Figure 7.17 and 7.18 the contour plot of r and θ are plotted. The nodes aligned along the crack are located at $\phi = \pm 0.01 \times 1.8$ with ψ at 1.8/9 intervals, and the tying element $\phi = \pm 0.01 \times 1.8, \psi = -0.001 \times 1.8$. In Table 7.6 and 7.5 shows the changes in the SIFs by varying the size of the J integral path. It can be concluded that the sensitivity of the SIFs to the size of integral path is very weak.

7.5.4 Branched crack in a finite plate

The final test presented here is a more challenging problem, namely a branched crack in a finite plate, as shown in Figure 7.22. The purpose of this test is to show the performance of the present method in dealing with a multiple crack problem. The dimensions of the plate are $h = 5$, $w = 5$, $\theta = 45^\circ$ and the crack lengths are

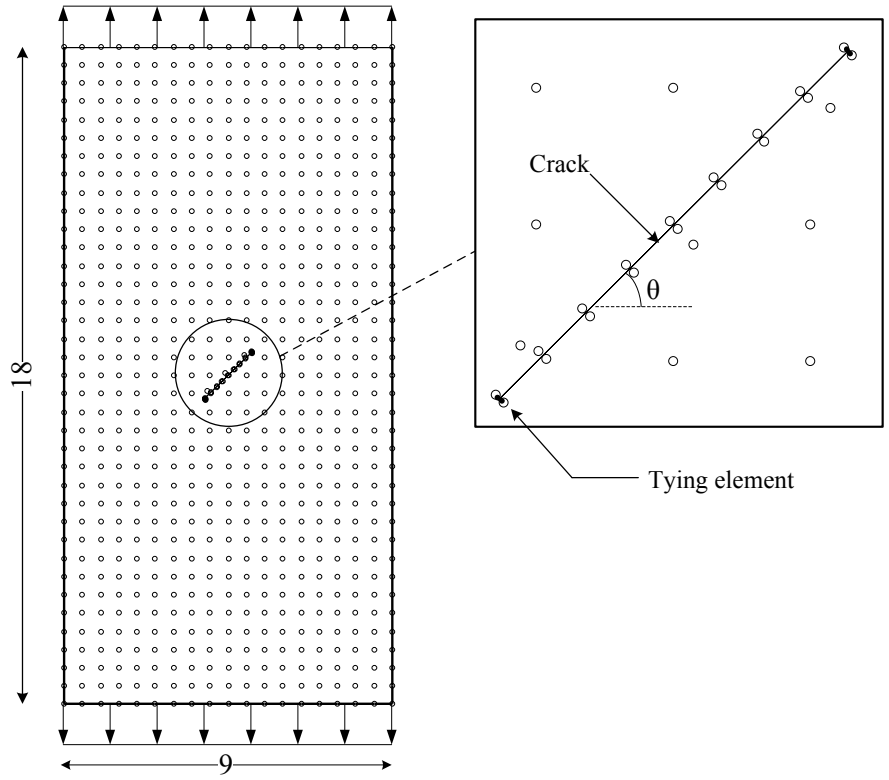


Figure 7.19: Details of the tying "element" for the slanted centre crack problem.

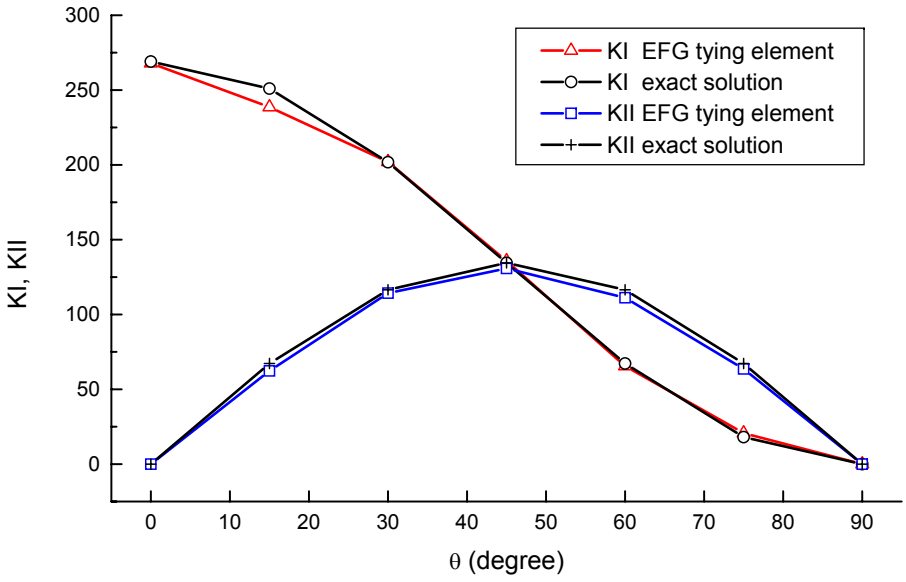


Figure 7.20: SIFs for the slanted centre crack problem.

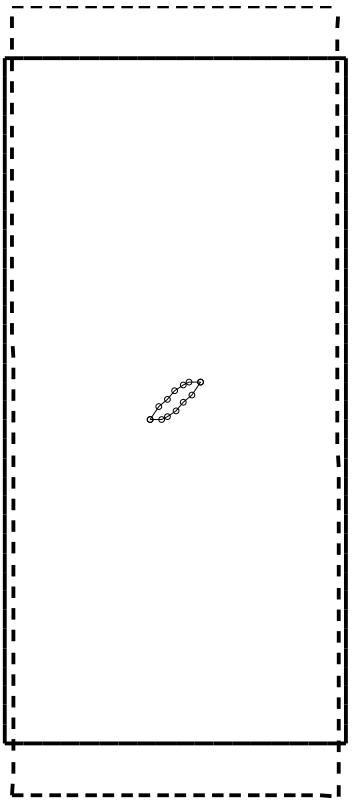


Figure 7.21: Initial and deformed profile of the slanted centre crack problem.

Pairs of nodes	3	4	5	6	7	8	9	Exact
mean K_{I}	191.869	193.269	195.260	198.170	201.346	201.691	201.992	201.780
error-K_{I} (%)	-4.912	-4.218	-3.231	-1.789	-0.215	-0.044	0.105	–
mean K_{II}	108.527	107.970	108.542	110.040	109.217	113.450	114.275	116.498
error-K_{II} (%)	-6.842	-7.320	-6.829	-5.543	-6.250	-2.616	-1.908	–

Table 7.4: Convergence of SIF with number of nodes aligned with crack for the slanted centre crack problem. (D/2a=0.4 and $\theta = 30°$).

D/2a	0.4	0.5	0.6	0.7	0.8	Exact
K_{I} (a)	201.985	204.265	202.978	202.789	200.607	201.780
K_{I} (b)	201.398	204.184	203.068	202.757	200.979	201.780
F_{I} (a)	0.751	0.759	0.754	0.754	0.756	0.750
F_{I} (b)	0.749	0.759	0.755	0.754	0.757	0.750

Table 7.5: Variation of mode I SIFs by changing the sizes of integral path for the slanted centre crack problem ($\theta = 30°$).

D/2a	0.4	0.5	0.6	0.7	0.8	Exact
K_{II} (a)	114.132	118.744	117.620	117.197	116.002	116.498
K_{II} (b)	114.416	118.746	118.115	117.441	115.490	116.498
F_{II} (a)	0.424	0.441	0.437	0.436	0.431	0.433
F_{II} (b)	0.425	0.441	0.439	0.437	0.429	0.433

Table 7.6: Variation of mode II SIFs by changing the sizes of integral path for the slanted centre crack problem ($\theta = 30°$).

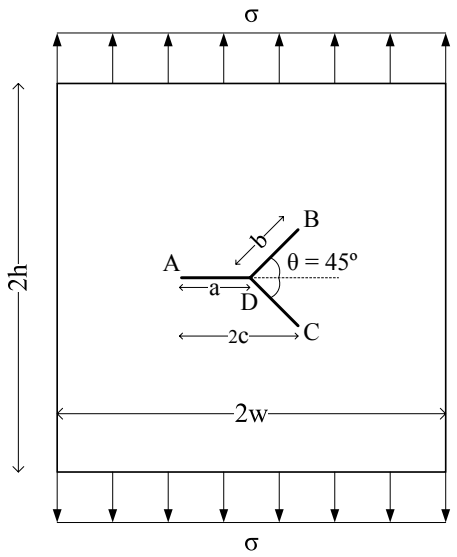


Figure 7.22: A symmetric branched crack in a finite plate under tension.

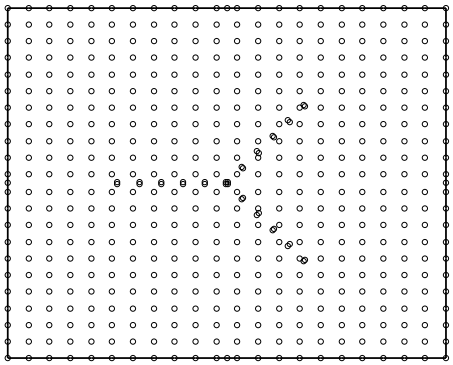


Figure 7.23: The nodal arrangement used for the branched crack problem.

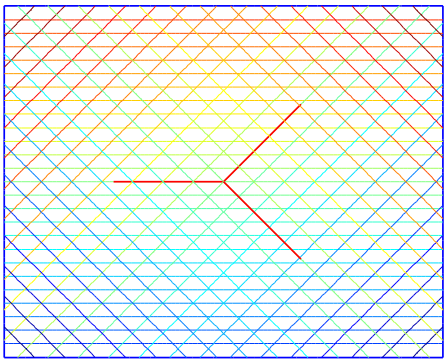


Figure 7.24: Contour plot of ϕ of the branched crack.

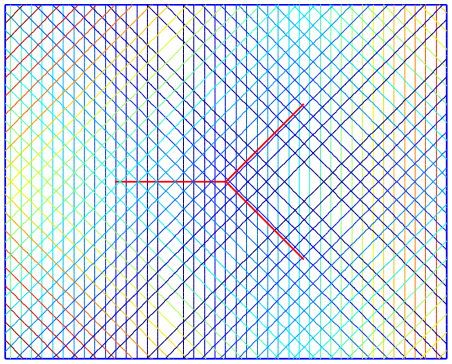


Figure 7.25: Contour plot of ψ of the branched crack.

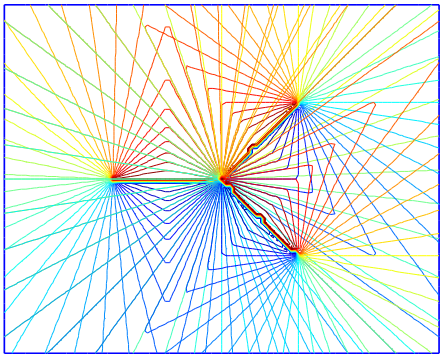


Figure 7.26: Contour plot of θ of the branched crack.

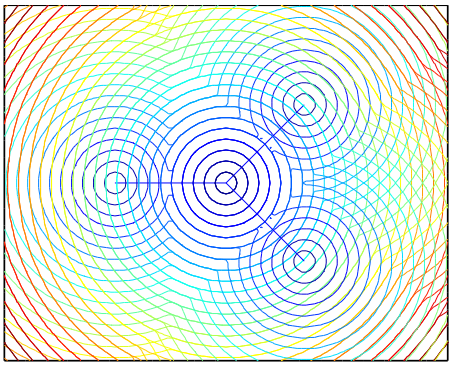


Figure 7.27: Contour plot of R of the branched crack.

$a = b = 2.5$. The plate is subjected to uniaxial traction $\sigma = 1$ applied at both ends. This example has been tested in the XFEM [157] for varying ratio of a/w but here we only test it for a certain crack length. The crack is treated as three cracks using two level sets for each of the crack as shown by Figures 7.24 to 7.27. Figure 7.23 shows the structured nodal arrangement (20×20 nodes) used for the test with 6 pairs of jump nodes for each crack and 4 additional nodes on the centre of the four edges to stop rigid body motion in x and y. As a/w tends to zero, a finer nodal arrangement and integration cell arrangement is needed to obtain results of acceptable accuracy, as in [157], however this is not our main focus here. The SIFs given by the present method are for $\mathrm{K_I^A} = 3.398$, $\mathrm{K_I^B} = 1.564$ and $\mathrm{K_{II}^B} = 1.765$ where the superscripts A and B indicate crack tips. The normalized SIFs are $\mathrm{F_I^A} = 1.312$, $\mathrm{F_I^B} = 0.604$ and $\mathrm{F_{II}^B} = 0.682$, about 4.07%, 4.75% and 5.72% error respectively, without enrichment which is a good result compared with the XFEM results.

7.6 Discussion

In this chapter a crack tip tying procedure is developed to remove the spurious crack extension found when using the visibility criterion alone in modelling fractures with meshless methods, in particular the EFGM. The ties are imposed at the crack tip and the singularity can be captured using an enriched basis. A minor drawback of the present method is oscillatory behaviour of the stress results very close to the crack tip, which is affected by the value of β. However the SIF results are not sensitive to the choice of β since the J integral is calculated along a path away from the crack tip. It should also be noted here that the closure condition is affected by β. In other words, β plays the role of controlling the crack opening ahead of the crack tip. For β set to zero there is no tying effect while for an infinite β the closure is strictly ensured at the crack tip, however in this case the global stiffness matrix is prone to ill-conditioning and hence should be avoided. For intermediate β values as used here, the crack tip closure is almost ensured. This is reflected in Figure 7.10 where the crack opening becomes almost zero at the crack tip and the residual error, which is negligible compared to the maximum crack opening displacement, vanishes in a small range ahead of the crack tip. The effect of β on the crack tip closure condition can be viewed as similar to the "internal scale length" of a material microstructure (or "atomic distance") noted as ℓ, used in non-local gradient elasticity [201, 202], however there are still major differences between the approaches. In the non-local formulation ℓ is introduced to model the cohesive zone ahead of a crack tip and the formulation leads to a non-singular stress field, while the method proposed here removes difficulties in fracture modelling using the EFGM and the singularity still dominates.

The present method is valid for any MLS-based meshless method and could easily be adapted to others. It can be used in 2D and 3D and avoids the potential difficulties in the use of the diffraction method met with multiple cracks and non-planar crack fronts and faces. The formulation is based on the use of level set coordinates, is simple to implement and can be used for multiple crack problems.

Chapter 8

A new 3D crack modelling framework

8.1 Introduction

In this chapter a new framework coupling the EFGM and LSM for 3D crack modelling is proposed. In this framework, level sets are used to describe and capture the crack geometry as it evolves; the visibility criterion is used to introduce the displacement jump based on level sets and the spurious extension problem caused by the visibility criterion is fixed using a tying procedure. The level sets domain is independent from the meshless nodal arrangement and a structured grid is used to avoid the potential problem of shifting of the original crack. A gradient projection method is proposed for finding the crack front and also for extending the velocity from the crack front to the entire level set domain without the need to solve PDEs. The framework brings together the developments discussed in this thesis and in this chapter additional issues relating to 3D are dealt with, and the overall implementation is presented. The major components of the framework are as follows:

1. Independent level set grid
    (a) Level set spatial derivatives following standard WENO scheme.
    (b) Vectorization of data and algorithms in level set computation.
2. Jump enrichment based on level sets

(a) Visibility criterion based on level sets for arbitrary curved crack.

(b) A tying procedure retaining the original crack front.

3. Gradient projection method

(a) Accurate definition of the crack front.

(b) Extension of front velocity without solving PDEs.

4. Refinement and efficient implementation based on level sets

(a) PU integration refinement based on level set near crack front.

(b) Localized jump enrichment based on level sets.

(c) Jump nodes and tying nodes arrangement based on level sets.

A number of 3D examples including an edge crack, a penny-shaped crack and an elliptic crack subjected to various boundary conditions are tested showing the feasibility and efficiency of this new framework.

8.2 A framework for 3D crack modelling

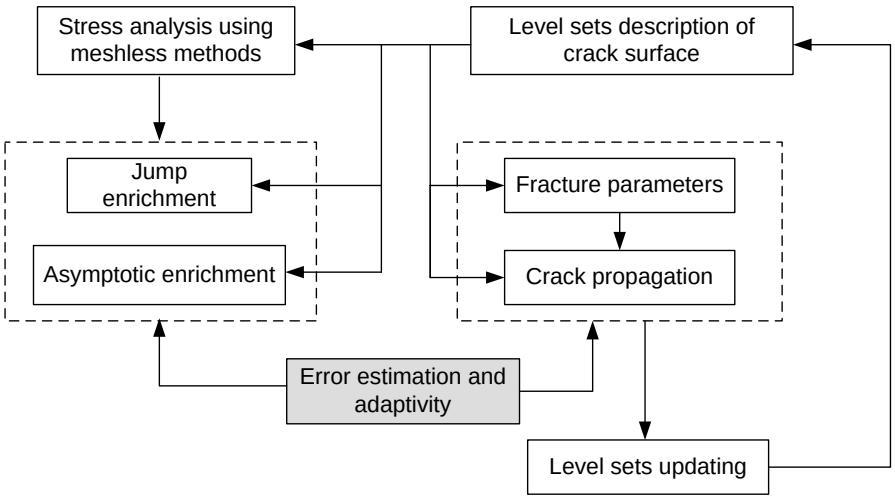


Figure 8.1: A framework coupling the level sets with the EFGM for 3D fracture modelling.

In this section, the framework is explained coupling the EFGM with level sets for fracture modelling. The level sets serve two main purposes: firstly they build the local coordinate system that is needed in the stress analysis as well as in the computation of fracture parameters. Secondly they help with the task of refinement of the meshless nodal arrangement and the integration points near the crack front. As shown in Figure 8.1, the geometry of initial cracks are described by level sets and are used in stress analysis for enrichment terms near the crack surfaces. The stress results are used to obtain fracture parameters to determine the propagation of the crack surface. The propagation of the crack surface is captured by the level sets representing of the new crack geometry. After that, stress analysis is repeated for the new crack along with the level sets updating process. The procedure is repeated until the material fails, i.e. an SIF is recorded greater than the fracture toughness of the material. However in the following test examples, we do not take problems for this far but propagate the crack for several steps to demonstrate the effectiveness of the framework. Before level sets are updated to a new crack geometry, it would be possible for an adaptive analysis using an error estimation process (as has been discussed in Chapter 4) between the stress and the fracture analyses, as shown in the shaded block in Figure 8.1. This process would improve accuracy of crack modelling, however is it is also beyond the scope of this thesis.

8.2.1 Independent level set grid and vectorized computation

Duflot [189] questioned the feasibility of coupling level sets with the EFGM for fracture and pointed out a crack surface might be spuriously shifted from its original position due to the lack of the Kronecker delta property of the EFGM shape functions. The coupling scheme in that paper refers to the same shape functions in the LSM as in the displacement field approximation which leads to difficulties. The following explains this issue and proposes a solution.

The level set values at a point (either a quadrature point or a node) are given by

$$\phi^h(\mathbf{x}) = \sum_I^n N_I(\mathbf{x})\phi_I \quad I = 1, 2, \cdots, n \tag{8.1}$$

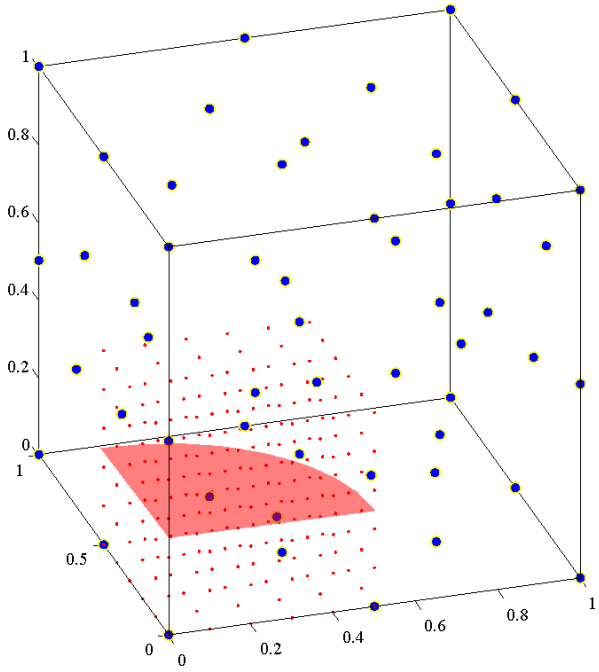


Figure 8.2: The structured level set grid decoupled from the meshless model.

where ϕ_I is the level set value at a node, I is the node index, n is the total number of nodes in support and $N_I(\mathbf{x})$ is the shape function at $\mathbf{x}$. The consequence of lack of a Kronecker delta property is that the zero normal function $\phi = 0$, comprising the crack surface and its extension tangent to the crack front, is not retained and can lead to a shift of the original crack location. This problem does not occur in the XFEM as its shape functions possess the Kronecker delta property. To remove this difficulty with the EFGM, it is proposed that the grid used for the level sets be decoupled from the meshless model so that a structured grid can be used for the level sets which is not necessarily be the same as the nodal arrangement. The independent level set grid will remove the problem since the approximation and spatial difference scheme used in the LSM based on a regular grid are all interpolatory. The advantages of using an independent level set grid are as follows.

1. The level set data storage and computation can be vectorized which can greatly reduce the computational cost. The advantages of using a structured grid in the level sets can be preserved and the solution accuracy can be ensured with stable numerical performance.

2. The lack of the Kronecker delta property of the shape functions in the meshless

method does not affect the level sets presentation of the original crack surface and its extension.

3. The level set grid can be placed locally and moves as the crack propagates.

In [116], the same nodal arrangement used for the XFEM is used for the LSM, which is a structured nodal arrangement. The decoupled LSM grid proposed here is not restricted to this requirement so that arbitrary nodal arrangements can be used taking full advantage of a meshless method. The only drawback here is the need for extra data storage for the level set grid.

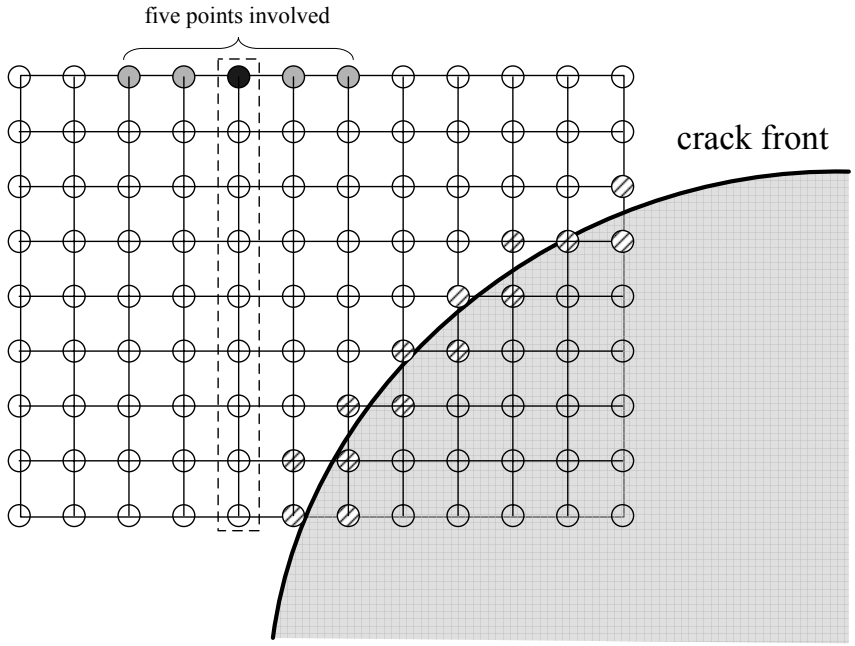


Figure 8.3: Vectorized computation of the LSM: regular stencils used in WENO and finding the neighbour points.

As the level sets are decoupled from the meshless model and to use a structured grid, most of the computation can be vectorized. Here we demonstrate an example of finding the spatial derivatives at grid points. If the level set grid is structured then the data access and storage are independent between each dimension. It means the spatial derivatives can be obtained in a certain dimension just using the information in that dimension. The spatial derivatives used here are based on the weighted essential non-oscillatory (WENO) scheme [203] in

which the derivatives of ϕ at a grid point, for example the derivatives at point $\mathbf{x}_{(i,j,k)}$ in x direction denoted as $\phi(\mathbf{x}_{(i,j,k)})_{,x}$, are calculated from the level values $\{\phi(\mathbf{x}_{(i-2,j,k)}),\ \phi(\mathbf{x}_{(i-1,j,k)}),\ \phi(\mathbf{x}_{(i,j,k)}),\ \phi(\mathbf{x}_{(i+1,j,k)})\ \phi(\mathbf{x}_{(i+2,j,k)})\}$ (refer to Appendix B.2 for details). Note the above expression is the same for all grid points inside the domain so instead of calculating one grid point at a time, the derivatives for all grid points can be obtained just within one step computation as explained in Figure 8.3.

8.2.2 A tying procedure with the visibility criterion in 3D

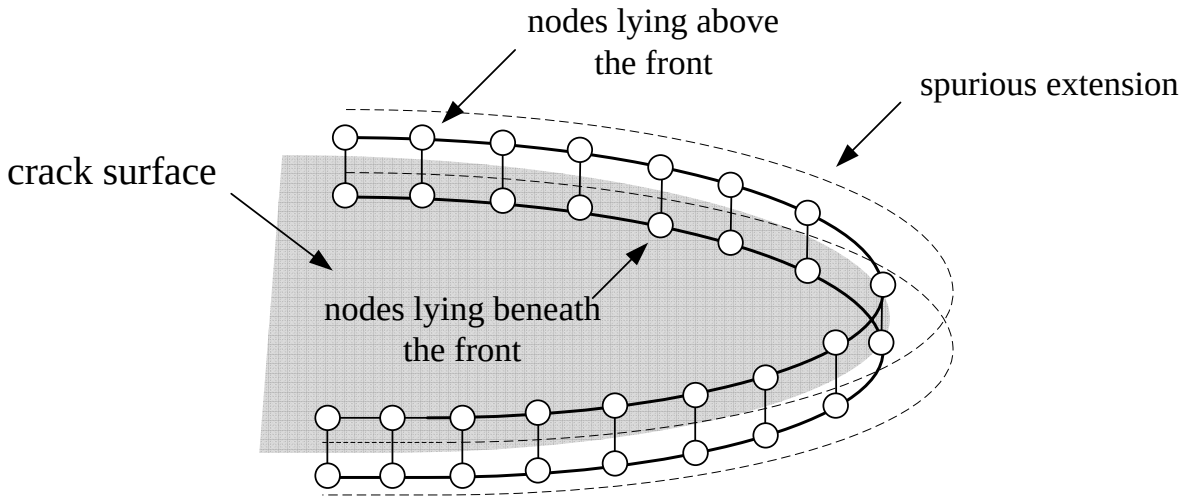


Figure 8.4: The tying "element" along arbitrary curved crack front in 3D.

When the visibility criterion is used to introduce the displacement jump across a crack, there exists spurious crack extension ahead of the crack as discussed in §7.2. To fix this problem, a tying procedure was proposed for 2D crack modelling in §7.3. In this section, we extend the tying formulation to 3D crack modelling. Though the underlying principles are the same as in 2D, some issues arise concerning implementation due the increasing complexity of crack geometries in 3D. A crack in 3D has a crack front that is a line, either straight or curved as shown in Figure 8.4. In 2D the tying procedure closes the crack tip where it corresponds to the original location of the crack tip while in 3D the tying term corresponding to Equation (7.5)

is applied to the crack front in the following form

$$W_{\text{tying}} = \frac{1}{2}\beta \int_{\Gamma_{\text{cr}}} [\mathbf{u}^i - \mathbf{u}^j] \cdot [\mathbf{u}^i - \mathbf{u}^j] \, \mathrm{d}\Gamma \tag{8.2}$$

where Γ_{cr} is the boundary of a crack front, $\mathbf{u}^i$ and $\mathbf{u}^j$ are displacements at pairs of nodes above and beneath the crack surface respectively lying along the crack front as shown in Figure 8.4 and the definition and value of β as a penalty parameter is the same as in Equation (7.5). Compared to Equation (7.3) applied to a crack tip, Equation (8.2) is used to apply the closure condition along the crack front. It calculates the energy release along a crack front due to the spurious extension when using the visibility criterion as shown in Figure 8.4. Similarly, Equation (8.2) is an additional term adding to the total energy described by (7.2). In implementation, pairs of tying nodes are arranged along a crack front as shown in Figure 8.4. This can be simply achieved using the procedure based on level sets which will be described in §8.2.6. In the test examples, the nodal spacing between neigbouring tying nodes lying on same side of crack surface is taken to be one half of the average nodal spacing near the crack surface.

8.2.3 Jump enrichment based on level sets

In §6.4.2, jump enrichment for arbitrary crack surfaces based on level sets was developed. For completeness, we repeat this formulation in the following.

- Consider a point of interest $\mathbf{x}$ and a set of N nodes in support of $\mathbf{x}$.
- Loop over each node $I\,(I = 1, 2, \cdots, N)$ for $\mathbf{x}$ to find the point $\mathbf{x}_c$ which satisfies the following conditions

$$\begin{cases} \mathbf{x}_c &= t(\mathbf{x} - \mathbf{x}_I) \quad (0 < t < 1) \\ \phi(\mathbf{x}_c) &= 0 \end{cases} \tag{8.3}$$

  Thus the point $\mathbf{x}_c$ found is the intersecting point of the line linking $\mathbf{x}$ and node I and the crack surface.

- If $\psi(\mathbf{x}_c) > 0$, i.e. point $\mathbf{x}_c$ falls outside the crack surface, then node I is excluded from the approximation at $\mathbf{x}$, otherwise node I included.

To find out the $\mathbf{x}_c$, we use the following simple iteration

$\mathbf{x}_c = \mathbf{x}_I$

`while` $|\phi(\mathbf{x})| > \mathrm{e}_{\mathrm{tol}}$

$\mathbf{x}_c = \tilde{\mathbf{x}}_I$

$t = \dfrac{\phi(\mathbf{x}_c)}{\phi(\mathbf{x}_c) - \phi(\mathbf{x})}$

$\tilde{\mathbf{x}}_c = \mathbf{x} - t(\mathbf{x}_c - \mathbf{x})$

$\mathbf{x} = \tilde{\mathbf{x}}_c$

`end`

where $\mathbf{x}_I$ are the Cartesian coordinates of node I, and $\mathrm{e}_{\mathrm{tol}}$ is the tolerance to stop the iteration ($\mathrm{e}_{\mathrm{tol}} = 0.001$ is used in the tests here). Since node I and the intersecting point is within the vicinity of crack, the iteration converges after one or two cycles.

8.2.4 Gradient projection method for locating the crack front

In this section, a method for locating a crack front based on projection along the gradient of level sets is proposed. It is believed to be the first time this method has been used in fracture modelling with level sets. The accurate positioning of the crack front is needed for two purposes, firstly, new nodes need to be inserted aligned with the new crack front to better simulate the effects of front curvature on the SIFs. Secondly, accurate location is required to calculate the SIFs along the front. The crack front is the solution to the intersection of the two level sets as

$$\mathbf{x}_{\mathrm{front}} = \psi^{-1}(0) \cap \phi^{-1}(0) \tag{8.4}$$

where $\mathbf{x}_{\mathrm{front}}$ is an infinite set of points along the front. In the numerical implementation, the solution to Equation 8.4 is taken at a limited number of points along the crack front. These sampling points can be determined from the level set grid so that they are not too dense or too sparse, and most importantly have the same level of resolution with respect to the grid point. The scheme in [204] is used to find the projection of a point on the interface which was developed for 2D problems with one level set. Here we extend it to find the intersection of two level sets in 3D. The

projection of a grid point $\mathbf{x}$ near the interface can be found as

$$\mathbf{x}^* = \mathbf{x} + \alpha_\phi \mathbf{p}_\phi \tag{8.5}$$

where $\mathbf{x}^*$ is the projection of the grid point on the interface and $\mathbf{p}(\mathbf{x})$ is the unit steepest ascent direction at $\mathbf{x}$ calculated by

$$\mathbf{p}_\phi = \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|} \ , \tag{8.6}$$

and α_ϕ is a scalar which is obtained by solving a quadratic polynomial equation

$$\phi(\mathbf{x}) + \|\phi(\mathbf{x})\|\alpha_\phi + \frac{1}{2}\left(\mathbf{p}_\phi^T \mathrm{He}(\phi)\mathbf{p}_\phi\right)\alpha_\phi^2 = 0 \tag{8.7}$$

in which $\mathrm{He}(\phi)$ is the Hessian matrix of second derivatives with respect to the state space variables as

$$\mathrm{He}(\phi) = \begin{bmatrix} \phi_{,xx} & \phi_{,xy} & \phi_{,xz} \\ \phi_{,yx} & \phi_{,yy} & \phi_{,yz} \\ \phi_{,zx} & \phi_{,zy} & \phi_{,zz} \end{bmatrix} \ . \tag{8.8}$$

In order to find the intersection of the two level sets, the above process is also applied to the tangent level set $\psi(\mathbf{x})$. Then the crack front can be found using the following steps:

1. Find set F_ϕ collecting all the grid points having their seven-point stencils cut by $\phi(\mathbf{x}) = 0$, and F_ψ collecting all the grid points having their seven-point stencils cut by $\psi(\mathbf{x}) = 0$.

2. Take the intersection of these grid points, i.e. $\{\mathbf{x}_{\text{near}}\} = F_\phi \cap F_\psi$, which are the points having their seven-point stencils cut the front.

3. Find the projection for each point in $\{\mathbf{x}_{\text{near}}\}$ on the crack front iterating as follows:

   `while` $\mathrm{e}_{\mathbf{x}} > \mathrm{e}_{\text{tol}}$

   $\tilde{\mathbf{x}} = \mathbf{x} + \alpha_\phi \mathbf{x}_\phi$

   $\mathbf{x}^* = \tilde{\mathbf{x}} + \alpha_\psi \mathbf{x}_\psi$

   $\mathbf{x} = \mathbf{x}^*$

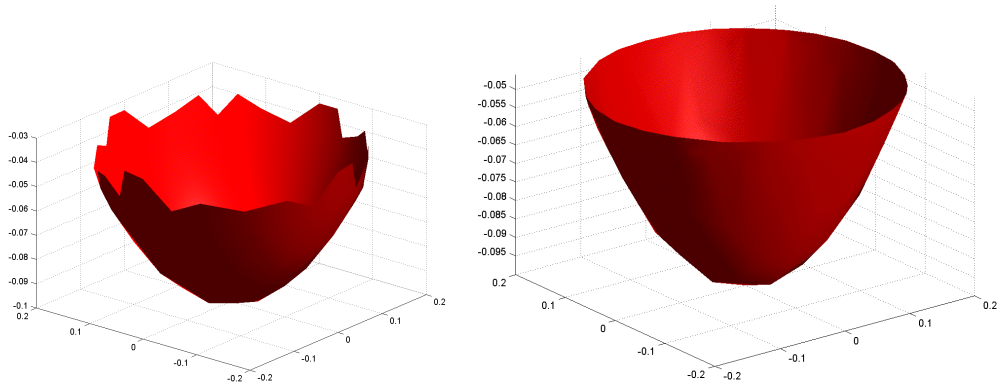
   `end`

4. These projected points represent the crack front and have same characteristic spacing with respect to the level set grid.

In step 1, the definition of a seven-point stencil at a grind point is shown in Figure 8.7 which comprises a grid point $\mathbf{x}_{\{i,j,k\}}$ itself and the six directly adjacent points, i.e. $\mathbf{x}_{\{i\pm1,j\pm1,k\pm1\}}$. The sets F_ϕ and F_ψ collect grid points neighbouring the interface $\phi = 0$ and $\psi = 0$ respectively. To determine whether a stencil is cut by an interface requires calculating to find if the level set values at that grid point takes opposite signs to any of the level set value at the six adjacent points. In step 3, α_ψ is obtained in a similar way to solve α_ϕ (using Equation (8.7)). The error norm $\mathrm{e}_\mathbf{x}$ is calculated as $\|\mathbf{x} - \mathbf{x}^*_\phi\|$ where $\|\cdot\|$ is the L_2 norm of a vector (refer to Equation (1.21)). The iteration converges as long as $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ are not tangential to each other. Since the two level sets are orthogonal to each other, the iteration converges quickly. (For example, in the test of the elliptic crack front described later in §8.4.3 convergence to less than $1\mathrm{e}-8$ is achieved after eight iterations.) The iterative procedure can be visualized as a point is moving in a zigzag path of small increments in directions orthogonal after to each other. When the point moves in the direction of the gradient of $\phi(\mathbf{x})$, $\psi(\mathbf{x})$ hardly changes. In Figure 8.5 the crack front found by direction interpolation and the proposed gradient projection method are compared. It can be clearly seen that for a curved crack surface where neither gradients of $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ are constant, the direct interpolation method fails to capture the correct crack front. The accuracy of this information is crucial to the computation of SIFs as well as for the correct enrichment of jump term in stress analysis.

8.2.5 Extending velocity without solving HJ-PDE

The velocity of the level sets along the crack front are determined by the crack growth rate calculated based on the SIFs. This is detailed in §6.3.2 and not repeated here. In this section, the process of extending velocity from the crack front to the entire domain is covered. In §6.3.3 the velocity extension method proposed by Gravouil et al. [116] was described. Recall that the method extends the velocity from the crack front to the entire level set domain by solving the following two pairs of HJ-PDEs



(a) By direction interpolation. (b) By gradient projection method

Figure 8.5: A comparison of the crack front found by the direction interpolation and gradient projection method for the lens shaped crack.

for V_ψ and V_ϕ to steady state respectively.

$$\begin{cases} \dfrac{\partial V_\psi}{\partial \tau} + \text{sign}(\phi)\dfrac{\nabla \phi}{|\nabla \phi|} \cdot \nabla V_\psi = 0 \\ \dfrac{\partial V_\psi}{\partial \tau} + \text{sign}(\psi)\dfrac{\nabla \psi}{|\nabla \psi|} \cdot \nabla V_\psi = 0 \end{cases} \tag{8.9}$$

$$\begin{cases} \dfrac{\partial V_\phi}{\partial \tau} + \text{sign}(\phi)\dfrac{\nabla \phi}{|\nabla \phi|} \cdot \nabla V_\phi = 0 \\ \dfrac{\partial V_\phi}{\partial \tau} + \text{sign}(\psi)\dfrac{\nabla \psi}{|\nabla \psi|} \cdot \nabla V_\phi = 0 \end{cases} \tag{8.10}$$

Solving the above PDEs is a time consuming process, even longer than updating the level sets. Of more concern is the accuracy of the front speed. From the tests undertaken here the method is found not to converge for a curved crack front, i.e. residual relative error is about $1\% - 3\%$. Secondly the velocity extended from the crack front is not maintained during the extension. This problem does not manifest itself for an edge crack, which has a straight crack front, only becoming significant for a curved crack front. Taking a penny-shaped crack with suppose constant velocity along the front. When the velocity is extended from the front to the whole domain, the error between successive iteration steps does not converge in solving Equations (8.9) to (8.10). In this particular test, v_ϕ and v_ψ along the crack front were assigned unit value, i.e. $v_\phi = 1$ and $v_\psi = 1$. Since the velocity is constant along the front,

an ideal extension process would give velocity at any point in the domain also of unit value for both v_ϕ and v_ψ. However the velocities obtained at some points were larger or smaller than this unit value. To address the above problems, a gradient projection method is developed in the following which does not involve solving these PDEs and preserves the front speed during extension. As in the previous section, the crack front is represented by a set of points and front speed is solved for these points. So the remainder of the task is to extend the speed from these points to the entire level set domain. We can find its projection on the crack front for each grid point and interpolate the speed from its neighbour points. The procedure is summarised as follows:

`Loop over each grid point` $\mathbf{x}_I$ `of level sets`

1. find the projection $\mathbf{x}^*$ of $\mathbf{x}_I$ on the crack front
2. find the two nearest points, denoted as $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ respectively, to $\mathbf{x}^*$ from $\{\mathbf{x}_{\text{front}}\}$
3. interpolate V_ϕ and V_ψ at $\mathbf{x}^*$ from $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$
4. assign the speed to the grid point $\mathbf{x}_I$

`end`

In the above, the speed at $\mathbf{x}^*$ is interpolated from the speed at $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ by

$$V_{\mathbf{x}^*} = V_\phi^{(1)} \frac{\dfrac{1}{S_1^2}}{\dfrac{1}{S_1^2}+\dfrac{1}{S_2^2}} + V_\phi^{(2)} \frac{\dfrac{1}{S_1^2}}{\dfrac{1}{S_1^2}+\dfrac{1}{S_2^2}} \tag{8.11}$$

where S_1 and S_2 are calculated by

$$S_1 = \|\mathbf{x}^* - \mathbf{x}^{(1)}\| \tag{8.12}$$

$$S_2 = \|\mathbf{x}^* - \mathbf{x}^{(2)}\| \ . \tag{8.13}$$

The above interpolation is a radial function but one could also design many other methods to interpolate. Tests carried out in this work show the above interpolation method to work well for straight or for curved crack fronts. The above process effectively extends the velocity from the crack front to the entire domain along the gradients of ϕ and ψ as shown in Figure 8.6.

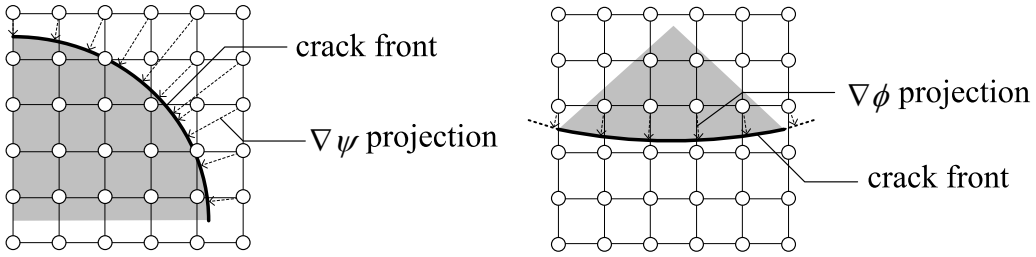


Figure 8.6: The front speed extended from the crack front along the gradient of level sets.

8.2.6 Refinement of nodes based on level sets

This section presents the use of level sets for building the meshless model and the reduction of the computational cost of stress analysis. Level sets are found to be useful in adding jump nodes and refining the integration points near a crack front. They also help to confine the region where the enrichment of field approximation is necessary.

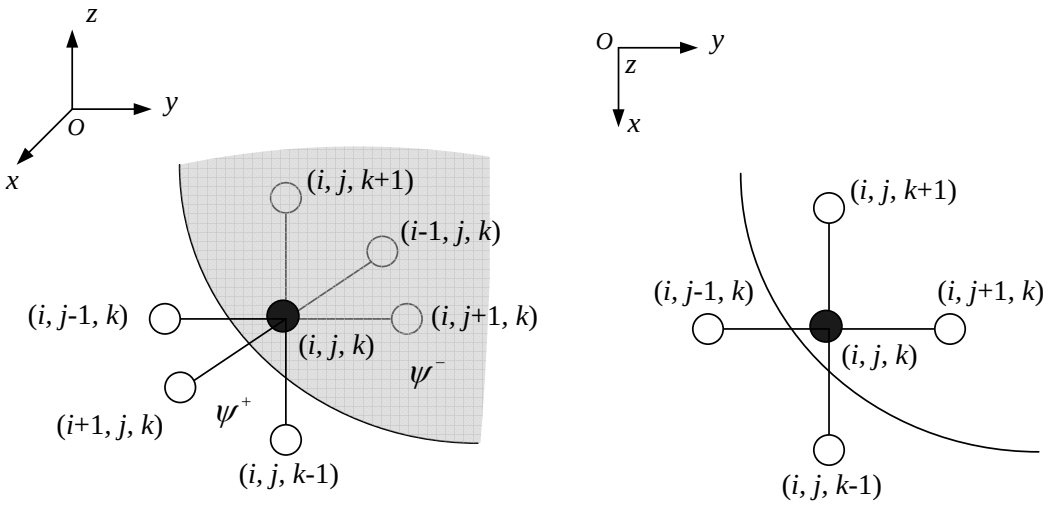


Figure 8.7: The seven-point stencil of a grid point cut by the interface.

As was discussed in §7.4.2, accuracy in fracture modelling using the EFGM is improved by inserting additional pairs of nodes aligned with the crack on either side. These nodes are termed as "jump nodes". As the crack grows, new jump nodes are needed to cope with the new crack geometry. This procedure, which is trivial in 2D, can become complicated in 3D considering the curvature of crack surface and front.

However with the help of level sets, it can be realized simply and the formulation is generally effective for arbitrary crack surfaces. The developed procedure is as follows:

1. Find the set of grid points $N_{\text{cut}} = \{\mathbf{x}_j, j = 1, \cdots, n_{\text{cut}}\}$ that have their seven-point stencil cut by the normal level sets $\phi(\mathbf{x}_j) = \pm\delta\phi$ which satisfy $\psi(\mathbf{x}_j) \leq \epsilon_\psi$, where $\delta\phi = 0.01L_K$ is the offset of jump nodes from the crack surface and L_K is the characteristic dimension of the crack e.g. the radius of a penny-shaped crack or half width of an edge crack, j is the index of a grid point (ϵ_ψ is a small positive value and $\epsilon_\psi = 0.001$ is used here).

2. Project the grid point N_{cut} found in the previous step according to §8.2.5 to get the set of points $N_K = N_\phi \cap N_\psi$ where $N_\psi = \{\mathbf{x}|\psi(\mathbf{x}) \leq \epsilon_\psi\}$, $N_\phi = \{\mathbf{x}|\phi(\mathbf{x}) = \pm\delta\phi\}$.

Examples including the addition of jump nodes using the above described algorithms are presented in §8.4.

8.2.7 PU integration refinement near the crack front

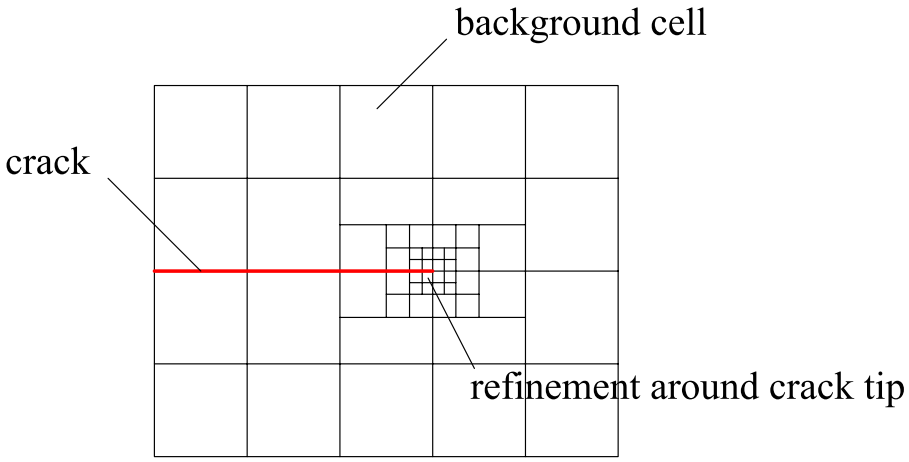


Figure 8.8: Partitioning integration cell around the crack tip in 2D.

In crack modelling, a large number of integration points are needed around the crack front (tip in 2D) due to the high energy release. It was pointed out by Fleming et al. [159] that using a high order integration scheme near the crack tip can generally improve the solution accuracy but refinement in locations behind the crack

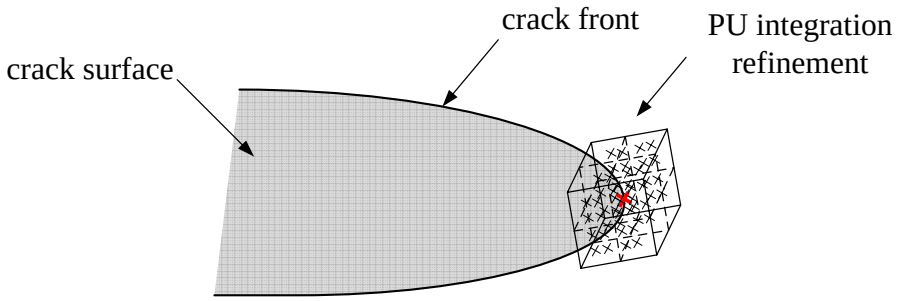


Figure 8.9: Refinement of integration in the vicinity of the crack front.

front does not increase accuracy. In [162], a high order integration scheme, 8×8 Gauss integration, is used throughout the domain. This is simple for implementation but expensive in computational cost. Partitioning the background cell near the crack tip was previously reported in [96, 205] as shown in Figure 8.8 where the background cell containing the crack tip is partitioned into smaller cells. When the crack propagates, the cells need to be updated which is similar to a remeshing process. To overcome this problem particular to 3D, an adaptation of the PU based integration method proposed in [162, 206] is presented here. Firstly, a basic background mesh is used which is not changed as the crack propagates and integration points are generated using a specific quadrature scheme. Secondly, all the integration points which are close to the crack front are located and the integration refined there using a PU function. An example in 3D is shown in Figure 8.9. A small cubic volume is constructed around the point, and the size of the which is determined by the integration spacing at this point, e.g. the distance to the nearest neighbouring integration points. Then the cube is partitioned into smaller $2 \times 2 \times 2$ cubes and within each sub-cube a $4 \times 4 \times 4$ Gauss integration scheme is used. Therefore the number of refined integration points is 512 times the original. The weight factor for each newly-refined integration point will be factorized by the weight of the original integration point thus conforming into the PU condition. The criterion to judge if the integration point is close to the crack front is based on the level sets. An integration point will be refined if it satisfies

$$\sqrt{\phi(\mathbf{x}_G)^2 + \psi(\mathbf{x}_G)^2} \leq F_G L_K \tag{8.14}$$

where $\mathbf{x}_G$ denotes the coordinates of the integration point, F_G is a factor determining

the size of the volume near the crack front that needs to be refined ($F_G = 0.1$ is used here), and the definition of L_K is the same as in §8.2.6. An example showing improvement of results using the PU based integration refinement is given in §8.4.2.

8.2.8 Confinement of jump enrichment

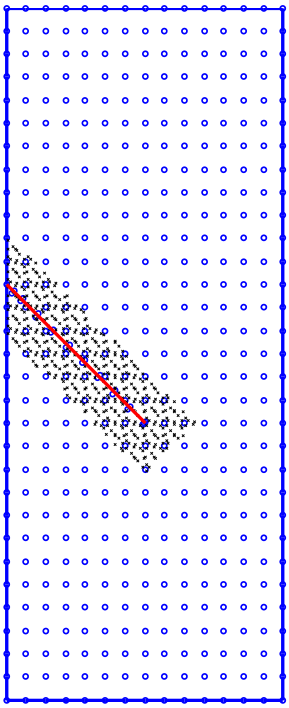


Figure 8.10: An illustration of the integration points in the vicinity of the 2D slanted edge crack where the displacement jump based on the level sets needs to be considered.

As discussed previously in 7.5 the total number of integration points required in meshless methods is much greater than in the FEM or XFEM for comparable accuracy, apart from the computational cost in obtaining the shape functions, the cost of introducing the displacement jump can be heightened for 3D problems where the number of integration points and number of nodes in support for each point is higher. Suppose the total number of quadrature points is n_Q and the average number of nodes in support is n_I, then the total calls to a visibility criterion algorithm is equal to $n_Q \times n_I$. This cost can be greatly reduced if the application of the visibility criterion is confined only within the vicinity of crack. This can be simply and

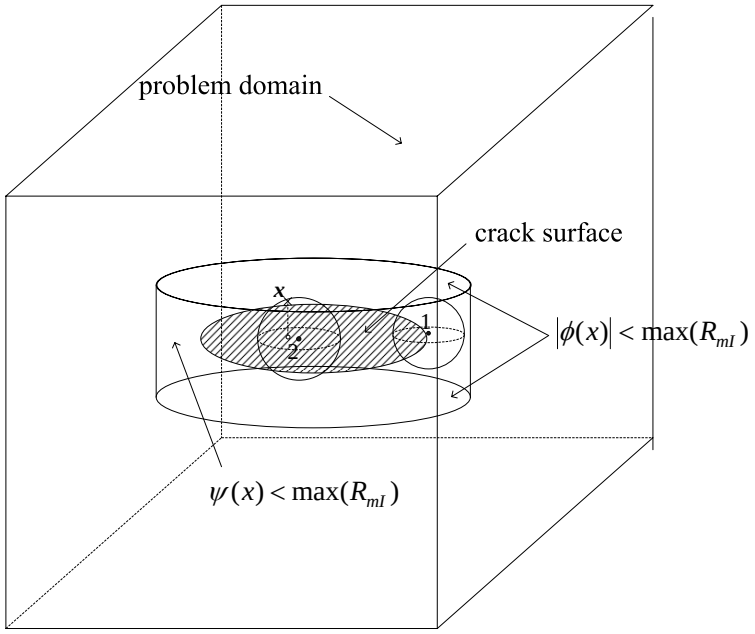


Figure 8.11: The volume in the vicinity of an arbitrary crack surface in 3D in which the displacement jump needs to be considered.

effectively realized with the help of level sets. For example, consider a crack surface shown as the shaded area in Figure 8.11 contained within a cubic domain, two integration points marked with A and B and four nodes labeled 1-4. The nodal supports are shown as spheres, where node 1 and 3 are in support of A and node 4 in support of B. Node 1 is close to the crack surface but is neither in support of A or B. Node 2 and A need to be considered for jump enrichment while node 4 and B do not. From above observations, it can be found that the visibility criterion is only necessary when an integration point and its supporting nodes are very close to the crack surface. The cylindrical volume bounding the crack confines the region where any point of interest falling inside will potentially involve the jump enrichment. Any point outside the volume is too far from the crack to have a supporting node with nodal support cut by the crack surface, and is therefore not necessary to be judged with the visibility criterion. In the following a criterion confining the jump enrichment is proposed. An integration point satisfying the following conditions will

be considered for jump enrichment

$$|\phi(\mathbf{x}_Q)| \leq R_{\max}$$
$$\psi(\mathbf{x}_Q) \leq R_{\max}$$

where $\mathbf{x}_Q$ are the coordinates of an integration point, and the subscript Q indicates the quadrature point and $R_{\max}$ is the maximum size of nodal support for all the nodes inside the domain. For 2D cracks, the confined region is a banded area near a crack. For example, the slanted edge crack problem in §7.5.2 will have the black points shown in Figure 8.10 to be considered for jump enrichment. In 3D the confined region becomes a volume near a crack surface. In tests it was found the above confinement method cuts down approximately 75% of the calls to visibility criterion. This will be demonstrated in the elliptic crack example in §8.4.3. It should be noted that the level of reduction varies between different problems regarding the size of a crack geometry and the size of the domain.

8.3 Fracture analysis

8.3.1 Computation of SIFs in 3D using LSC

The J integral in 3D is used here since it is simple to implement and requires less computational cost. For 3D problems, the J integral needs to include the energy contribution from x_3 components which are not present in a 2D crack problem (refer to §5.4.2). So the J integral including the area terms from x_3 components [173] is then

$$J^M = \int_\Gamma \left(W^M n_1 - \sigma_{ij}^M \frac{\partial u_i^M}{\partial x_1} n_j \right) \mathrm{d}\Gamma + \int_{\Omega(\Gamma)} \left(\frac{\partial \sigma_{i1}^M}{\partial x_1} + \frac{\partial \sigma_{i2}^M}{\partial x_2} \right) \frac{\partial u_i^M}{\partial x_1} \mathrm{d}\Omega \qquad (8.15)$$
$$- \int_{\Omega(\Gamma)} \sigma_{i3}^M \frac{\partial^2 u_i^M}{\partial x_1 \partial x_3} \mathrm{d}\Omega \quad M = \mathrm{I}, \mathrm{II}, \mathrm{III}$$

where $\Omega(\Gamma)$ is the area bounded by Γ, the contour of J integral and the definition of W and other terms are the same as in Equation (6.4). For non-planar crack surface, Equation (8.15) is not accurate as the crack surfaces cannot be assumed traction free and an additional term taking into account the traction along crack surfaces needs to be added as described in [180]. However the integration is performed in the

vicinity of the crack front then it does not effectively involve the non-planar part and can be regarded as planar crack surface. Note Equation (8.15) is performed with all the variables mapped to the local coordinates which changes along the crack front. The unit base vectors in the local coordinate system can be built from the level sets as

$$\mathrm{e}_1 = \frac{\nabla\psi}{\|\nabla\psi\|} \tag{8.16}$$

$$\mathrm{e}_2 = \frac{\nabla\phi}{\|\nabla\phi\|} \tag{8.17}$$

$$\mathrm{e}_3 = \mathrm{e}_1 \times \mathrm{e}_2. \tag{8.18}$$

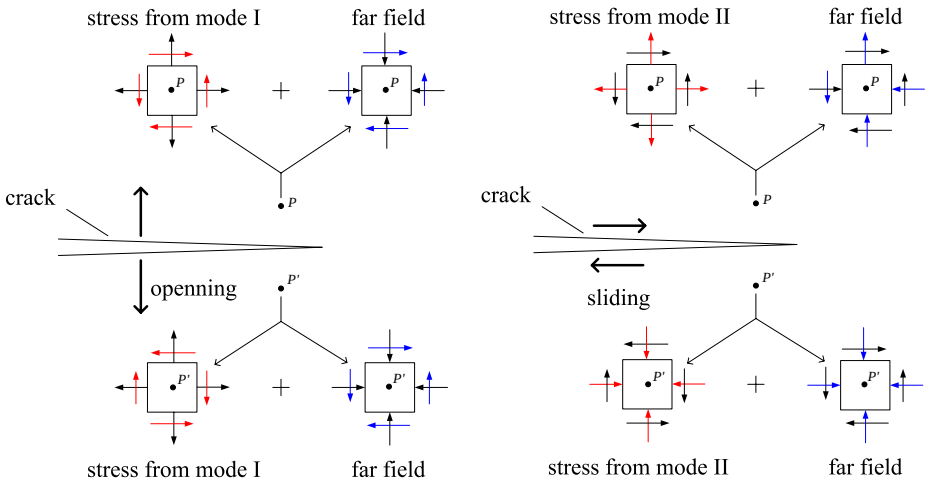
The transformation that relates the global tensor with the local tensor in 3D is similar to the formulation described in §6.3.1 in 2D. In the all the test examples in §8.4, the J integral is performed along a circular path around a point lying on the crack front where it is divided into 40 smaller line segments, and for each segment a four-point Gaussian integration rule is used.

8.3.2 Determining the signs of SIFs

The decomposition of the J integral allows the calculation of SIFs for each mode independently as described in §5.4.2. However a drawback of the method is that the signs of SIFs cannot be determined. As shown by Equations (5.50), the J integral as the energy release rate is always positive and information on the signs of the SIFs is not provided. In 2D examples tested in Chapter 7, the sign of each mode can be easily judged from the deformation of the crack. However for 3D cracks, it becomes difficult to do so, particularly when the SIFs change signs along the crack front. As will be seen in the following mixed-mode 3D crack examples, the relation between deformation and crack mode is complicated to visualize as compared to 2D. To solve this problem, a simple method is proposed in the following which adds an additional step after the J integral calculation. The idea is to multiply the near tip field obtained by numerical modelling over the exact solution by the following equation

$$\mathrm{sign}(K_{\mathrm{M}}) = \mathrm{sign}\left(\boldsymbol{\sigma}^{\mathrm{aux(M)}} \cdot \boldsymbol{\sigma}^{\mathrm{num(M)}}\right) \quad \mathrm{M} = \mathrm{I},\ \mathrm{II},\ \mathrm{III} \tag{8.19}$$

where $\boldsymbol{\sigma}^{\mathrm{aux(M)}}$ is a stress vector comprising a certain stress component, e.g. local shear stress σ_{12}, along the integration path Γ. The superscript aux indicates the auxiliary solution of the near tip field, num indicates the stress results from numerical modelling and M indicates results corresponding to a certain crack mode. The choice of stress component for $\boldsymbol{\sigma}$ will be discussed in the following. The auxiliary field of each mode is directly calculated using the analytical solutions of each mode while the numerical solutions for each mode follows the stress decomposition method described in §5.4.2. For example, to determine the sign of K_{I} we firstly substitute $K_{\mathrm{I}}^{\mathrm{aux}} = 1$, $K_{\mathrm{II}}^{\mathrm{aux}} = 0$, $K_{\mathrm{III}}^{\mathrm{aux}} = 0$ into Equations (C.1.10) to obtain the auxiliary stress around a pure mode I crack. Then we substitute the exact stress and numerical solution of mode I in Equation (8.19) to determine the sign of K_{I}. Similarly for K_{II}, we use $K_{\mathrm{I}} = 0$, $K_{\mathrm{II}} = 1$, $K_{\mathrm{III}} = 0$ which produces a pure mode II stress field. In an initial trial, this procedure was tested for the mixed-mode edge crack problem in §7.5.2 choosing stress components $\sigma_{11}^{\mathrm{aux(M)}}$ with $\sigma_{11}^{\mathrm{sol(M)}}$. However the procedure yields an incorrect sign for the SIF of K_{I}. A close examination of stress results showed influence from far field uniform elastic deformation, illustrated as blue arrows in Figure 8.12, results in a stress which can take opposite signs from its corresponding near tip field values. To exclude the influence from this far field uniform elastic deformation, use is made of the antisymmetric stress component of each mode as shown in Figure 8.12. For example, in a mode I crack the shear stress σ_{12}, shown as the red arrows in Figure 8.12(a), takes opposite signs at a point P above the crack and its symmetric point P' beneath the crack. On the other hand, the normal stresses σ_{11} and σ_{22} take same signs at both P and P'. Since in each mode, the antisymmetric components are calculated by subtracting stresses at pairs of points symmetric to the crack, the influence from far field stresses is cancelled out. Therefore to determine the sign for K_{I}, stress component σ_{12} should be used for Equation (8.19) while for K_{II} and K_{III}, σ_{11} and σ_{13} should be chosen respectively.



(a) antisymmetric stress component in mode I

(b) by gradient projection method

Figure 8.12: The sign of symmetric and antisymmetric crack mode compared to the with the far field stresses.

8.4 Numerical results

In this section the full 3D framework for crack modelling incorporating all the components described above is tested with a number of single or mixed-mode crack examples. In each of the examples, crack size is varied and various boundary conditions are modelled. In all examples, the material parameters used are, Young's modulus $E = 1000$ and Poisson's ratio $\nu = 0.3$. The nodal arrangement for meshless models are generated by `gmsh` [142]. Here the fatigue crack propagation criterion is used to propagate the crack surface. As shown by Equation (5.66), the choice of C determines the size of crack increment and in the following $C = 0.1$ is used. Of course different values of C can result in a change of crack path and the smaller C the better the resolution of a crack path. However in implementation, considerations should also be given not only to the computational cost but also to a large enough size of C to perform the J integral. A convergence study of crack path with respect to the choice of C is presented in [207] indicating an iterative process to find the size of crack increment. However in the following examples this iteration procedure is omitted and the main purpose is to show the validity of the proposed framework. Five examples will be tested and these are arranged in ascending order

of difficulty and challenge. An overview of these examples is shown in Figure 8.13. The first example is an edge crack under tension, which has a straight crack front and under pure mode I. In the second, a penny-shaped crack which has a constant curvature along crack front is tested. It is followed by a more challenging problem of varying curvature along crack front of an elliptic crack. Mixed mode II and III cracking is tested in the fourth example of a penny-shaped crack under shear. The last example is an inclined penny-shaped crack subjected to all three crack modes and it is propagated for several steps which involve kinks the along crack front.

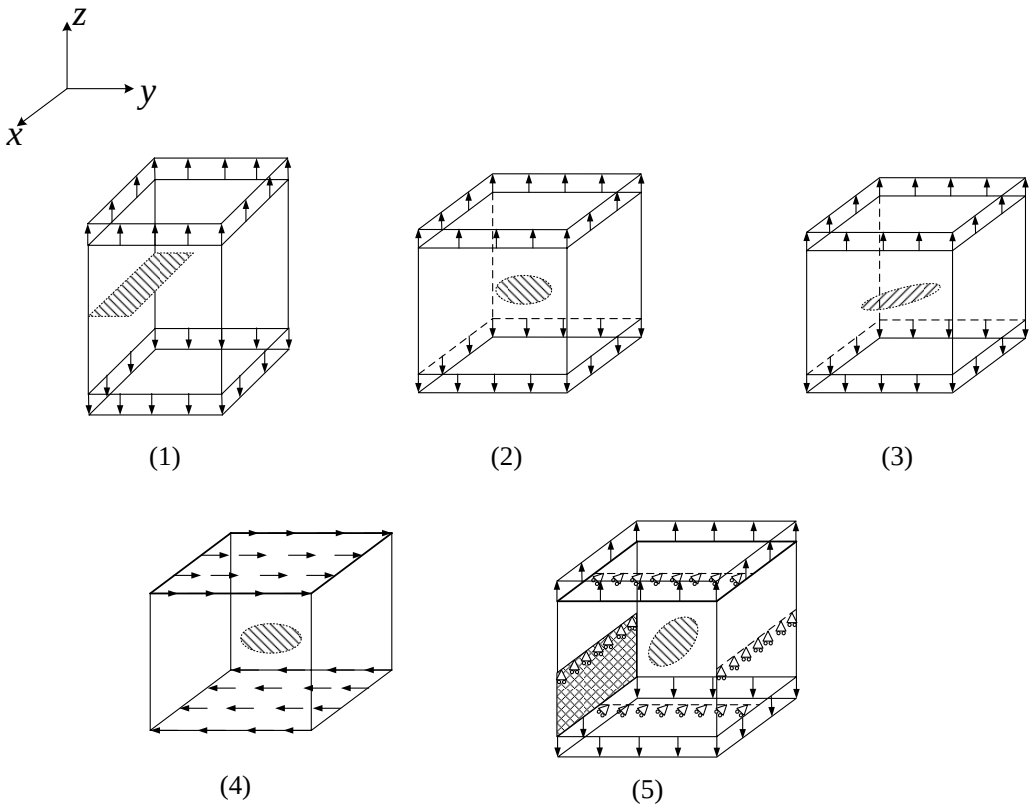


Figure 8.13: An overview of the five test examples.

8.4.1 Edge crack under tension

Stress analysis and predictions of SIFs are first tested for an edge crack in a 3D thick block as shown in Figure 8.14. A similar example has been previously used in [106, 108]. However in [108] the two side faces are perpendicular to the crack front are fully fixed the in x direction. So the block is actually enforced follow a

plane strain condition, which is not ideal as a 3D problem for testing. Here we do not apply an ideal plane strain boundary condition and instead the thickness t of the block is increased as shown in Figure 8.14, so parts of the model will be close to plane strain condition as like in [106]. The dimensions of the block used here are $w = 2$, $H = 2$ and $t = 2$, and uniaxial normal tractions $\sigma = 1$ are applied on the upper and lower surfaces of the block as shown in Figure 8.14. The two zero level sets of the crack surface are plotted in Figure 8.15. It can be seen that the crack front is here positioned from $x = -1$ to $x = 1$ at $z = 1$. A structured nodal arrangement of $21 \times 13 \times 13$ ($w \times H \times t$) nodes is used and additional jump nodes are added based on level sets according to §8.2.6. The zero level sets $\phi = 0$ and $\psi = 0$ are plotted in Figure 8.17. A number of tests have been carried out by varying the width of the crack surface a from 0.2 to 1.2. The deformation for $a = 0.8$ is plotted in Figure 8.16 showing that with the tying procedure applied to crack front there is no problem of spurious crack extension. The SIF results obtained are listed in Table 8.1 and are compared with the reference results in [108]. This is a pure mode I problem and K_{II} and K_{III} are zero. In the Table, K_{I} is the result at the centre of the front i.e. $x = 0$. The SIFs along the crack front from $x = -1$ to $x = 1$ for $a = 0.6$ vary and an example for $a = 0.6$ is plotted in Figure 8.17. It can be seen in 3D the stress field does not match a pure plane strain assumption (as expected) and K_{I} increases as the front goes from the boundary to the inside of the model. The SIF reaches a maximum at a half width of the model which is the point along front having the closest condition to the plane strain assumption. The SIFs along width of the model are also symmetric.

a	0.2	0.4	0.6	0.8	1.0	1.2
KI (Ref.)	0.9750	1.6703	2.5399	3.6780	5.3351	8.0577
KI (present)	0.8954	1.6024	2.4373	3.5537	5.0670	7.5946
FI(Ref.)	1.2300	1.4900	1.8500	2.3200	3.0100	4.1500
FI (present)	1.1296	1.4294	1.7752	2.2416	2.8587	3.9115
Error(%)	-8.2	-4.1	-4.0	-3.4	-5.0	-5.7

Table 8.1: The SIF results of the single edge crack under uniaxial tension.

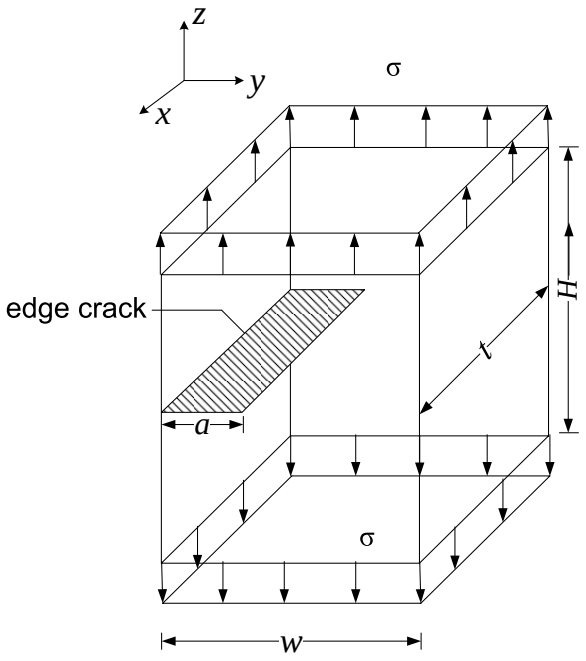


Figure 8.14: The geometry and boundary conditions of the single edge crack under uniaxial tension.

8.4.2 Penny-shaped crack under tension

The second example is a cuboid containing a penny shaped crack inside subjected to uniaxial tension. The geometry of the problem is shown in Figure 8.18. The crack surface is located in the centre of the cube and is perpendicular to the loading direction. The dimensions of cube are $2 \times 2 \times 2$ units and the radius of the penny-shaped crack a is varied from 0.2 to 0.7. The level set description of the crack surface has been previously plotted in Figure 6.13 for $a = 0.3$. The major difference from of from the edge crack example, above is the presence of a curved crack front, which requires better accuracy of geometric description. This is again a pure mode I problem and K_{II} and K_{III} are zero. The exact solution for the SIFs assuming the crack in an infinite domain is given in [200] as

$$K_{\mathrm{I}} = F_{\mathrm{I}}(\alpha, \theta) 2\sigma \sqrt{\frac{a}{\pi}} \tag{8.20}$$

where F_{I} is a factor depending on the size of the crack which tends to 1 if the boundary effects can be neglected with respect to the size of the crack. The SIFs obtained by the present method are compared with reference solutions in Table 8.2. The results show satisfactory agreement with reference solutions even without

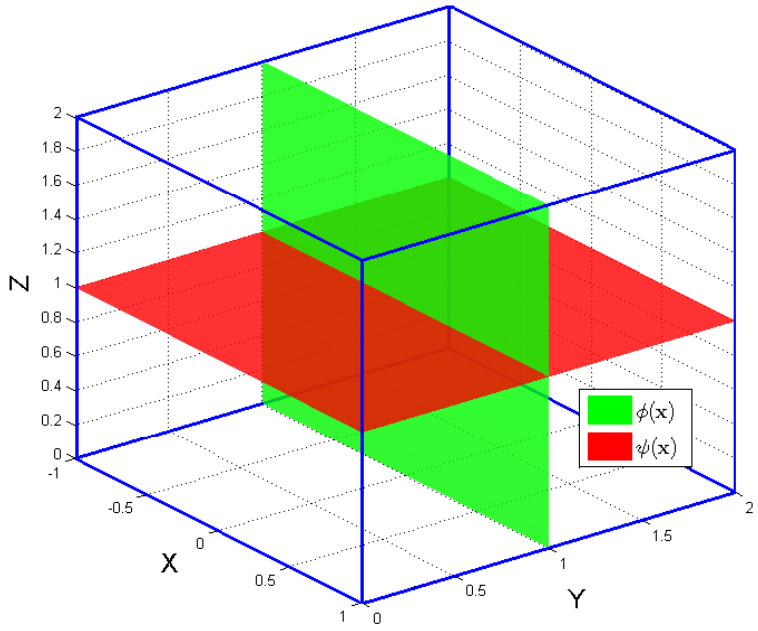
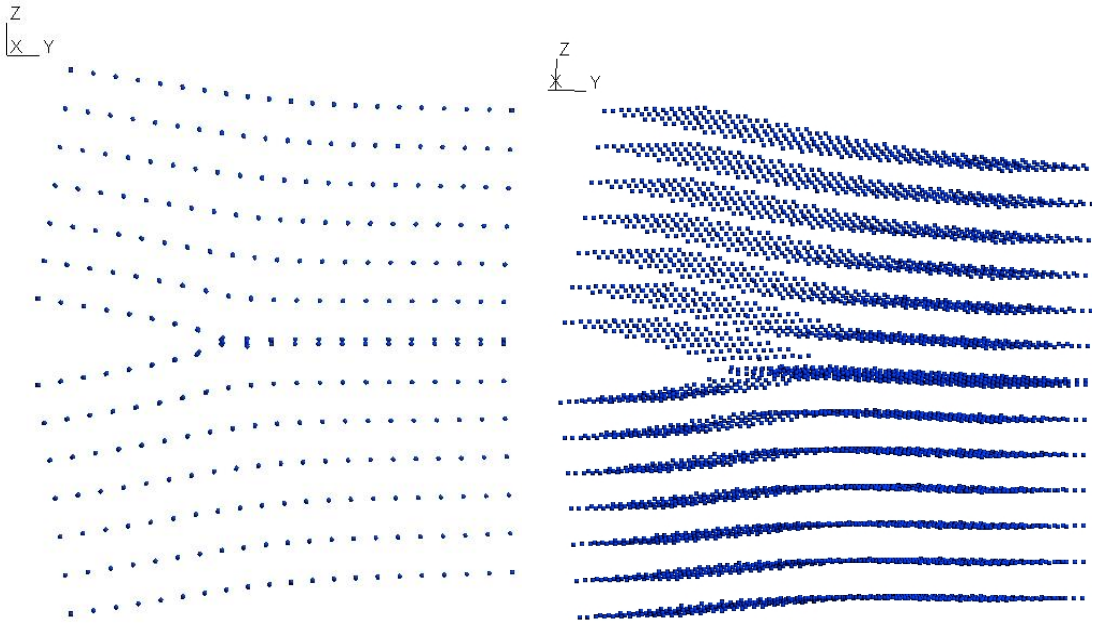


Figure 8.15: The level set description of the single edge crack under uniaxial tension.



(a) x view (b) xy view

Figure 8.16: Deformations of the 3D edge crack problem.

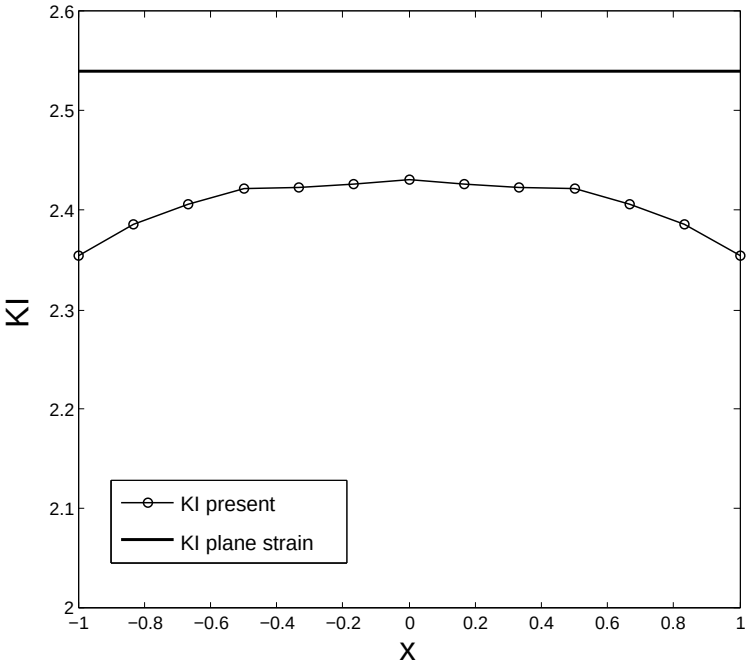


Figure 8.17: The K_{I} along the crack front for a 3D edge crack under uniaxial tension.

enrichment. It shows that for a penny-shaped crack with a curved crack front, the enrichment is not as critical to the SIF results as that for a crack with a straight front. The curvature radius of the front in the former is a constant while that in the latter is infinite. Note that the edge crack problem is close to a 2D edge crack problem where the enrichment is developed from the analytical solutions. A question arises for a penny-shaped crack whether the 2D enrichment method is still as effective for that as for the edge crack. This problem is worthy of further investigations and an effective enrichment method in 3D for curved crack front is a topic of interest. The accuracy of the SIFs however deteriorates when the size of crack becomes very small. Improved accuracy can be achieved using the PU refinement scheme described in §8.2.7 and the results are listed in Table 8.2, marked "PU".

8.4.3 Elliptic crack under tension

In the third example, an elliptic crack subjected to uniaxial tension perpendicular to the crack surface is tested. The crack surface is lying in the plane of $z = 0.5$ as shown in Figure 8.19 and the dimensions of the crack are $a = 0.6$ and $b = 0.3$. Compared to the penny-shaped crack where the curvature is constant along the front, in this

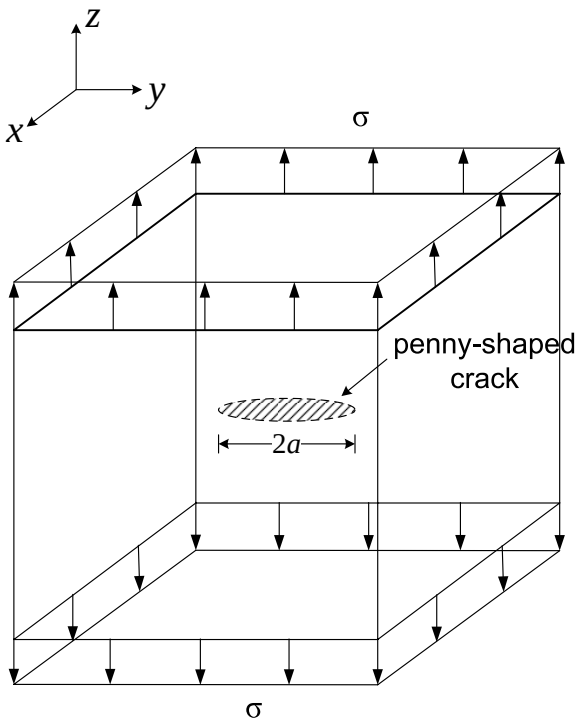


Figure 8.18: The geometry and boundary conditions of the penny-shaped crack under uniaxial tension.

a	0.3	0.4	0.5	0.6	0.7
FI(Ref.)	1.0270	1.0251	1.0518	1.1059	1.1770
KI (Ref.)	0.6180	0.7316	0.8392	0.9666	1.1113
KI (present)	0.5930	0.7016	0.8437	0.9638	1.1082
error KI (%)	-5.997	-4.094	0.534	-0.292	-0.267
KI (PU)	0.6250	0.7293	0.8365	0.9638	1.1082
error KI (PU)(%)	-0.925	-0.309	-0.324	-0.292	-0.267

Table 8.2: The SIF results of the penny-shaped crack under uniaxial tension.

problem the crack front has a varying curvature and hence the SIFs change along the front. This feature is difficult to capture using explicit methods such as using piecewise triangular facets since a smooth curvature variation cannot be correctly described. A finer nodal arrangement is used as compared to the previous example and the nodal density is about four-times the previous number with about 4000 nodes used in total. Due to the symmetry of the problem, only a quarter of the problem is modelled. The level sets of the crack are plotted in Figure 8.22(a) and the crack front obtained using the gradient projection method is shown in Figure 6.25. The integration points are processed using the localized jump enrichment described in §8.2.8 and only 25% of the all integration points are considered for the jump enrichment (marked with red crosses in Figure 8.21 with the blue cricles being nodes). The total computational time is reduced by 55% compared to the calculation without confining the region of jump enrichment. The exact solution to this problem assuming the crack embeded in an infinite domain is given in [200] as

$$K_{\mathrm{I}} = \frac{\sigma}{E(k)}\sqrt{\frac{\pi b}{a}}\left(\frac{a^2(a/b)^2\tan^2\theta + b^2}{1+(a/b)^2\tan^2\theta}\right)^{\frac{1}{4}} \tag{8.21}$$

where

$$\text{for} \quad a \geq b, \quad k = \sqrt{1-\frac{b^2}{a^2}}\,, \quad E(k) = \int_0^{\pi/2}(1-k^2\sin^2\theta)\,\mathrm{d}\theta \tag{8.22}$$

$$\text{for} \quad a \geq b, \quad k_1 = \sqrt{1-\frac{a^2}{b^2}}\,, \quad E(k) = \frac{b}{a}E(k_1)\,. \tag{8.23}$$

The exact solutions are calculated using Equation (8.21) at selected points along the crack front and are compared with those by the present new framework in Table 8.3. The position of the points on the crack front is indicated by θ which the angle between the unit outer normal at a point lying on the crack front and the positive x axis. The K_{I} results are plotted in Figure 8.20 showing slightly larger values than the exact solution. This is mainly due to the boundary effects as the exact solutions are assumed for an infinite domain. This agrees with the findings in [108] that a crack close to boundary tends to have larger SIFs than that of the same geometry and far from a boundary. The K_{I} results also indicate larger SIFs at the ends aligned with the short axis than those aligned with long axis as noted in Figure 8.22(b). It should also be noted that the error in the SIF along crack front is not constant.

However the trends of the SIF variation are the same as the analytical solution. A finer nodal arrangement locally is expected to improve accuracy, however this is not available at the moment using our in-house code in Matlab which is restrained by the memory limitation in solving the global stiffness matrix.

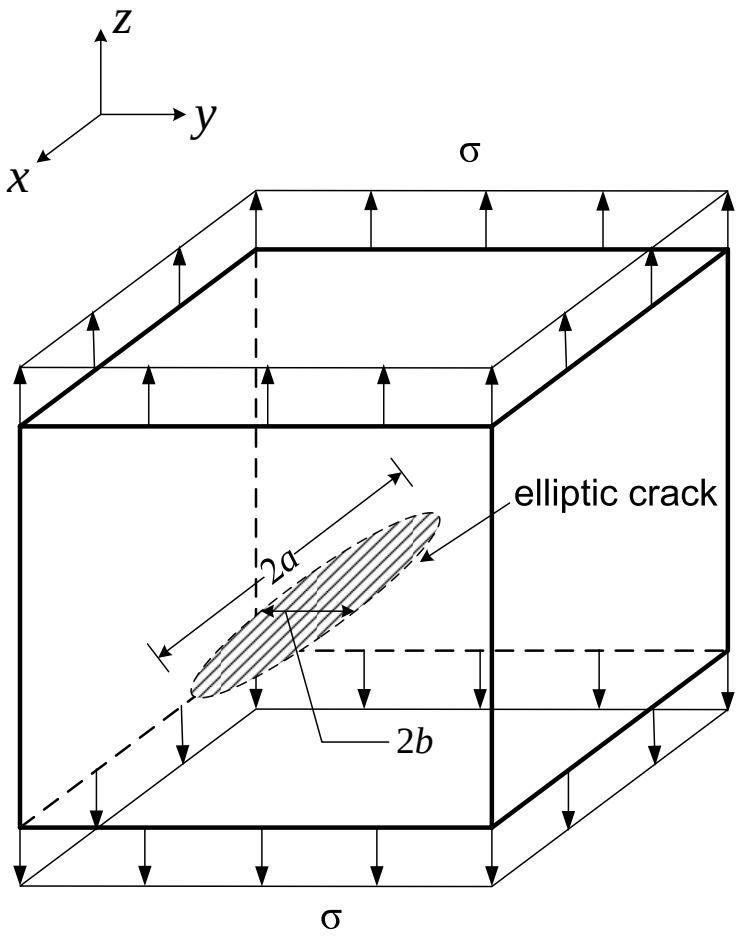


Figure 8.19: The geometry and boundary conditions of the elliptic crack under uniaxial tension.

8.4.4 Penny-shaped crack under shear

The next example is a mixed mode problem using the geometry of the previous penny-shaped crack example and only changing the boundary conditions to pure shear as shown in Figure 8.23. Uniform shear stresses $\tau_{zx} = 1$ are applied on the upper and lower surfaces of the cube as shown in Figure 8.23. The radius of the penny-shaped crack surface in this example is set to $a = 0.5$. This is a mixed mode K_{II} and K_{III} problem. In [200] the analytical solutions of SIFs given for the same

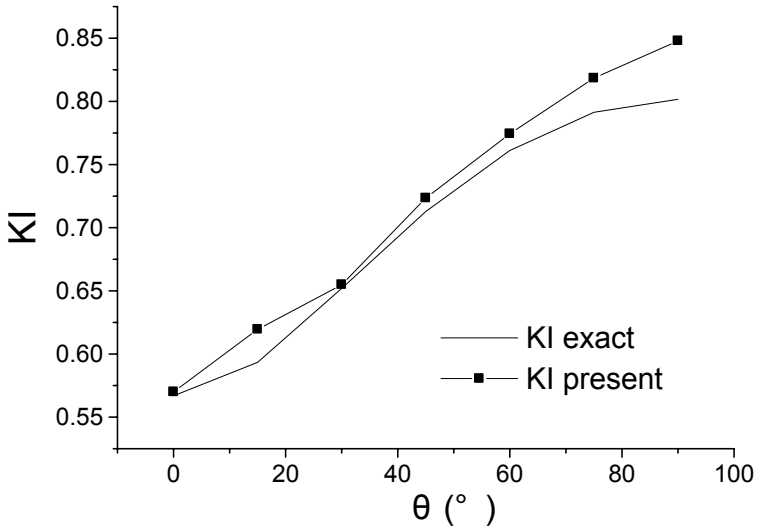


Figure 8.20: Plot of K_{I} results along the crack front (starting from an end of short axis aligned with positive x axis).

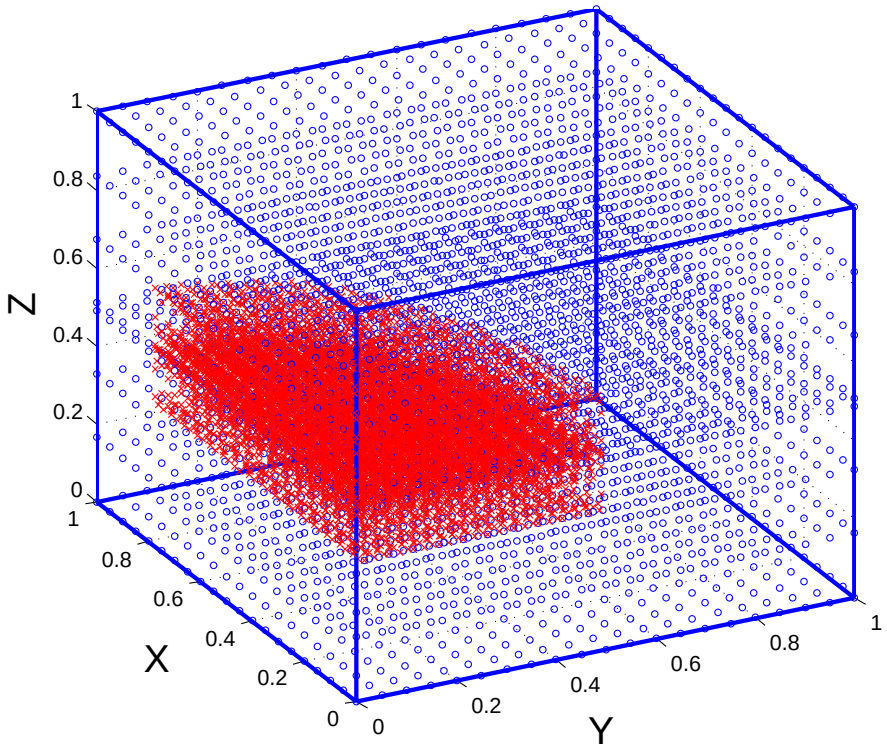
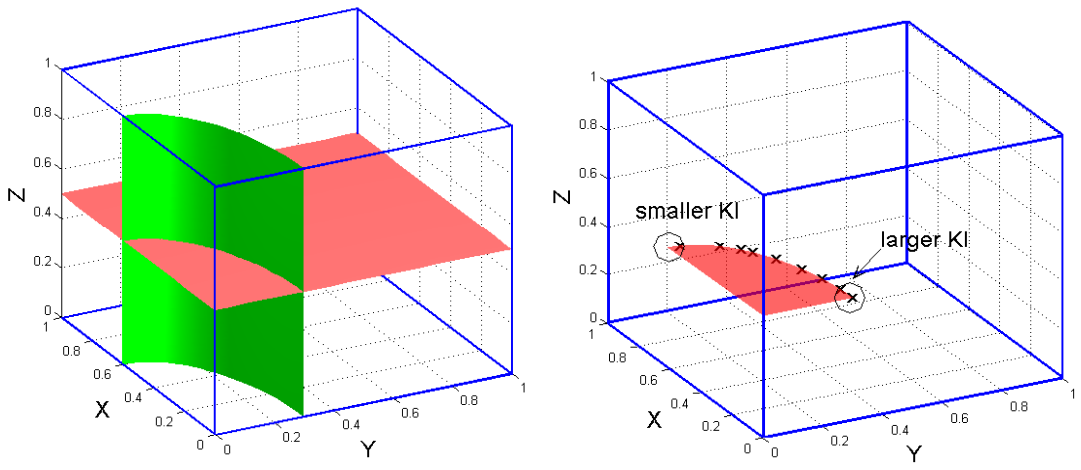


Figure 8.21: An elliptic crack and the banded region of integration points to introduce the displacement jump based on level sets.



(a) The plot of zero level sets ϕ and ψ

(b) Crack surface and front given by the level sets

Figure 8.22: The level set representation of the crack surface and the crack front point found by gradient projection method.

$\theta(\circ)$	0	15	30	45	60	75	90
FI(Ref.)	0.5839	0.6112	0.6716	0.7342	0.7840	0.8152	0.8257
KI (Ref.)	0.5669	0.5934	0.6520	0.7128	0.7611	0.7914	0.8016
KI (present)	0.5700	0.6195	0.6549	0.7235	0.7744	0.8184	0.8479
Error(%)	0.5543	4.4053	0.4451	1.5053	1.7452	3.4108	5.7759

Table 8.3: The SIF results of the elliptic crack under uniaxial tension.

type of problem but for an infinite domain are

$$K_{\mathrm{II}} = \frac{4\tau\sqrt{a}\cos\theta}{(2-\nu)\sqrt{\pi}} \tag{8.24}$$

$$K_{\mathrm{III}} = \frac{4\tau\sqrt{a}(1-\nu)\sin\theta}{(2-\nu)\sqrt{\pi}} \tag{8.25}$$

where θ is the angle of a unit normal at a point going along the crack front with respect to the positive x axis. The ratio between K_{II} and K_{III} is determined by Poisson's ratio as $\frac{K_{\mathrm{III}}}{K_{\mathrm{II}}} = 1-\nu$. For the specific example here $\nu = 0.3$ is used and hence $\frac{K_{\mathrm{III}}}{K_{\mathrm{II}}} = 0.7$. In Figure 8.24 the SIF results for K_{II} and K_{III} are plotted and the ratio between K_{II} and K_{III} is about 0.74. Also notable is that the K_{II} results here take opposite signs to those given in [189] (Figure 21 in [189]). The signs depends on the orientation of local coordinates and should be consistent. Here x_1 is the unit outer normal along crack front, x_2 is in the direction of the upper outer normal of crack surface and is aligned with positive z and x_3 is determined by the cross of x_1 and x_2. Of course x_2 and x_3 can alter their directions at the same time however they should be consistent with each other. The analytical solutions of K_{II} here take opposite signs to those given in [189] and in fact there is a sign error in this reference that signs of K_{II} are not consistent with K_{III}. This can be easily proven by the fact that at $\theta = 0^\circ$, the crack surface deforms in same direction of a positive pure mode II crack.

8.4.5 Inclined penny-shaped crack under tension

In the fifth example, the penny-shaped crack in the previous example is rotated along x axis 30° clockwise so it becomes inclined to the applied traction as shown in Figure 8.25. The level sets description of the initial crack surface are plotted in 8.27. This is a mixed mode crack problem involving all modes, unlike any of the previous examples. Analytical solutions for an infinite domain containing the crack

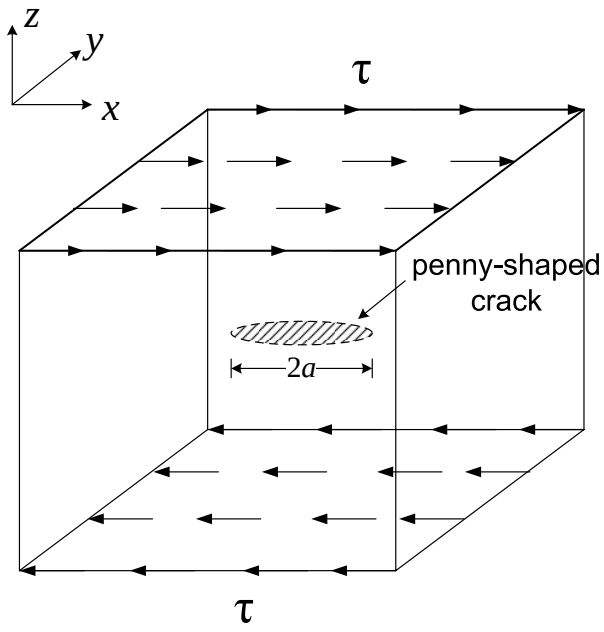


Figure 8.23: The geometry and boundary conditions of the penny-shaped crack subjected to pure shear.

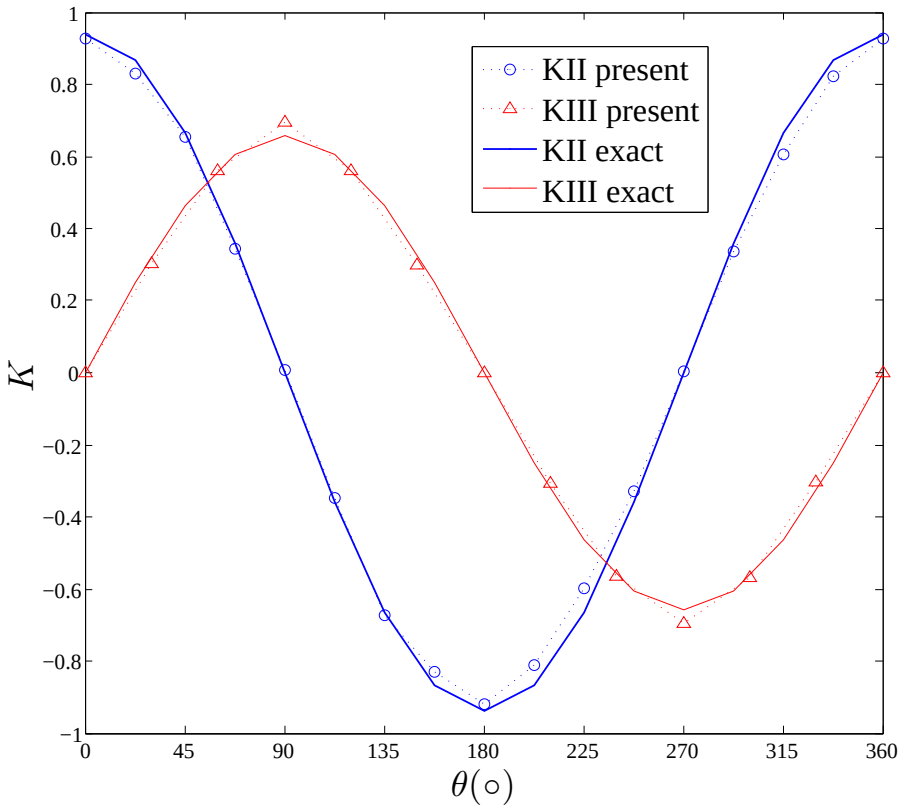


Figure 8.24: The SIF results of the penny-shaped crack subjected to pure shear.

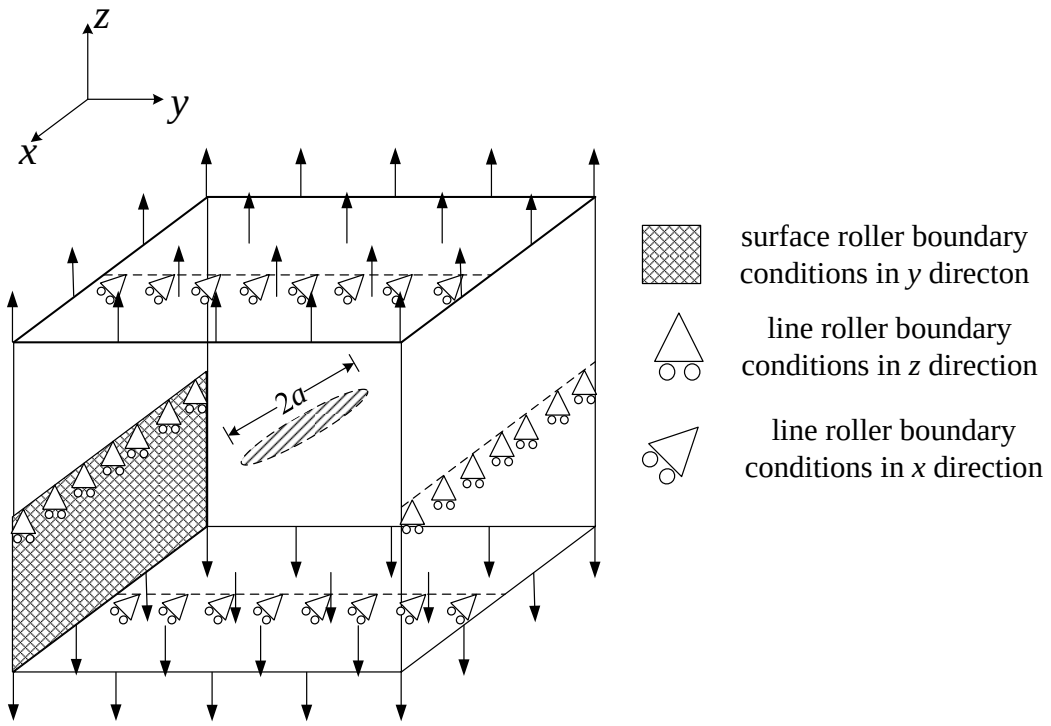


Figure 8.25: The geometry and boundary conditions of the inclined penny-shaped crack under uniaxial tension.

are given in [200] as

$$K_{\mathrm{I}} = 2\sqrt{\frac{a}{\pi}}\cos^2\beta \tag{8.26}$$

$$K_{\mathrm{II}} = \frac{4}{2-\nu}\sqrt{\frac{a}{\pi}}\sin\beta\cos\beta\sin\theta \tag{8.27}$$

$$K_{\mathrm{III}} = -\frac{4(1-\nu)}{2-\nu}\sqrt{\frac{a}{\pi}}\sin\beta\cos\beta\cos\theta \tag{8.28}$$

where β is the angle of the inclined crack surface with respect to the loading direction and in this particular example here $\beta = 30^\circ$ is used, θ is defined as in Equation (8.21), and a is the radius of crack ($a = 0.5$ here). Analytical solutions indicate that for a given crack geometry and loading magnitude K_{I} is constant, K_{II} follows a sine rule and K_{III} follows a cosine rule. The relation between K_{II} and K_{III} is opposite to the pure shear example. This can be easily understood in that the projection of traction onto the crack surface as shear stress is opposite in direction to the pure shear problem. In this example almost the same number of integration points are used as in the previous example and the integration points near the crack front are refined locally by the PU integration based on level sets as shown in Figure 8.26.

The SIF results for the three crack modes given by the new method are plotted and compared with the analytical solutions in Figure 8.28 showing fairly good agreement. According to the analytical solutions K_{I} should be constant along the crack front. However in Figure 8.28 the numerical solutions for K_{I} is not exactly constant but has two "bumps" at about 90° and 270° at the crack front. The reason for this is the analytical solutions assume an infinite domain while the present example has a finite domain so that the two inclined ends which are closer to the boundary will include a greater boundary effect. These points correspond to θ of 90° and 270°. The crack geometry at the second step of propagation is shown in Figure 8.29. It can be seen the increment of crack surface is tilted at the two ends and is starting to become perpendicular to the loading direction. If the angle of the new crack surface is measured ($\nabla\phi$ along the crack front), it is found not to be exactly perpendicular to the traction but at an angle of 82°. In a third step of calculation, the addition of jump nodes based on level sets is shown in Figure 8.31. Results show K_{I} is found remain almost constant along the crack front and the mean K_{I} increases to 0.69 while the magnitude of K_{II} is about 0.1 and K_{III} is almost zero. The crack surface therefore becomes more perpendicular to the traction and the crack surface at third step of propagation is shown in Figure 8.30. This agrees with the observations of the same example in [108]. The deformation near crack surface at the third step is plotted from various view aspects in Figure 8.4.5.

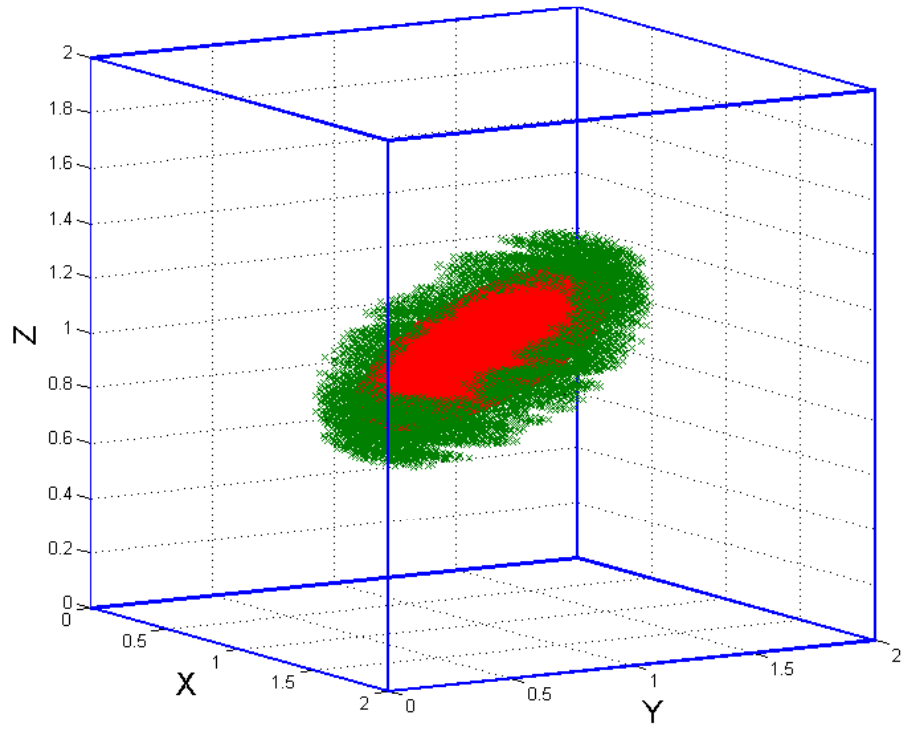


Figure 8.26: The geometry and boundary conditions of the inclined penny-shaped crack under uniaxial tension.

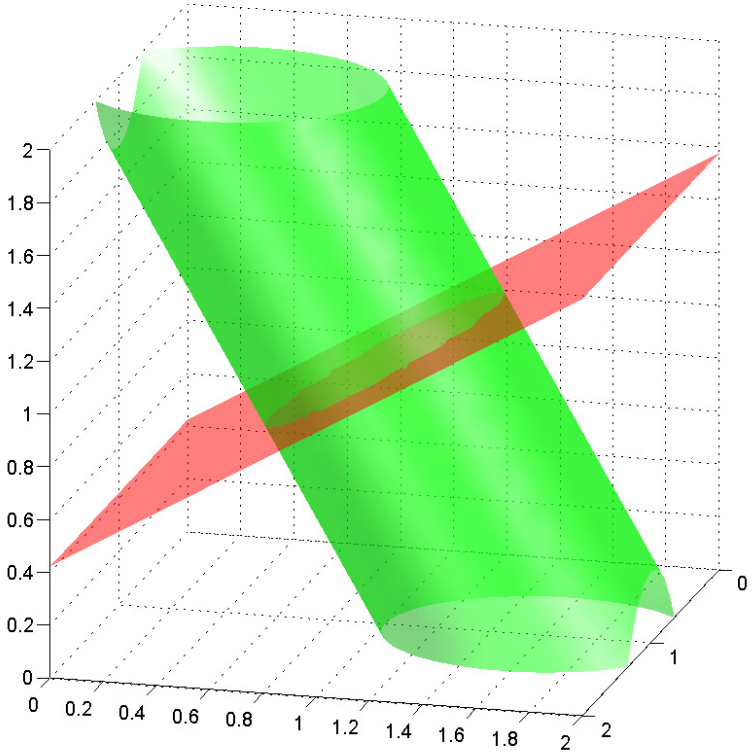


Figure 8.27: The level sets description of the inclined penny-shaped crack under uniaxial tension.

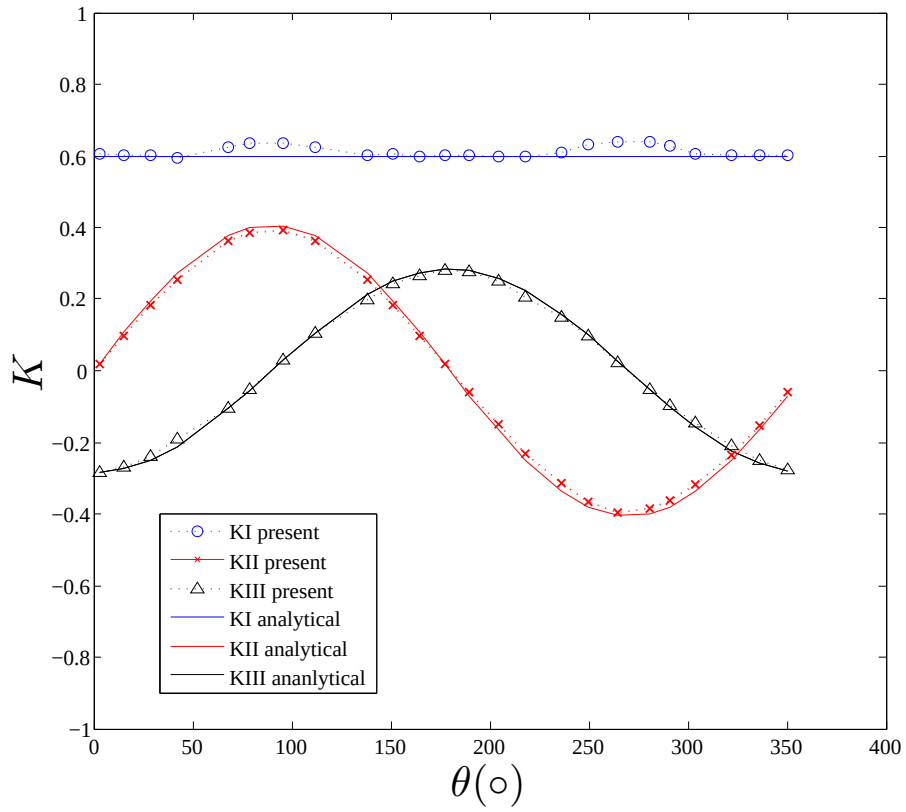
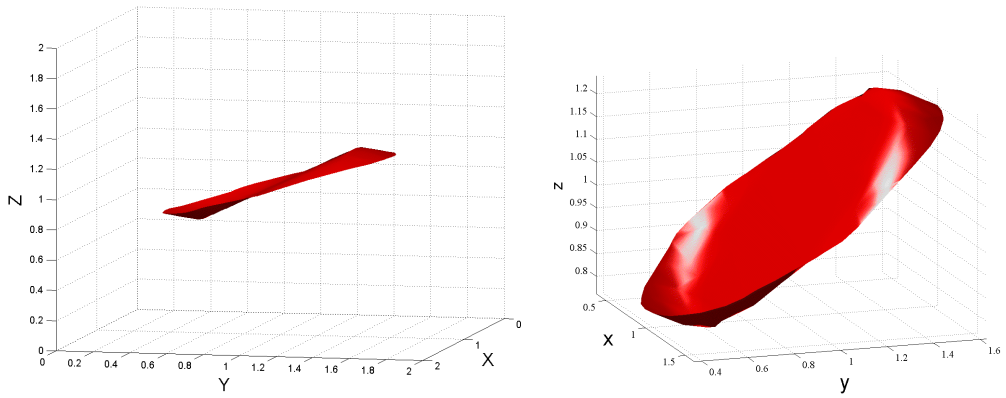
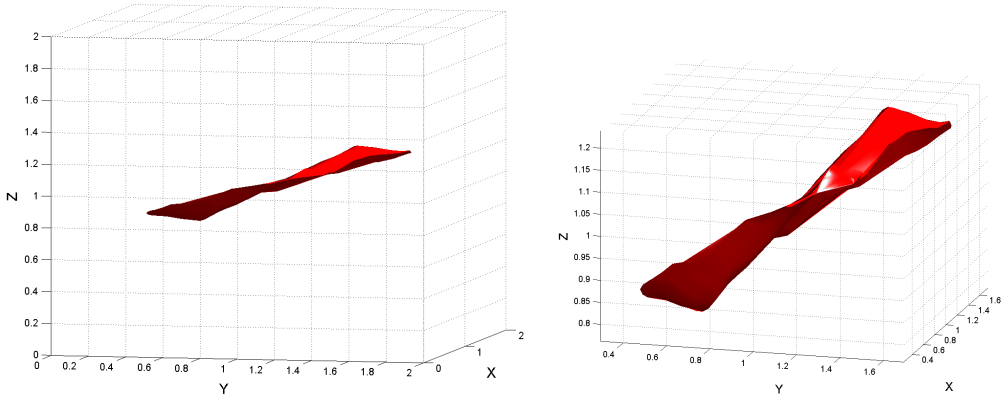


Figure 8.28: The SIF results of the inclined penny-shaped crack under uniaxial tension.



(a) (b)

Figure 8.29: The crack surface of inclined penny-shaped crack at second step of propagation.



(a) (b)

Figure 8.30: The crack surface of inclined penny-shaped crack at third step of propagation.

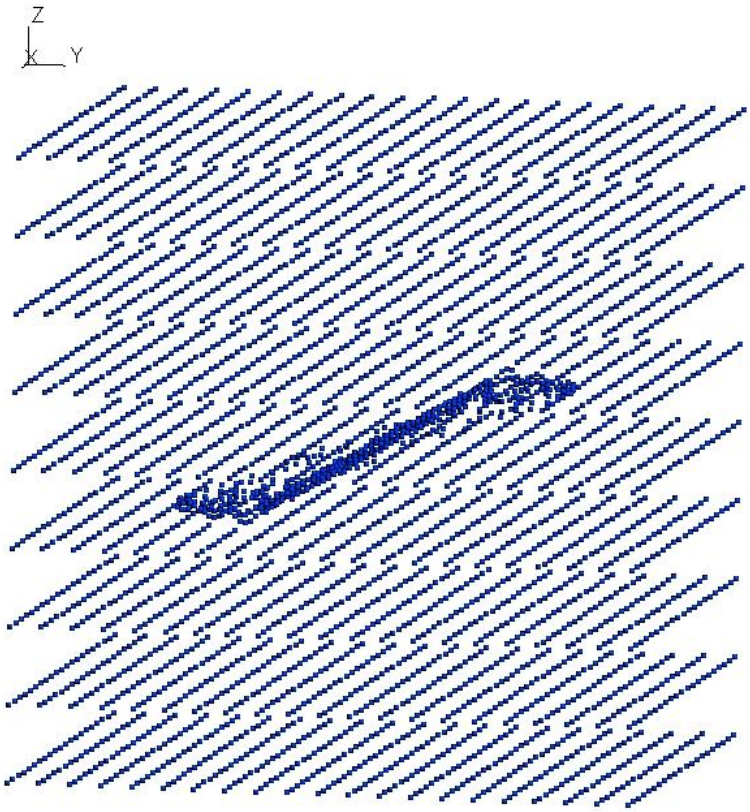
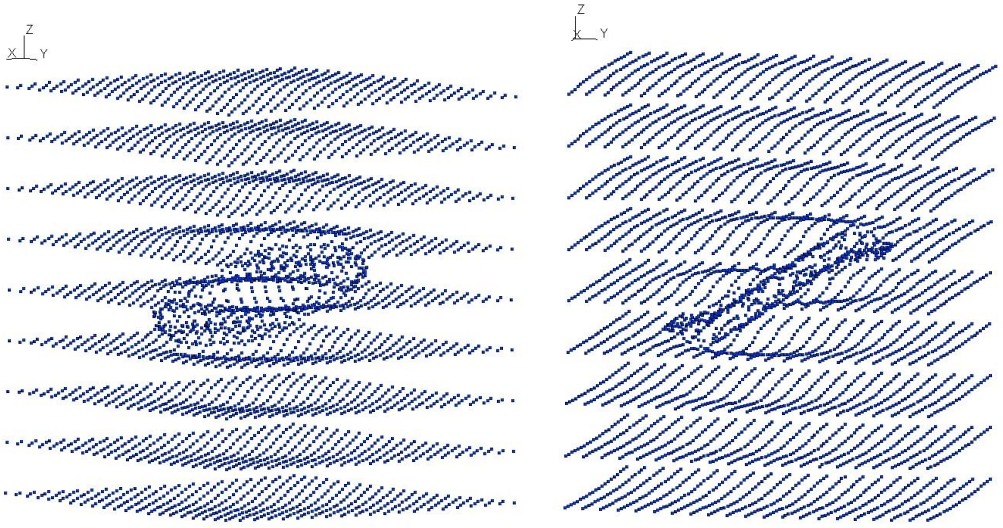
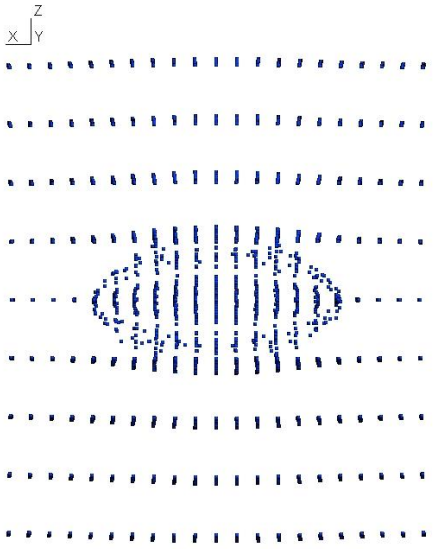


Figure 8.31: The nodal arrangement based on level sets for inclined penny-shaped crack at third step of propagation.

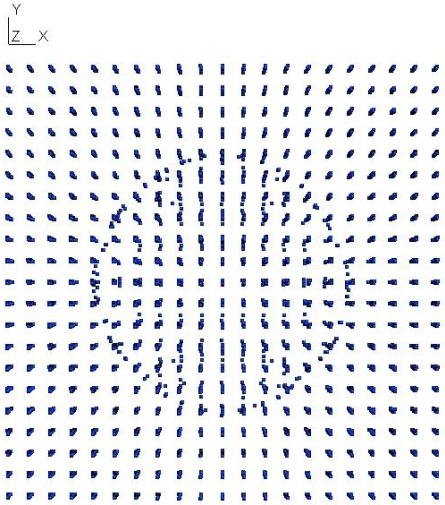


(a) xy view 1

(b) xy view 2



(c) y view



(d) z view

Figure 8.32: The deformation of an inclined penny-shaped crack after two steps of crack propagation.

8.5 Summary

In this chapter a new 3D fracture modelling framework is developped coupling the EFGM with level sets and brining together the bulk of ideas developed in the previous seven chapters. The EFGM is used for stress analysis and level sets are used to describe and capture the crack geometry as it evolves. A structured grid is used for the level sets domain which is decoupled from the meshless nodal arrangement. This feature removes the potential crack shifting problem when level sets are interpolated by the EFGM shape functions. The use of a structured grid also allows the algorithms and data storage of level sets to be vectorized, which is computationally optimal. The displacement jump near a crack is introduced using the simplest visibility criterion, which avoids the difficulty of defining the diffraction path when using the diffraction rule. The problem of spurious crack extension with the visibility criterion is fixed by using a tying procedure along crack front, which is extended from 2D. The formulation is based on level sets which makes it simple to implement and has a general formulation for various crack geometry. Apart from representing crack geometry, level sets are explored to provide new benefits such as refining the meshless model and reducing the computational cost. Examples are jump nodes aligned with crack surfaces and integration refinement near crack front which can be simply implemented with level sets. The framework is validated by a number of 3D crack examples against reference solutions and has shown its feasibility in 3D fracture modelling. An interesting finding in the results is, without enrichment the straight crack front tend to be of lower accuracy than those for curved crack fronts. Though the results of the examples presented here are not completely accurate they show the feasibility of the present framework.

Chapter 9

Conclusions and recommendations for future work

9.1 Conclusions

Modelling cracks and their propagation in 3D is an ongoing research topic in computational mechanics. The challenge in this topic is in finding an efficient and accurate numerical method that can accommodate arbitrary evolving cracks. In this thesis meshless methods are proposed as the basis for a new numerical model. At the start of the thesis various issues relating to meshless methods in general are dealt with although the vision is always on the 3D fracture modelling developed later. Studies are presented concerning the solution accuracy, error control, modified weak forms and implementation. One major focus of the thesis is the innovative use of level sets in 3D fracture modelling with the EFGM which has been explored in a number of different ways. Difficulties related to accurate jump enrichment in the EFGM and advancement of level sets are highlighted and solved by the developed methods. A new framework is proposed bringing all together these aspects coupling the EFGM and LSM for 3D crack modelling. The developments in this thesis that are regarded as new are as follows:

1. The derivation of the Nitsche-like modified weak form for elasticity which overcomes the drawbacks of currently used methods in dealing with the essential boundary conditions.

2. Efficient data storage and algorithms are developed especially for the meshless methods, and the idea of future parallel computation is attempted in the data structure and algorithms which is shown to reduce the computational cost in 3D problems.

3. A new formulation for determining shape functions and their derivatives using orthogonal basis functions is proposed, which preserves some positive features and avoids inaccuracies of a previously developed method.

4. Insights on error control and convergence of meshless methods are provided through the analysis of the formulations. Reasons for inefficiency in attempts of doing adaptive analysis in some currently available methods are revealed.

5. New method to introduce the displacement jump based on the level sets in the EFGM for arbitrary crack geometry is presented.

6. A cheap and simple tying procedure for closing the crack tip is proposed which enables the use of the visibility criterion for introducing the displacement jump while maintain the results accuracies.

7. The use of an independent structured level set grid with the EFGM for 3D fracture modelling which enables the vectorization of level set data storage and computation is presented.

8. A gradient projection method is developed for locating the crack front and extending the velocity from the crack front to the entire level set domain without solving PDEs.

9. Refinement of integration points near a crack front based on partition of unity and level sets is proposed.

10. Formulations for level sets for confining the jump enrichment across a crack surface and the addition of jump nodes on either sides of a crack are proposed.

9.2 Recommendations for future work

There are many interesting ideas that could develop from the work presented in this thesis:

- Further work on adaptive analysis of meshless methods. Currently available methods in the literature were found not effective for fracture modelling. There is a research gap in development a robust adaptive analysis scheme in meshless methods for fracture modelling. The recovery-based error estimation developed in the XFEM [199, 208, 209] and goal-oriented error estimation [210–212] developed in the FEM and the XFEM could be borrowed and developed in the context of meshless methods.

- Further testing of the Nitsche-like method. It will be interesting to do more tests and compare the consistency on an essential boundary using the Nitsche-like method with other methods such as penalty method, negative penalty method [213] and Lagrange multiplier method especially in dealing with surface boundary conditions in 3D. Another interesting test is to see whether the boundary locking problem appearing in the penalty method [64] will also appear in the Nitsche-like method.

- Studies on the link between the tying scheme and nonlocal gradient elasticity. In the tying scheme proposed in this thesis, the parameter β plays the role of controlling the crack opening ahead of the crack tip which is similar to the "internal scale length" of a material microstructure used in non-local gradient elasticity [201, 202]. It would be interesting to find the underlying mechanical links between these two parameters.

- Testing the 3D framework on more challenging and real problems. Examples in this thesis are all academic problems with reference or analytical solutions. The present framework must be tested by more challenging real problems such as brittle crack propagation in rock slope and fatigue crack propagation process in steel structures.

- Parallel computation for a large number of cracks. An obvious improvement of level sets is to use the localized level set method [186] where the level set domain is constructed in a narrow band surrounding the crack. The method is faster in computation and is ideal for large problems of many cracks. Parallel data storage and algorithms attempted in this thesis can be further developed using this method.

- Reconstruction and visualization of damage or failures of material using level sets. Level sets are used in this thesis to represent an explicit crack and capture crack evolution. For inverse problems, level sets also can be used to recover the geometry of defect or crack such as in boundary electrical measurement [214].

- Applying the present method in multiscale modelling at a certain level. The FEM or XFEM is often used in multiscale modelling [215]. In molecular or nano scale modelling such as DNA molecules [216, 217] and nano tubes [218, 219], meshless methods have shown particular advantages in dealing with moving interfaces and large deformations. In microscale crack modelling, meshless methods have also shown advantages over the FEM in calculating the material forces where "smoothness" of shape functions are needed [220]. It will be an interesting application to use the developed framework at a certain material scale.

Bibliography

[1] R. Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of American Mathematical Society*, 49:1–23, 1943.

[2] G.R. Johnson, R.A. Stryk, and S.R. Beissel. SPH for high velocity impact computation. *Computer Methods in Applied Mechancis and Engineering*, 139: 347–373, 1996.

[3] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[4] T. Rabczuk and T. Belytschko. Adaptivity for structured meshfree particle methods in 2D and 3D. *International Journal for Numerical Methods in Engineering*, 63:1559–1582, 2005.

[5] B. Muravin and E. Turkel. Multiple crack weight for solution of multiple interacting cracks by meshless numerical methods. *International Journal for Numerical Methods in Engineering*, 67:1146–1159, 2006.

[6] T. Kundu. *Fundamentals of fracture mechanics*. CRC Press Taylor & Francis Group Boca Raton, 2008.

[7] O. Huber, J. Nickel, and G. Kuhn. On the decomposition of the J-integral for 3d crack problems. *International Journal of Fracture*, 64:339–348, 1993.

[8] R.W. Clough and E. Wilson. Early finite element research at Berkeley. *Presentation on the Fifth U.S. National Conference on Computational Mechanics, August 4-6*, 1999.

[9] M.J. Turner, R.W. Clough, R.W. Martin, and L.J. Topp. Stiffness and deflection analysis of complex structures. *Journal of the Aeronautical Sciences*, 23: 805–823, 1956.

[10] Y. Zheng. Enabling technologies for large-scale multidisciplinary simulations, semi-plenary speech. In *9th World Congress on Computational Mechanics and 4th Asian Pacific Congress on Computational Mechanics, Sydney, Australia*, 2010.

[11] S.R. Idelsohn, E. Oñate, N. Calvo, and F. Del Pin. The meshless finite element method. *International Journal for Numerical Methods in Engineering*, 58:893–912, 2003.

[12] O.C. Zienkiewicz, M.S. Huang, and M. Pastor. Localization problems in plasticity using finite-elements with adaptive remeshing. *International Journal for Numerical and Analytical Methods in Geomechanics*, 19:127–148, 1995.

[13] J.F. Remacle, C. Geuzaine, G. Compere, and E. Marchandise. High-quality surface remeshing using harmonic maps. *International Journal for Numerical Methods in Engineering*, 83:403–425, 2010.

[14] A.R. Maligno, S. Rajaratnam, S.B. Leen, and E.J. Williams. A three-dimensional (3D) numerical study of fatigue crack growth using remeshing techniques. *Engineering Fracture Mechanics*, 77:94–111, 2010.

[15] L.B. Lucy. Numerical approach to testing of fission hypothesis. *Astronomical Journal*, 82:1013–1024, 1977.

[16] R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics - theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.

[17] J.J. Monaghan. Why particle methods work. *SIAM Journal on Scientific and Statistical Computing*, 3:422–433, 1982.

[18] J.J. Monaghan and R.A. Gingold. Shock simulation by the paticle method (SPH). *Journal of Computational Physics*, 52:374–389, 1983.

[19] R.A. Gingold and J.J. Monaghan. Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*, 46: 429–453, 1982.

[20] C. Federrath, R. Banerjee, P.C. Clark, and R.S. Klessen. Modeling collapse and accretion in turbulent gas clouds: implementation and comparison of sink particles in AMR and SPH. *Astrophysical Journal*, 713:269–290, 2010.

[21] J.W. Swegle and S.W. Attaway. On the feasibility of using smoothed hydrodynamics paticle methods in hydrodynamics. *Computaional Mechanics*, 17: 151–168, 1995.

[22] G.R. Johnson and S.R. Beissel. Normalized smoothing functions for SPH impact computations. *International Journal for Numerical Methods in Engineering*, 39:2725–2741, 1996.

[23] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, 47:1189–1214, 2000.

[24] J.W. Swegle, D.L. Hicks, and S.W. Attaway. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116:123–134, 1995.

[25] C.T. Dyka. Addressing tension instability in SPH methods. Technical report, NLR/MR/6384, 1994.

[26] M.B. Liu and G. R. Liu. Smoothed particle hydrodynamics (SPH): an overview and recent developments. *Archives of Computational Methods in Engineering*, 17:25–76, 2010.

[27] P. Lancaster and K. Salkauskas. Surfaces generated by moving least-squares methods. *Mathematics of Computation*, 37:141–158, 1981.

[28] B. Nayroles, G. Touzot, and P. Villon. The diffuse elements method. *Comptes Rendus de L'Academie des Sciences Serie II*, 313:133–138, 1991.

[29] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics*, 10: 307–318, 1992.

[30] W.K. Liu, S. Jun, and Y.F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids*, 20:1081–1106, 1995.

[31] E. Oñate, S. Idelsohn, O.C. Zienkiewicz, and R.L. Taylor. A finite point method in computational mechanics. applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, 39: 3839–3866, 1996.

[32] J.M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[33] I. Babuška, U. Banerjee, and J.E. Osborn. Meshless and generalized finite element methods: A survey of some major results. *Lecture notes in computational science and engineering: Meshfree Methods for Partial Equations, Springer-Verlag Berlin*, 26:1–20, 2003.

[34] T. L. Zhu, J. D. Zhang, and S. N. Atluri. A meshless numerical method based on the local boundary integral equation (LBIE) to solve linear and non-linear boundary value problems. *Engineering Analysis With Boundary Elements*, 23: 375–389, 1999.

[35] S.N. Atluri, H.-G. Kim, and J.Y. Cho. A critical assessment of the truly meshless local Petrov-Galerkin (MLPG) and Local Boundary Integral Equation (LBIE) methods. *Computational Mechanics*, 24:348–372, 1999.

[36] S.N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22:117–127, 1998.

[37] S.N. Atluri and T. Zhu. New concepts in meshless methods. *International Journal for Numerical Methods in Engineering*, 47:537–556, 2000.

[38] J. Braun and M. Sambridge. A numerical-method for solving partial-differential equations on highly irregular evolving grids. *Nature*, 376:655–660, 1995.

[39] N. Sukumar, B. Moran, A.Y. Semenov, and V.V. Belikov. Natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering*, 50:1–27, 2001.

[40] G.R. Liu and Y.T. Gu. A point interpolation method for two-dimensional solids. *International Journal for Numerical Methods in Engineering*, 50:937–951, 2001.

[41] G.R. Liu and Z.H. Tu. An adaptive procedure based on background cells for meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 191:1923–1943, 2002.

[42] J.G. Wang and G.R. Liu. A point interpolation meshless method based on radial basis functions. *International Journal for Numerical Methods in Engineering*, 54:1623–1648, 2002.

[43] G.R. Liu, Y.T. Gu, and K.Y. Dai. Assessment and applications of point interpolation methods for computational mechanics. *International Journal for Numerical Methods in Engineering*, 59:1373–1397, 2004.

[44] C.A. Duarte and J.T. Oden. H-p clouds an h-p meshless method. *Numerical methods for Partial Diffrential Equations*, 12:673–705, 1996.

[45] C.A. Duarte and J.T. Oden. An h-p adaptive method using clouds. 139: 237–262, 1996.

[46] J.T. Oden, C.A. Durate, and O.C. Zienkiewicz. A new cloud-based hp finite element method. *Computer Methods in Applied Mechanics and Engineering*, 153:117–126, 1998.

[47] J. N. Fang, A. Parriaux, M. Rentschler, and C. Ancey. Improved SPH methods for simulating free surface flows of viscous fluids. *Applied Numerical Mathematics*, 59:251–271, 2009.

[48] H.T. Yang and Y.Q. He. Solving heat transfer problems with phase change via smoothed effective heat capacity and element-free Galerkin methods. *International Communications in Heat and Mass Transfer*, 37:385–392, 2010.

[49] J.Y. Tan and L.H. Liu. Inverse geometry design of radiating enclosure filled with participating media using meshless method. *Numerical Heat Transfer Part A-Applications*, 56:132–152, 2009.

[50] S. F. Li and W. K. Liu. Numerical simulations of strain localization in inelastic solids using mesh-free methods. *International Journal for Numerical Methods in Engineering*, 48:1285–1309, 2000.

[51] S.F. Li, W.K. Liu, A.J. Rosakis, T. Belytschko, and W. Hao. Mesh-free Galerkin simulations of dynamic shear band propagation and failure mode transition. *International Journal of Solids and Structures*, 39:1213–1240, 2002.

[52] H. Askes. *Advanced spatial discretisation strategies for localised failure: mesh adaptivity and meshless methods*. PhD thesis, Delft University of Technology, Netherlands, 2000.

[53] H. Askes and E.C. Aifantis. Numerical modeling of size effects with gradient elasticity - formulation, meshless discretization and examples. *International Journal of Fracture*, 117:347–358, 2002.

[54] J. Sladek, V. Sladek, J. Krivacek, P.H. Wen, and C. Zhang. Meshless local Petrov - Galerkin (MLPG) method for Reissner c Mindlin plates under dynamic load. *Computer Methods in Applied Mechanics and Engineering*, 196: 2681–2691, 2007.

[55] P. Krysl and T. Belytschko. Analysis of thin plates by the element-free Galerkin method. *Computational Mechanics*, 17:26–35, 1996.

[56] J.S. Chen, C.H. Pan, C.T. Wu, and W.K. Liu. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering*, 139:195–227, 1996.

[57] C.V. Le, H. Askes, and M. Gilbert. A novel numerical procedure for limit analysis of plates: adaptive EFG method combined with SCP. In C. Sansour, editor, *Proc. 17th UK Conference on Computational Mechanics*, pages 291–294, Nottingham, UK, 2009.

[58] C.V. Le, M. Gilbert, and H. Askes. Limit analysis of plates using the EFG method and second-order cone programming. *International Journal for Numerical Methods in Engineering*, 78:1532–1552, 2009.

[59] S. Li, W. Hao, and W.K. Liu. Numerical simulations of large deformation of thin shell structures using meshfree methods. *Computational Mechanics*, 25: 102–116, 2000.

[60] J.S. Chen, C. Pan, C.M.O.L. Roque, and H.P. Wang. A lagrangian reproducing kernel particle method for metal forming analysis. *Computational Mechanics*, 22:289–307, 1998.

[61] I. Alfaro, J. Yvonnet, E. Cueto, F. Chinesta, and M. Doblare. Meshless methods with application to metal forming. *Computer Methods in Applied Mechanics and Engineering*, 195:6661–6675, 2006.

[62] R. Trobec, M. Sterk, and B. Robic. Computational complexity and parallelization of the meshless local Petrov-Galerkin method. *Computers & Structures*, 87:81–90, 2009.

[63] S.R. Idelsohn and E. Oñate. To mesh or not to mesh. That is the question... *Computer Methods in Applied Mechanics and Engineering*, 195:4681–4696, 2006.

[64] J.Y. Cho, Y.M. Song, and Y.H. Choi. Boundary locking induced by penalty enforcement of essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 197:1167–1183, 2008.

[65] H. Askes, R. de Borst, and O. Heeres. Conditions for locking-free elasto-plastic analyses in the Element-Free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 173:99–109, 1999.

[66] C.K. Lee and C.E. Zhou. On error estimation and adaptive refinement for element free Galerkin method part I: stress recovery and a posteriori error estimation. *Computers & Structures*, 82:413–428, 2004.

[67] C.K. Lee and C.E. Zhou. On error estimation and adaptive refinement for element free Galerkin method part II: adaptive refinement. *Computers & Structures*, 82:429–443, 2004.

[68] H.J. Chung and T. Belytschko. An error estimate in the EFG method. *Computational Mechanics*, 21:91–100, 1998.

[69] C.K. Lee and Y.Y. Shuai. An automatic adaptive refinement procedure for the reproducing kernel particle method. part i: Stress recovery and a posteriori error estimation. *Computational Mechanics*, 40:399–413, 2007.

[70] R. Rossi and M.K. Alves. An h-adaptive modified element-free Galerkin method. *European Journal of Mechanics: A:Solids*, 24:782–799, 2005.

[71] H.S. Lu and J.S. Chen. *Adaptive Galerkin paticle method. Method Meshfree methods for partial differential equations. Lecture notes in computational science and engineering.* Springer, Berlin, 2003.

[72] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a-posteriori error-estimates. 1: the recovery technique. *International Journal for Numerical Methods in Engineering*, 33:1331–1364, 1992.

[73] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a-posteriori error-estimates. 2: error-estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33:1365–1382, 1992.

[74] Y. Krongauz and T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131:133–145, 1996.

[75] Y.T. Gu and G.R. Liu. Meshless local Petrov-Galerkin (MLPG) method in combination with finite element and boundary element approaches. *Computational Mechanics*, 26:536– 546, 2000.

[76] Y.T. Gu and G.R. Liu. A coupled element free Galerkin/boundary element method for stress analysis of two-dimensional solids. *Computational Mechanics*, 27:188–198, 2001.

[77] A.J. Deeks and C.E. Augarde. A hybrid meshless local Petrov-Galerkin method for unbounded domains. *Computer Methods in Applied Mechanics and Engineering*, 196:843–852, 2007.

[78] J.R. Lund and J.P. Byrne. Leonardo da Vinci's tensile strength tests: Implications for the discovery of engineering mechanics. *Civil Engineering and Environmental Systems*, 18:243–250, 2001.

[79] T.L. Anderson. *Fracture mechanics : fundamentals and applications*. CRC Press, 1995.

[80] G.R. Irwin. Fracutre dynamics. *Fracturing of metals, American Society for Metals,*, Clevand:147–166, 1948.

[81] A.A. Griffith. Phenomena of rupture and flow in solids. *Philosophical Transactions of the Royal Society of London, Series A*, 221:163–198, 1921.

[82] H.M. Westergaard. Bearing pressure and cracks. *Journal of Applied Mechanics*, 6:49–53, 1939.

[83] M.L. Williams. On the stress distribution at the base of a stationary crack. *Journal of Applied Mechanics*, 24:109–114, 1957.

[84] A.A. Wells. The condition of fast fracture in aluminium alloys with particular references to Comet failures. *British Welding Association Report*, April, 1955.

[85] C.E. Inglis. Stress in a plate due to the presence of cracks and sharp corneres. *Transaction of theInstitue of Naval Architectures*, 55:219–241, 1913.

[86] G. Francfort and J.J. Marigo. Towards an energetic theory of brittle fracture. *Comptes Rendus Mecanique*, 330:225–233, 2002.

[87] G.A. Francfort and J.J. Marigo. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids*, 46:1319–1342, 1998.

[88] S.K. Chan, I.S. Tuba, and W.K. Wilson. On the finite elemnt method in linear fracture mechanics. *Engineering Fracture Mechanics*, 2:1–17, 1970.

[89] R.D. Henshell and K.G. Shaw. Crack tip finite elements are unnecessary. *International Journal for Numerical Methods in Engineering*, 9:495–507, 1975.

[90] R.S. Barsoum. On the use of isoparametric finite elements in linear fracture mechanics. *International Journal for Numerical Methods in Engineering*, 10: 25–37, 1976.

[91] R. Simpson. *Enrichment of the boundary element method through the partition of unity method for accurate stress analysis*. PhD thesis, Durham University, 2010.

[92] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999.

[93] N. Möes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999.

[94] T. Belytschko, R. Gracie, and G. Ventura. A review of extended/generalized finite element methods for material modeling. *Modelling and Simulation in Materials Science and Engineering*, 17:043001, 2009.

[95] J. Rannou, A. Gravouil, and M.C. Baietto-Dubourg. A local multigrid X-FEM strategy for 3-D crack propagation. *International Journal for Numerical Methods in Engineering*, 77:581–600, 2009.

[96] T. Belytschko, Y.Y. Lu, and L. Gu. Crack-propagation by element-free Galerkin methods. *Engineering Fracture Mechanics*, 51:295–315, 1995.

[97] N. Sukumar, B. Moran, T. Black, and T. Belytschko. An element-free Galerkin method for three-dimensional fracture mechanics. *Computational Mechanics*, 20:170–175, 1997.

[98] P. Krysl and T. Belytschko. Element-free Galerkin method: Convergence of the continuous and discontinuous shape functions. *Computer Methods in Applied Mechanics and Engineering*, 148:257–277, 1997.

[99] P. Krysl and T. Belytschko. The element free Galerkin method for dynamic propagation of arbitrary 3-D cracks. *International Journal for Numerical Methods in Engineering*, 44:767–800, 1999.

[100] T. Rabczuk and T. Belytschko. A three-dimensional large deformation meshfree method for arbitrary evolving cracks. *Computer Methods in Applied Mechanics and Engineering*, 196:2777–2799, 2007.

[101] T. Rabczuk, S. Bordas, and G. Zi. A three-dimensional meshfree method for continuous multiple-crack initiation, propagation and junction in statics and dynamics. *Computational Mechanics*, 40:473–495, 2007.

[102] T. Rabczuk, Bordas S., and G. Zi. On three-dimensional modelling of crack growth using partition of unity methods. *Computers & Structures*, Article in press:doi:10.1016/j.compstruc., 2008.

[103] T. Rabczuk, J.-H. Song, and T. Belytschko. Simulations of instability in dynamic fracture by the cracking particles method. *Engineering Fracture Mechanics*, 76:730–741, 2009.

[104] T. Rabczuk and G. Zi. A meshfree method based on the local partition of unity for cohesive cracks. *Computational Mechanics*, 39:743–760, 2007.

[105] G. Zi, T. Rabczuk, and W. Wall. Extended meshfree methods without branch enrichment for cohesive cracks. *Computational Mechanics*, 40:367–382, 2007.

[106] R. Brighenti. Application of the element-free Galerkin meshless method to 3-d fracture mechanics problems. *Engineering Fracture Mechanics*, 72:2808–2820, 2005.

[107] N. Sukumar, D.L. Chopp, E. Bechet, and N. Möes. Three-dimensional non-planar crack growth by a coupled extended finite element and fast marching method. *International Journal for Numerical Methods in Engineering*, 76: 727–748, 2008.

[108] M. Duflot. A meshless method with enriched weight functions for three-dimensional crack propagation. *International Journal for Numerical Methods in Engineering*, 65:1970–2006, 2006.

[109] J.A. Sethian. Fast marching methods. *SIAM Review*, 41:199–235, 1999.

[110] I.M. Mitchell. The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing*, 35:300–329, 2008.

[111] I.M. Mitchell. *Application of level set methods to control and reachability problems in continuous and hybird systems*. PhD thesis, Scientific Computing and Computational Mathematics, Stanford University, 2002.

[112] I.M. Mitchell. A toolbox of level set methods (version 1.1). Technical report, Department of Computer Science, University of British Columbia, Vancouver, BC, Canda, 2007.

[113] S. Bordas, T. Rabczuk, and G. Zi. 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, 2008.

[114] S. Bordas, G. Zi, and T. Rabczuk. Three-dimensional non-linear fracture mechanics by enriched meshfree methods without asymptotic enrichment. In A. Combescure, R. de Borst, and T. Belytschko, editors, *IUTAM Symposium on Discretization Methods for Evolving Discontinuities*, pages 21–36, 2007.

[115] M. Stolarska, D.L. Chopp, N. Möes, and T. Belytschko. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 51:943–960, 2001.

[116] A. Gravouil, N. Möes, and T. Belytschko. 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, 2002.

[117] N. Möes, A. Gravouil, and T. Belytschko. Non-planar 3D crack growth by the extended finite element and level sets - part I: Mechanical model. *International Journal for Numerical Methods in Engineering*, 53:2549–2568, 2002.

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

[119] G.R. Liu. *Meshfree methods: moving beyond the finite element method*. CRC Press LLC, Florida, 2003.

[120] T. Belytschko, Y.Y. Lu, L. Gu, and M. Tabbara. Element-free Galerkin methods for static and dynamic fracture. *International Journal for Numerical Methods in Engineering*, 32:2547–2570, 1995.

[121] J. Dolbow and T. Belytschko. An introdution to programming the meshless element-free Galerkin method. *Archives of Computational Methods in Engineering*, 5:207–241, 1998.

[122] S. Fernandez-Mendez and A. Huerta. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193:1257–1275, 2004.

[123] Y.C. Cai and H.H. Zhu. Direct imposition of essential boundary conditions and treatment of material discontinuities in the EFG method. *Computational Mechanics*, 34:330–338, 2004.

[124] J.S. Chen and H.P. Wang. New boundary condition treatments in meshfree computation of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 187:441–468, 2003.

[125] N. Sukumar. Construction of polygonal interpolants: a maximum entropy approach. *International Journal for Numerical Methods in Engineering*, 61: 2159–2181, 2004.

[126] Y.C. Cai, X.Y. Zhuang, and C. Augarde. A new partition of unity finite element free from the linear dependence problem and possessing the delta property. *Computer Methods in Applied Mechanics and Engineering*, 199: 1036–1043, 2010.

[127] T. Belytschko, D. Organ, and Y. Krongauz. A couple finite element-element free Galerkin method. *Computational Mechanics*, 17:186–195, 1995.

[128] S.P. Timoshenko and J.N. Goodier. *Theory of elasticity*. McGraw-Hill, New York, 1970.

[129] C. Lanczos. *The variational principles of mechanics*. Dover Publications, New York, 1986.

[130] K. Washizu. *Variational methods in elasticity and plasticity*. Pergamon, New York, 2nd edition, 1975.

[131] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen Mathematischen Seminar Universitat Hamburg*, 36:7–15, 1971.

[132] Y. Lu, T. Belytschko, and L. Gu. A new implementation of the element free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 113:397–414, 1994.

[133] S. Skatulla and C. Sansour. Essential boundary conditions in meshfree methods via a modified variational principle. *Computer Assisted Mechanics and Engineering Sciences*, 15:123–142, 2008.

[134] P.C. Hammer, O.J. Marlowe, and Stroud A.H. Numerical integration over simplexes and cones. *Mathematical Tables and Other Aids to Computation*, 10:130–137, 1956.

[135] S. Beissel and T. Belytschko. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139:49–74, 1996.

[136] J. Dolbow and T. Belytschko. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics*, 23:219–230, 1999.

[137] J.S. Chen, C.T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.

[138] T.P. Fries and T Belytschko. Convergence and stabilization of stress-point integration in mesh-free and particle methods. *International Journal for Numerical Methods in Engineering*, 74:1067–1087, 2008.

[139] M. Duflot and N.D. Hung. A truly meshless Galerkin method based on a moving least squares quadrature. *Communications in Numerical Methods in Engineering*, 18:441–449, 2002.

[140] Y. Liu and T. Belytschko. A new support integration scheme for the weak-form in mesh-free methods. *International Journal for Numerical Methods in Engineering*, 82:699–715, 2010.

[141] J.F. Remacle and C. Geuzaine. *Gmsh finite element grid generator*. Available at www.geuz.org/gmsh, 2008.

[142] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309–1331, 2009.

[143] M. Fleming. *The element-free Galerkin method for fatigue and quasi-static fracture*. PhD thesis, Northwestern University, USA, 1997.

[144] O.C. Zienkiewicz and K. Morgan. *Finite elements and approximations*. Dover Publications Inc., Mineola, New York, 2000.

[145] M. Ainsworth and J.T. Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142:1–88, 1997.

[146] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, sixth edition, 2005.

[147] L. Gavete, J.L. Cuesta, and A. Ruiz. A numerical comparison of two different approximations of the error in a meshless method. *European Journal of Mechanics: A: Solids*, 21:1037–1054, 2002.

[148] T.P. Fries and T. Belytschko. The intrinsic partition of unity method. *Computational Mechanics*, 40:803–814, 2007.

[149] M.G. Armentano. Error estimates in Sobolev spaces for moving least square approximations. *SIAM Journal on Numerical Analysis*, 39:38–51, 2001.

[150] X. Li and J. Zhu. A Galerkin boundary node method and its convergence analysis. *Journal of Computational and Applied Mathematics*, 230:314–328, 2009.

[151] C. Zuppa. Good quality point sets and error estimates for moving least square approximations. *Applied Numerical Mathematics*, 47:575–585, 2003.

[152] T. Belytschko, L. Gu, and Y.Y. Lu. Fracture and crack growth by element free Galerkin methods. *Modelling and Simulation in Materials Science and Engineering*, 2:519–534, 1994.

[153] D. Organ, M. Fleming, T. Terry, and T. Belytschko. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational Mechanics*, 18:225–235, 1996.

[154] T. Belytschko, G. Ventura, and J.X. Xu. New methods for discontinuity and crack modelling in EFG. *Lecture notes in computational science and engineering: Meshfree methods for partial differential equations*, 26:37–50, 2003.

[155] J. Planas and M. Elices. Asymptotic analysis of a cohesive crack: 2. influence of the softening curve. *International Journal of Fracture*, 64:221–237, 1993.

[156] C. Daux, N. Möes, J. Dolbow, N. Sukumar, and T. Belytschko. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 48:1741–1760, 2000.

[157] E. Béchet, H. Minnebot, N. Möes, and B. Burgardt. Improved implementation and robustness study of the X-FEM for stress analysis around cracks. *International Journal for Numerical Methods in Engineering*, 64:1033–1056, 2005.

[158] S.E. Benzley. Representation of singularities with isoparametric finite elements. *International Journal for Numerical Methods in Engineering*, 8:537–545, 1974.

[159] M. Fleming, Y.A. Chu, B. Moran, and T. Belytschko. Enriched element-free Galerkin methods for crack tip fields. *International Journal for Numerical Methods in Engineering*, 40:1483–1504, 1997.

[160] Y.C. Yoon, S.H. Lee, and T. Belytschko. Enriched meshfree collocation method with diffuse derivatives for elastic fracture. *Computers & Mathematics with Applications*, 51:1349–1366, 2006.

[161] Z. Zhang, K. M. Liew, Y.M. Cheng, and Y. Y. Lee. Analyzing 2D fracture problems with the improved element-free Galerkin method. *Engineering Analysis with Boundary Elements*, 32:241–250, 2008.

[162] M. Duflot and H. Nguyen-Dang. A meshless method with enriched weight functions for fatigue crack growth. *International Journal for Numerical Methods in Engineering*, 59:1945–1961, 2004.

[163] X. Zhuang and C. Augarde. The effects of nodal arrangement on solution accuracy in meshless methods. *Proc. 17th annual conference of Association of Computational Mechanics in Engineering, ACME, Nottingham, UK*, 2009.

[164] X. Zhuang and C. Augarde. Aspects of the use of orthogonal basis functions in the element-free Galerkin method. *International Journal for Numerical Methods in Engineering*, 81:366–380, 2010.

[165] J.R. Rice. A path independent integral and the approximate analysis of strain concentration by notches and cracks. *Journal of Applied Mechanics*, 35:379–386, 1968.

[166] F.H.K. Chen and R.T. Shield. Conservation laws in elasticity of J-integral type. *Zeitschrift fur Angewandte Mathematik und Physik*, 28:1–22, 1977.

[167] M. Stern, E.B. Becker, and R.S. Dunham. A contour integral computation of mixed-mode stress intensity factors. *International Journal of Fracture*, 12: 359–368, 1976.

[168] G.P. Nikishkov and S.N. Atluri. Calculation of fracture-mechanics parameters for an arbitrary 3-dimensional crack, by the equivalent domain integral method. *International Journal for Numerical Methods in Engineering*, 24: 1801–1821, 1987.

[169] M. Gosz and B. Moran. An interaction energy integral method for computation of mixed-mode stress intensity factors along non-planar crack fronts in three dimensions. *Engineering Fracture Mechanics*, 69:299–319, 2002.

[170] B. Moran and C.F. Shih. Crack tip and associated domain integrals from momentum and energy-balance. *Engineering Fracture Mechanics*, 27:615–642, 1987.

[171] C.F. Shih, B. Moran, and T. Nakamura. Energy-release rate along a 3-dimensional crack front in a thermally stressed body. *International Journal of Fracture*, 30:79–102, 1986.

[172] J.D. Eshelby. *Energy relations and the energy momentum tensor in continuum mechanics*, chapter in Elastic Behavior of Solids. McGraw-Hill, New York, 1970.

[173] R.H. Rigby and M.H. Aliabadi. Decomposition of the mixed-mode J-integral revisited. *International Journal of Solids and Structures*, 35:2073–2099, 1998.

[174] G.P. Cherepanov. *Mechanics of brittle fracture*. McGraw-Hill, New Yok, 1979.

[175] G.P. Cherepanov. The propagation of cracks in a continuum medium. *International Journal for Numerical Methods in Engineering*, 31:503–512, 1967.

[176] H. Kitagawa, H. Okamura, and H. Ishikawa. Application of J-integral to mixed-mode crack problems. *Transaction of JSME*, 760:46–48, 1976.

[177] A. Portela, M.H. Aliabadi, and D.P. Rooke. The dual boundary element method - effective implementation for crack problems. *International Journal for Numerical Methods in Engineering*, 33:1269–1287, 1992.

[178] P.H. Wen and M.H. Aliabadi. A contour integral for the evaluation of stress intesity factors. *Applied Mathematical Modelling*, 19:450–455, 1995.

[179] M. Gosz, J. Dolbow, and B. Moran. Domain integral for stress intensity factor computation along curved three-dimensional interface cracks. *International Journal of Solids and Structures*, 35:1763–1783, 1997.

[180] J.H. Chang and D.J. Wu. Stress intensity factor computation along a non-planar curved crack in three dimensions. *International Journal of Solids and Structures*, 44:371–386, 2007.

[181] K.B. Broberg. On crack paths. *Engineering fracture mechanics*, 28:663–679, 1987.

[182] P.S. Paris and M.P. Gomez. A rational analytic theory of fatigue. *The Trend in Engineering*, 13:9–14, 1961.

[183] B. Lawn. *Fracture of brittle solids*. Cambridge University Press, 2nd edition, 1993.

[184] J.A. Sethian. *Level sets methods and fast marching methods*. Cambridge University Press, Cambridge, 1996.

[185] A.J.M. Spencer. *Continuum mechanics*. Dover, New York, 2003.

[186] D.P. Peng, B. Merriman, S. Osher, H.K. Zhao, and M.J. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.

[187] D. Adalsteinsson and J.A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148:2–22, 1999.

[188] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer, New York, 2003.

[189] M. Duflot. A study of the representation of cracks with level sets. *International Journal for Numerical Methods in Engineering*, 70:1261–1302, 2007.

[190] P. Burchard, L. T. Cheng, B. Merriman, and S. Osher. Motion of curves in three spatial dimensions using a level set approach. *Journal of Computational Physics*, 170:720–741, 2001.

[191] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[192] G. Russoa and P. Smerekab. A remark on computing distance functions. *Journal of Computational Physics*, 163:51–67, 2000.

[193] X. Zhuang, C.E. Augarde, and S.P.A. Bordas. Accurate fracture modelling using meshless methods and level sets: formulation and 2D modelling. *International Journal for Numerical Methods in Engineering*, Accepted, 2010.

[194] D. Eberly. Distance from a point to an ellipse in 2D. www.geometrictools.com, 2008.

[195] P.S. Heckbert. *Graphics gems IV*. AP Professional (Academic Press), Boston, 1994.

[196] D. P. Rooke and D.J. Cartwright. *Compendium of stress intensity factors*. HMSO, Great Britain. Ministry of Defence. Procurement Executive London., 1976.

[197] G. Zi and T. Belytschko. New crack-tip elements for XFEM and applications to cohesive cracks. *International Journal for Numerical Methods in Engineering*, 57:2221–2240, 2003.

[198] G.T. Houlsby, G. Liu, and C.E. Augarde. A tying scheme for imposing displacement constraints in finite element analysis. *Communications in Numerical Methods in Engineering*, 16:721–732, 2000.

[199] S. Bordas and M. Duflot. Derivative recovery and a posteriori error estimate for extended finite elements. *Computer Methods in Applied Mechanics and Engineering*, 196:3381–3399, 2007.

[200] Y. Murakami. *The stress intensity factors handbook*. Pergamon Press, Oxford, 1987.

[201] H. Askes and I.M. Gitman. Non-singular stresses in gradient elasticity at bimaterial interface with transverse crack. *International Journal of Fracture*, 156:217–222, 2009.

[202] A.C. Eringen, C.G. Speziale, and B.S. Kim. Crack-tip problem in non-local elasticity. *Journal of the Mechanics and Physics of Solids*, 25:339–355, 1977.

[203] X.D. Liu, S. Osher, and T. Chan. Weighted essentially nonoscillatory schemes. *Journal of Computational Physics*, 115:200–212, 1994.

[204] T.Y. Hou, Z.L. Li, S. Osher, and H.K. Zhao. A hybrid method for moving interface problems with application to the hele-shaw flow. *Journal of Computational Physics*, 134:236–252, 1997.

[205] S.H. Lee and Y.C. Yoon. An improved crack analysis technique by element-free Galerkin method with auxiliary supports. *International Journal for Numerical Methods in Engineering*, 56:1291–1314, 2003.

[206] G. Ventura, J. X. Xu, and T. Belytschko. A vector level set method and new discontinuity approximations for crack growth by efg. *International Journal for Numerical Methods in Engineering*, 54:923–944, 2002.

[207] A. Portela, M.H. Aliabadi, and D.P. Rooke. Dual boundary element incremental-analysis of crack-propagation. *Computers & Structures*, 46:237–247, 1993.

[208] S. Bordas, M. Duflot, and P. Le. A simple error estimator for extended finite elements. *Communications in Numerical Methods in Engineering*, 24:961–971, 2008.

[209] M. Duflot and S. Bordas. A posteriori error estimation for extended finite elements by an extended global recovery. *International Journal for Numerical Methods in Engineering*, 76:1123–1138, 2008.

[210] T. Gratsch and K.J. Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & Structures*, 83:235–265, 2005.

[211] J. Panetier, P. Ladeveze, and L. Chamoin. Strict and effective bounds in goal-oriented error estimation applied to fracture mechanics problems solved with xfem. *International Journal for Numerical Methods in Engineering*, 81: 671–700, 2010.

[212] E. Stein, M. Rueter, and S. Ohnimus. Error-controlled adaptive goal-oriented modeling and finite element approximations in elasticity. *Computer Methods in Applied Mechanics and Engineering*, 196:3598–3613, 2007.

[213] H. Askes and S. Ilanko. The use of negative penalty functions in linear systems of equations. *Proceedings Of The Royal Society A-Mathematical Physical And Engineering Sciences*, 462:2965–2975, 2006.

[214] D. Alvarez, O. Dorn, N. Irishina, and M. Moscoso. Crack reconstruction using a level-set strategy. *Journal of Computational Physics*, 228:5710–5721, 2009.

[215] T. Belytschko, S. Loehnert, and J.H. Song. Multiscale aggregating discontinuities: A method for circumventing loss of material stability. *International Journal for Numerical Methods in Engineering*, 73:869–894, 2008.

[216] R.L. Barnett, P. Maragakis, A. Turner, M. Fyta, and E. Kaxiras. Multiscale model of electronic behavior and localization in stretched dry DNA. *Journal Of Materials Science*, 42:8894–8903, 2007.

[217] J.S. Chen, H.L. Teng, and A. Nakano. Wavelet-based multi-scale coarse graining approach for DNA molecules. *Finite Elements in Analysis and Design*, 43: 346–360, 2007.

[218] D. Qian, G.J. Wagner, and W.K. Liu. A multiscale projection method for the analysis of carbon nanotubes. 193:1603–1632, 2004.

[219] J. Ghanbari and R. Naghdabadi. Multiscale nonlinear constitutive modeling of carbon nanostructures based on interatomic potentials. *CMC-Computers Materials & Continua*, 10:4–64, 2009.

[220] P. Venkataraman, T.Y. Ng, and H. Li. Development of a novel multi-scale numerical technique. *Computational Materials Science*, 49:131–134, 2010.

Appendix A

Derivations

A.1 The derivatives of enriched basis functions

In §5.3.2, the enriched basis function is used to capture the singular stress field in the vicinity of a crack tip as

$$\mathbf{p}^a(\mathbf{x}) = \left\{1, x, y, \sqrt{r}\sin\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}, \sqrt{r}\sin\frac{\theta}{2}\sin\theta, \sqrt{r}\cos\frac{\theta}{2}\sin\theta\right\} \ . \tag{A.1.1}$$

where the first three terms are from linear basis and the latter four terms are the enriched part. The derivatives of basis function $\mathbf{p}^a(\mathbf{x})$ are required in obtaining the derivatives of the shape functions as described in §2.2. In the following, the derivatives of $\mathbf{p}^a(\mathbf{x})$ are provided for the latter four terms and those for the first three terms are trivial. The distance from a point of interest $\mathbf{x}$ to a crack tip is calculated by

$$r = \sqrt{\Delta x^2 + \Delta y^2} \tag{A.1.2}$$

where Δx and Δy are difference of x and y coordinates with respect to the crack tip. The angle θ is going in a counter clockwise direction and is calculated by

$$\theta = \text{atan}\left(\frac{\Delta y}{\Delta x}\right) \tag{A.1.3}$$

Note that r and θ are only dependent of x and y coordinates thus in 3D problem the derivatives with respect to z are zeros. Following the chain rule, the derivatives

of the enriched part can be obtained as

$$\frac{\partial\sqrt{r}\cos\frac{\theta}{2}}{\partial x}=\frac{1}{2r\sqrt{r}}\left(\Delta x\cos\frac{\theta}{2}+\Delta y\sin\frac{\theta}{2}\right) \tag{A.1.4}$$

$$\frac{\partial\sqrt{r}\cos\frac{\theta}{2}}{\partial y}=\frac{1}{2r\sqrt{r}}\left(-\Delta x\sin\frac{\theta}{2}+\Delta y\cos\frac{\theta}{2}\right) \tag{A.1.5}$$

$$\frac{\partial\sqrt{r}\sin\frac{\theta}{2}}{\partial x}=\frac{1}{2r\sqrt{r}}\left(\Delta x\sin\frac{\theta}{2}-\Delta y\cos\frac{\theta}{2}\right) \tag{A.1.6}$$

$$\frac{\partial\sqrt{r}\sin\frac{\theta}{2}}{\partial y}=\frac{1}{2r\sqrt{r}}\left(\Delta x\cos\frac{\theta}{2}+\Delta y\sin\frac{\theta}{2}\right) \tag{A.1.7}$$

$$\frac{\partial\sqrt{r}\sin\frac{\theta}{2}\sin\theta}{\partial x}=\frac{1}{2r\sqrt{r}}\left(\Delta x\sin\frac{\theta}{2}\sin\theta-\Delta y\cos\frac{\theta}{2}\sin\theta-2\Delta y\sin\frac{\theta}{2}\cos\theta\right) \tag{A.1.8}$$

$$\frac{\partial\sqrt{r}\sin\frac{\theta}{2}\sin\theta}{\partial y}=\frac{1}{2r\sqrt{r}}\left(\Delta x\cos\frac{\theta}{2}\sin\theta-2\Delta x\sin\frac{\theta}{2}\sin\theta+\Delta y\sin\frac{\theta}{2}\sin\theta\right) \tag{A.1.9}$$

$$\frac{\partial\sqrt{r}\cos\frac{\theta}{2}\sin\theta}{\partial x}=\frac{1}{2r\sqrt{r}}\left(\Delta x\cos\frac{\theta}{2}\sin\theta+\Delta y\sin\frac{\theta}{2}\sin\theta-2\Delta y\cos\frac{\theta}{2}\cos\theta\right) \tag{A.1.10}$$

$$\frac{\partial\sqrt{r}\cos\frac{\theta}{2}\sin\theta}{\partial y}=\frac{1}{2r\sqrt{r}}\left(-\Delta x\sin\frac{\theta}{2}\sin\theta+2\Delta x\cos\frac{\theta}{2}\cos\theta+\Delta y\cos\frac{\theta}{2}\cos\theta\right)\ . \tag{A.1.11}$$

A.2 Derivation of domain interaction integral

The interaction energy integral can be used to calculate the stress intensity factor (SIF) along the arbitrary crack front where the influence from the curvature of crack front can be considered. The method works out the energy released from the "virtual" advancing of crack front and extract the SIF from the energy with the auxiliary displacement field and stress field. It is a domain form of the well known J integration.

Consider an tubular domain denoted as Ω which is going around a part of a curved crack front as shown in Figure 5.16. The crack surface is denoted as S and

an auxiliary surface denoted as C is inserted tangent to S. Ω is the volume bounded by an inner surface S_0, outer surface S_t and two end surfaces of the tubular domain. The upper and lower faces of a crack surface are distinguished by superscripts $+$ and $-$ respectively. For example, S^+ and S^- are upper and lower faces of S. S and C are cut by S_t and the parts of S and C between S_t and S_0, shown as the pink and blue surfaces respectively in Figure 5.16, are denoted as S' and C' in the following derivation. An interaction integral $\tilde{I}$ is introduced for calculating the total energy release along the crack front as

$$\tilde{I} = \int_{S_t} P_{lj} \Delta a c_l(s) n_j \mathrm{d}S \tag{A.2.12}$$

where P_{lj} is the interaction form of the Eshelby energy momentum tensor [172] given as

$$P_{lj} = \left(\sigma_{ik}\varepsilon^{\mathrm{aux}}_{ik}\delta_{ij} - u^{\mathrm{aux}}_{i,l}\sigma_{ij} - u_{i,l}\sigma^{\mathrm{aux}}_{ij}\right) \ , \tag{A.2.13}$$

Δa is the virtual crack advancement along the crack front and is multiplied by a coefficient function $c_l(s)$ varying along the crack front, q_l is a test function defined by (5.56) and n_j is the unit outer normal along the integration path $\mathrm{d}S$ [169]. The choice of q_l has been shown is not important as long as it vanishes at the two ends of the integration domain [168]. It has been proven in [175] that $\tilde{I}$ is related to the SIFs as S_t approach to the crack front as follows

$$I = \frac{2(1-\nu^2)}{E}\left(K_{\mathrm{I}}K^{\mathrm{aux}}_{\mathrm{I}} + K_{\mathrm{II}}K^{\mathrm{aux}}_{\mathrm{II}}\right) + \frac{1}{\mu} \tag{A.2.14}$$

$$I = \frac{\lim\limits_{S_t \to 0} \tilde{I}}{\int_{L_c} \Delta a \mathrm{d}S} \tag{A.2.15}$$

where L_c is the length of crack front under consideration and $\int_{L_c} \Delta a \mathrm{d}s$ calculates the total crack advancement, and I calculates the energy release rate along the crack front in an average sense from $\tilde{I}$. Equation A.2.14 shows the SIFs can be solved if $\tilde{I}$ is known. However the direct calculation of $\tilde{I}$ will involve an integration of singularity terms when S_t approaches to zero. So we wish to find an integration form yielding an equivalent results to (A.2.14). Now we look at a volume integration for calculating the energy releas rate along the crack front [169] as

$$\int_{\Omega} (P_{lj}q_l)_{,j} \mathrm{d}\Omega = \int_{\Omega} \left(P_{lj,j}q_l + P_{lj}q_{l,j}\right) \mathrm{d}\Omega \ . \tag{A.2.16}$$

By using the divergence theorem and note q_l vanishes on S_0 and two ends of the tubular domain, the *l.h.s* of Equation (A.2.16) can be written into surface integral as

$$\int_{\Omega}(P_{lj}q_l)_{,j}\mathrm{d}\Omega = -\int_{S_t}P_{lj}q_l n_j \mathrm{d}S + \int_{C'^{+}+C'^{-}}P_{lj}q_l n_j \mathrm{d}S + \int_{S'^{+}+S'^{-}}P_{lj}q_l n_j \mathrm{d}S \tag{A.2.17}$$

where n_j is the unit outer normal at any point along $\partial\Omega$. Note that the unit outer normal is pointing outwards, thus the negative sign in $-S_t$ represents subtraction of volume Ω_t from Ω. Substituting Equation (A.2.12) and (5.56) into (A.2.17) results

$$\int_{\Omega}\left(P_{lj,j}q_l + P_{lj}q_{l,j}\right)\mathrm{d}\Omega = -\tilde{I} + \int_{C'^{+}+C'^{-}}P_{lj}q_l n_j \mathrm{d}S + \int_{S'^{+}+S'^{-}}P_{lj}q_l n_j \mathrm{d}S \ . \tag{A.2.18}$$

In the limit case when S_t approach to crack front, then S'^{+} and S'^{-} become S^{+} and S^{-}, and C'^{+} and C'^{-} become C^{+} and C^{-}. Applying this limit condition to Equation (A.2.18) and denoting $\lim\limits_{S_t\to 0}\tilde{I}$ as $\bar{I}$ gives

$$\bar{I} = \int_{\Omega}\left(P_{lj,j}q_l + P_{lj}q_{l,j}\right)\mathrm{d}\Omega + \int_{C_{+}+C^{-}}P_{lj}q_l n_j \mathrm{d}S + \int_{S^{+}+S^{-}}P_{lj}q_l n_j \mathrm{d}S \ . \tag{A.2.19}$$

Thus the integration $\tilde{I}$ involving singular terms as S_t vanishes is converted to a domain integration. In the following, we will derive the formulation to calculate P_{lj} and $P_{lj,j}$ which is needed in calculating Equation (A.2.19). The Eshelby momentum tensor P_{lj} in Equation (A.2.12) can be expressed as

$$\mathbf{P} = \boldsymbol{\sigma} : \boldsymbol{\varepsilon}^{\mathrm{aux}} - \nabla\mathbf{u}^{aux}\cdot\sigma - \nabla\mathbf{u}\cdot\sigma^{\mathrm{aux}} \ . \tag{A.2.20}$$

And $P_{lj,j}$ can be written as

$$\nabla\cdot\mathbf{P}^{T} = \nabla\cdot\left(\boldsymbol{\sigma} : \boldsymbol{\varepsilon}^{\mathrm{aux}} - \nabla\mathbf{u}^{aux}\cdot\sigma - \nabla\mathbf{u}\cdot\sigma^{\mathrm{aux}}\right) \ . \tag{A.2.21}$$

The Betti's reciprocal theorem gives symmetric results as

$$\boldsymbol{\sigma} : \boldsymbol{\varepsilon}^{aux} = \frac{1}{2}\boldsymbol{\sigma}^{\mathrm{aux}} : \boldsymbol{\varepsilon} + \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}^{\mathrm{aux}} \ . \tag{A.2.22}$$

Substituting into Equation (A.2.20) gives

$$\mathbf{P} = \left(\frac{1}{2}\boldsymbol{\sigma}^{\mathrm{aux}} : \boldsymbol{\varepsilon} + \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}^{aux}\right) - \nabla\mathbf{u}^{aux}\cdot\sigma - \nabla\mathbf{u}\cdot\sigma^{\mathrm{aux}} \ . \tag{A.2.23}$$

In the following, Equation (A.2.21) is derived by applying the $\nabla\cdot$ operation to each term on in the bracket respectively. Applying $\nabla\cdot(\cdot)^T$ to the first term gives

$$\nabla\cdot\left(\frac{1}{2}\boldsymbol{\sigma}^{aux}:\boldsymbol{\varepsilon}\right)^T=\frac{1}{2}\left(\boldsymbol{\sigma}^{aux}\nabla\right):\boldsymbol{\varepsilon}+\frac{1}{2}\boldsymbol{\sigma}^{aux}:\left(\boldsymbol{\varepsilon}\nabla\right)\ . \tag{A.2.24}$$

If the material tensor that relates $\boldsymbol{\sigma}$ to $\boldsymbol{\varepsilon}$ is symmetric, the relation exists

$$\left(\boldsymbol{\sigma}^{\text{aux}}\nabla\right):\boldsymbol{\varepsilon}=\boldsymbol{\sigma}^{\text{aux}}:\left(\boldsymbol{\varepsilon}\nabla\right) \tag{A.2.25}$$

so that Equation (A.2.24) becomes

$$\nabla\cdot\left(\frac{1}{2}\boldsymbol{\sigma}^{\text{aux}}:\boldsymbol{\varepsilon}\right)^T=\boldsymbol{\sigma}^{\text{aux}}:\left(\boldsymbol{\varepsilon}\nabla\right)\ . \tag{A.2.26}$$

Then applying $\nabla\cdot(\cdot)^T$ to the second term in Equation (A.2.20), we get

$$\nabla\cdot\left(\frac{1}{2}\boldsymbol{\sigma}:\boldsymbol{\varepsilon}^{\text{aux}}\right)^T=\boldsymbol{\sigma}:\left(\boldsymbol{\varepsilon}^{\text{aux}}\nabla\right)\ . \tag{A.2.27}$$

Applying $\nabla\cdot(\cdot)^T$ to the third term in Equation (A.2.20) gives

$$\nabla\cdot\left(\nabla\mathbf{u}^{\text{aux}}\cdot\boldsymbol{\sigma}\right)^T=\boldsymbol{\sigma}:\left(\nabla\mathbf{u}^{\text{aux}}\nabla\right)+\left(\nabla\mathbf{u}^{\text{aux}}\right)\cdot\left(\nabla\cdot\sigma\right)\ . \tag{A.2.28}$$

Similarly the derivatives of the fourth term is obtained as

$$\nabla\cdot\left(\nabla\mathbf{u}\cdot\boldsymbol{\sigma}^{\text{aux}}\right)^T=\boldsymbol{\sigma}^{\text{aux}}:\left(\nabla\mathbf{u}\nabla\right)+\left(\nabla\mathbf{u}\right)\cdot\left(\nabla\cdot\sigma^{\text{aux}}\right)\ . \tag{A.2.29}$$

With Equation (A.2.26) to (A.2.29) obtained, $\nabla\cdot\mathbf{P}^T$ can be calculated as

$$\begin{aligned}\nabla\cdot\mathbf{P}^T=\boldsymbol{\sigma}:\left(\boldsymbol{\varepsilon}^{\text{aux}}\nabla-\nabla\mathbf{u}^{\text{aux}}\nabla\right)-\left(\nabla\mathbf{u}\right)\cdot\left(\nabla\cdot\boldsymbol{\sigma}^{\text{aux}}\right)\\ -\boldsymbol{\sigma}^{\text{aux}}:\left(\boldsymbol{\varepsilon}\nabla-\nabla\mathbf{u}\nabla\right)-\left(\nabla\mathbf{u}^{\text{aux}}\right)\cdot\left(\nabla\cdot\boldsymbol{\sigma}\right)\end{aligned} \tag{A.2.30}$$

Note that the real stress field (stress results given by the numerical method) satisfied equilibrium condition which is $\nabla\cdot\boldsymbol{\sigma}=0$. The symmetry of σ^{aux} and compatibility condition between real displacement and strain gives

$$\boldsymbol{\sigma}^{\text{aux}}:\left(\boldsymbol{\varepsilon}\nabla-\nabla\mathbf{u}\nabla\right)=0, \tag{A.2.31}$$

So finally Equation (A.2.30) becomes

$$\nabla\cdot\mathbf{P}^T=\boldsymbol{\sigma}:\left(\boldsymbol{\varepsilon}^{\text{aux}}\nabla-\nabla\mathbf{u}^{\text{aux}}\nabla\right)-\left(\nabla\mathbf{u}\right)\cdot\left(\nabla\cdot\boldsymbol{\sigma}^{\text{aux}}\right)\ . \tag{A.2.32}$$

It should be noted that the auxiliary stress field does not satisfied the equilibrium condition that $\nabla \cdot \boldsymbol{\sigma}^{\text{aux}} \neq 0$ and the strain is not the symmetric gradient of displacement that

$$\boldsymbol{\sigma} : (\boldsymbol{\varepsilon}^{\text{aux}} \nabla - \nabla \mathbf{u}^{\text{aux}} \nabla) \neq 0, \tag{A.2.33}$$

and all the terms in Equation (A.2.32) need to be calculated with care.

Appendix B

Algorithms and schemes

B.1 The initialization of lens-shaped crack

```
for i = 1:num_crack
h = crack(i).h;
r = crack(i).r;
cpt = crack(i).centre;
a = crack(i).angle/2;
r = crack(i).r;
h = crack(i).h;

d = h-r/cos(a);
rim = r*sin(a);

gFi = zeros(size(grid.xs{3}));

R = sqrt((grid.xs{1}-cpt(1)).^2+(grid.xs{2}-cpt(2)).^2);
Rd = (h-grid.xs{3}).*tan(a);

IndexIn1 = (R <= Rd);
IndexIn2 = (grid.xs{3} <= h);
```

```
IndexIn = ((IndexIn1.*IndexIn2)>0);
IndexOut = ~IndexIn;

InFi = sqrt(R.^2+(grid.xs{3}-h).^2)-r;
OutFi = ((grid.xs{3}-d)./tan(a)-R).*sin(a);

gFi = -IndexIn.*InValue + IndexOut.*OutValue;
gSi = R-rim;
end
```

B.2 HJ WENO scheme for spatial derivatives

In this thesis, the spatial derivatives of a level set is obtained using the Hamilton-Jacobi weighted essential non-oscillatory (HJ WENO) scheme. The formulation of the scheme is described in the following for 1D field (assuming x direction). Since structured grid is used in computation, the derivative with respect to a certain axis is independent from those in other axes and the formulations in 3D problems are the same for y and z directions as for x direction. The backward and forward derivatives at a grid point are noted as $\phi^-_{,x}$ and $\phi^+_{,x}$ respectively, where the superscripts $-$ and $+$ denote backward and forward difference. In the following the formulation of calculating $\phi^-_{,x}$ will be presented and that of $\phi^+_{,x}$ follows a similar procedure. To calculate the derivatives at a grid point, e.g. i, four neighbour points are used as shown in Figure B.1, namely $i-1$, $i-2$, $i+1$ and $i+2$. Define the backward difference coefficients at the five point stencil as $v_1 = D^-\phi_{i-2}$, $v_2 = D^-\phi_{i-1}$, $v_3 = D^-\phi_i$, $v_4 = D^-\phi_{i+1}$ and $v_5 = D^-\phi_{i+2}$ where D^- is the first divided backward difference scheme, e.g. $D^-\phi_i$ is calculated by

$$D^-\phi_i = \frac{\phi_i - \phi_{i-1}}{\Delta x} \ . \tag{B.2.1}$$

and Δx is the spacing between i and $i-1$. Three potential derivatives at i are calculated by

$$\phi^1_{,x} = \frac{v_1}{3} - \frac{7v_2}{6} + \frac{11v_3}{6} \ , \tag{B.2.2}$$

$$\phi^2_{,x} = -\frac{v_2}{6} + \frac{5v_3}{6} + \frac{v_4}{3} \ , \tag{B.2.3}$$

and

$$\phi^3_{,x} = -\frac{v_3}{3} + \frac{5v_4}{6} - \frac{v_5}{6} \ . \tag{B.2.4}$$

In the WENO HJ scheme, the derivatives at i is calculated as a weighted combination of the three potential derivatives as

$$\phi_{,x} = \omega_1\phi^1_{,x} + \omega_2\phi^2_{,x} + \omega_3\phi^3_{,x} \tag{B.2.5}$$

where ω_k, ω_2 and ω_3 are weights for $\phi^1_{,x}$, $\phi^2_{,x}$ and $\phi^3_{,x}$ respectively and conform the following conditions

$$0 \le \omega_k \le 1, \quad k = 1,\, 2,\, 3 \tag{B.2.6}$$

$$\omega_1 + \omega_2 + \omega_3 = 1 \ . \tag{B.2.7}$$

In [203] the suggestions on choosing ω_k are given. Define the smoothness estimates of the three potential derivatives as

$$S_1 = \frac{13}{12}(v_1 - 2v_2 + v_3)^2 + \frac{1}{4}(v_1 - 4v_2 + 3v_3)^2 \ , \tag{B.2.8}$$

$$S_2 = \frac{13}{12}(v_2 - 2v_3 + v_4)^2 + \frac{1}{4}(v_2 - v_4)^2 \ , \tag{B.2.9}$$

and

$$S_3 = \frac{13}{12}(v_3 - 2v_4 + v_5)^2 + \frac{1}{4}(3v_3 - 4v_4 + v_5)^2 \ , \tag{B.2.10}$$

respectively. And then calculate the smoothness coefficients as

$$\alpha_1 = \frac{0.1}{S_1 + \epsilon^2} \ , \tag{B.2.11}$$

$$\alpha_2 = \frac{0.6}{S_2 + \epsilon^2} \ , \tag{B.2.12}$$

and

$$\alpha_3 = \frac{0.3}{S_3 + \epsilon^2} \tag{B.2.13}$$

with

$$\epsilon = 10^{-6}\max v_1^2,\, v_2^2,\, v_3^2,\, v_4^2,\, v_5^2 + 10^{-99} \tag{B.2.14}$$

where 10^{-99} is set to avoid devision over zero when solving α_k with ϵ. Finally, the weights ω_k are calculated by

$$\omega_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}\ , \tag{B.2.15}$$

$$\omega_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3}\ , \tag{B.2.16}$$

and

$$\omega_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}\ . \tag{B.2.17}$$

In smooth regions, S_k will be very small, and if S_k is small enough compared with ϵ then $\alpha_1 \approx 0.1\epsilon^{-2}$, $\alpha_2 \approx 0.6\epsilon^{-2}$ and $\alpha_3 \approx 0.6\epsilon^{-2}$. In this case the weights ω_k become

$$\omega_1 = 0.6\,, \quad \omega_2 = 0.3\,, \quad \omega_3 = 0.1\ . \tag{B.2.18}$$

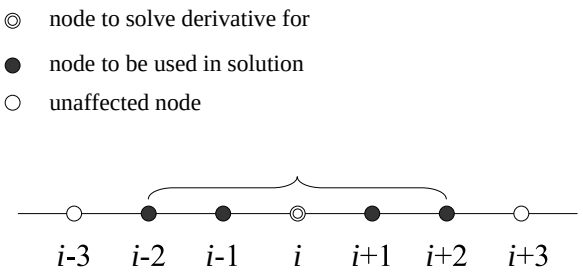


Figure B.1: The four neighbour points used to calculate the derivative at i.

B.3 TVD Runge-Kutta scheme for explicit time integration

In this thesis the time integration is implemented using the third order total variation Runge-Kutta scheme [112]. In the following ϕ_n denotes the value of ϕ at current time step i.e. $t = t_n$ (or pseudo time then noted as τ_n) in nth step of time integration, and ϕ_{n+1} denotes ϕ at $t = t_n + \Delta t$ with δt denotes the increment of time t. All

the other superscripts of ϕ follow the same rule. An forward Euler step is used to calculate firstly calculate the solution at $t = t_n + \Delta t$

$$\frac{\phi^{n+1} - \phi^n}{\Delta t} + V^n \cdot \nabla \phi^n = 0 \ . \tag{B.3.19}$$

And then further apply the Euler scheme to get the solution at $t = t_n + 2\delta t$

$$\frac{\phi^{n+2} - \phi^{n+1}}{\Delta t} + V^{n+1} \cdot \nabla \phi^{n+1} = 0 \ . \tag{B.3.20}$$

The intermediate for $\phi^{n+\frac{1}{2}}$ at $t = t_n + \frac{1}{2}\Delta t$ tis calculated by

$$\phi^{n+\frac{1}{2}} = \frac{3}{4}\phi^n + \frac{1}{4}\phi^{n+2} \tag{B.3.21}$$

followed by another Euler step as

$$\frac{\phi^{n+\frac{3}{2}} - \phi^{n+\frac{1}{2}}}{\Delta t} + V^{n+\frac{1}{2}} \cdot \nabla \phi^{n+\frac{1}{2}} = 0 \ . \tag{B.3.22}$$

Then ϕ^{n+1} at $t = t_n + \Delta_t$ is finally obtained by averaging

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^{n+\frac{1}{2}} \ . \tag{B.3.23}$$

Equation (B.3.23) produces a third-order accurate approximate to ϕ at $t = t_n + \Delta t$

Appendix C

Supplementary materials

C.1 The auxiliary fields for 3D domain integral

The auxiliary displacement field used here is the well-known out of plane displacement field around crack tip

$$u_1^{\text{aux}} = \frac{K_{\text{I}}^{\text{aux}}}{8\mu}\sqrt{\frac{2r}{\pi}}\left[(5-8\nu)\cos\frac{\theta}{2} - \cos\frac{3\theta}{2}\right] + \frac{K_{\text{II}}^{\text{aux}}}{8\mu}\sqrt{\frac{2r}{\pi}}\left[(9-8\nu)\sin\frac{\theta}{2} + \sin\frac{3\theta}{2}\right] \tag{C.1.1}$$

$$u_2^{\text{aux}} = \frac{K_{\text{I}}^{\text{aux}}}{8\mu}\sqrt{\frac{2r}{\pi}}\left[(7-8\nu)\sin\frac{\theta}{2} - \sin\frac{3\theta}{2}\right] - \frac{K_{\text{II}}^{\text{aux}}}{8\mu}\sqrt{\frac{2r}{\pi}}\left[(3-8\nu)\cos\frac{\theta}{2} + \cos\frac{3\theta}{2}\right] \tag{C.1.2}$$

$$u_3^{\text{aux}} = \frac{K_{\text{III}}^{\text{aux}}}{8\mu}\sqrt{\frac{2r}{\pi}}\sin\frac{\theta}{2}. \tag{C.1.3}$$

The auxiliary stress field around the crack tip uses the Westergaard solution of stress field near a crack tip as

$$\sigma_{11}^{\text{aux}} = \frac{K_{\text{I}}^{\text{aux}}}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1-\sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) - \frac{K_{\text{II}}^{\text{aux}}}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\left(2+\cos\frac{\theta}{2}\cos\frac{3\theta}{2}\right) \tag{C.1.4}$$

$$\sigma_{22}^{\text{aux}} = \frac{K_{\text{I}}^{\text{aux}}}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1+\sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) + \frac{K_{\text{II}}^{\text{aux}}}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} \tag{C.1.5}$$

$$\sigma_{12}^{\text{aux}} = \frac{K_{\text{I}}^{\text{aux}}}{\sqrt{2\pi r}}\sin\frac{\theta}{2}\cos\frac{\theta}{2}\cos\frac{3\theta}{2} + \frac{K_{\text{II}}^{\text{aux}}}{\sqrt{2\pi r}}\cos\frac{\theta}{2}\left(1-\sin\frac{\theta}{2}\sin\frac{3\theta}{2}\right) \tag{C.1.6}$$

$$\sigma_{23}^{\text{aux}} = \frac{K_{\text{III}}^{\text{aux}}}{\sqrt{2\pi r}} \cos\frac{\theta}{2} \tag{C.1.7}$$

$$\sigma_{13}^{\text{aux}} = -\frac{K_{\text{III}}^{\text{aux}}}{\sqrt{2\pi r}} \sin\frac{\theta}{2} \tag{C.1.8}$$

$$\sigma_{33}^{\text{aux}} = \nu(\sigma_{11}^{\text{aux}} + \sigma_{22}^{\text{aux}}) \tag{C.1.9}$$

The displacement field is defined locally in the 3D curvilinear coordinate system. As the coordinates rotates along the crack front, the displacement varies from point to point at the crack front. It can be seen that the displacement is independent of ξ_3, so all the derivatives with respect to ξ_3 vanishes. The strain tensor can be simplified into

$$\varepsilon_{11}^{\text{aux}} = \frac{\partial u_1}{\partial \xi_1} \tag{C.1.10}$$

$$\varepsilon_{12}^{\text{aux}} = \frac{1}{2}\left(\frac{\partial u_1}{\partial \xi_2} + \frac{\partial u_2}{\partial \xi_1}\right) \tag{C.1.11}$$

$$\varepsilon_{13}^{\text{aux}} = \varepsilon_{31} = \frac{1}{2}\frac{\partial u_3}{\partial \xi_1} \tag{C.1.12}$$

$$\varepsilon_{22}^{\text{aux}} = \frac{\partial u_2}{\partial \xi_2} \tag{C.1.13}$$

$$\varepsilon_{23}^{\text{aux}} = \varepsilon_{32} = \frac{1}{2}\frac{\partial u_3}{\partial \xi_2}\ , \tag{C.1.14}$$

and $\varepsilon_{33}^{\text{aux}}$ is zero.

C.2 The auxiliary fields for 2D contour integral

The contour integral method [167] for calculating SIFs in 2D has been introduced in 5.4.3. The auxiliary displacement components used are as follows

$$\hat{u}_r = \frac{1}{2\sqrt{2\pi r}(1+\kappa)}\left((2\kappa+1)\cos\frac{3\theta}{2} - 3\cos\frac{\theta}{2}\right)c_1 \tag{C.2.15}$$
$$+ \frac{1}{2\sqrt{2\pi r}(1+\kappa)}\left((2\kappa+1)\sin\frac{3\theta}{2} - \sin\frac{\theta}{2}\right)c_2$$

$$\hat{u}_\theta = \frac{1}{2\sqrt{2\pi r}(1+\kappa)}\left(-(2\kappa-1)\sin\frac{3\theta}{2} + 3\sin\frac{\theta}{2}\right)c_1 \tag{C.2.16}$$

$$+ \frac{1}{2\sqrt{2\pi r}(1+\kappa)}\left((2\kappa-1)\cos\frac{3\theta}{2} - \cos\frac{\theta}{2}\right)c_2 \tag{C.2.17}$$

And the auxiliary stress field is given by

$$\hat{\sigma}_r = \frac{G}{2\sqrt{2\pi r^3}(1+\kappa)}\left[\left(7\cos\frac{3\theta}{2} - 3\cos\frac{\theta}{2}\right)c_1 + \left(7\sin\frac{3\theta}{2} - \sin\frac{\theta}{2}\right)c_2\right] \quad \text{(C.2.18)}$$

$$\hat{\sigma}_\theta = \frac{G}{2\sqrt{2\pi r^3}(1+\kappa)}\left[\left(\cos\frac{3\theta}{2} + 3\cos\frac{\theta}{2}\right)c_1 + \left(\sin\frac{3\theta}{2} + \sin\frac{\theta}{2}\right)c_2\right] \quad \text{(C.2.19)}$$

$$\hat{\sigma}_{r\theta} = \frac{G}{2\sqrt{2\pi r^3}(1+\kappa)}\left[3\left(\sin\frac{3\theta}{2} + \sin\frac{\theta}{2}\right)c_1 - \left(3\cos\frac{3\theta}{2} + \cos\frac{\theta}{2}\right)c_2\right] . \quad \text{(C.2.20)}$$