



**João Guilherme
Gaspar Cordeiro
Ferreira**

Numerical Modelling of Fatigue Crack Growth Using XFEM

Modelação Numérica de Propagação de Fendas de Fadiga
Usando XFEM



**João Guilherme
Gaspar Cordeiro
Ferreira**

Numerical Modelling of Fatigue Crack Growth Using XFEM

Modelação Numérica de Propagação de Fendas de Fadiga
Usando XFEM

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob orientação científica de Joaquim Alexandre Mendes de Pinho da Cruz, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro, e de João Alexandre Dias de Oliveira, Professor Auxiliar do Departamento de Engenharia Mecânica da Universidade de Aveiro.

O júri / The jury

Presidente / President

Prof. Doutor Ricardo José Alves de Sousa

Professor Auxiliar da Universidade de Aveiro

Vogais / Committee

Prof. Doutor Fernando Jorge Ventura Antunes

Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Prof. Doutor Joaquim Alexandre Mendes de Pinho da Cruz

Professor Auxiliar da Universidade de Aveiro (orientador)

Agradecimentos / Acknowledgements

Em primeiro lugar quero agradecer ao Professor Alexandre Cruz e ao Professor João Oliveira, por todo o tempo que despenderam comigo, não só durante a elaboração deste trabalho como também durante as inúmeras aulas que presenciei.

Agradeço também a todas as pessoas que direta ou indiretamente contribuíram para alegrar a minha vida. Não vou estar a nomear ninguém pois felizmente são tantas pessoas que iria certamente acabar por me esquecer de alguém.

Em último lugar, quero agradecer àquela pessoa que tornou tudo isto possível devido à sua persistência quando nem tudo corria pelo melhor.

Palavras-chave

XFEM; FEM; Método *level-set*; Descontinuidade; Fenda

Resumo

O Método dos Elementos Finitos (*Finite Element Method* – FEM) é uma das ferramentas mais eficientes para a obtenção de soluções numéricas de Equações Diferenciais Parciais (EDP) em mecânica dos sólidos computacional. Esta técnica numérica tem vindo a ser utilizada extensivamente durante as últimas décadas para a obtenção de soluções aproximadas de EDP, tanto a nível de engenharia como a nível científico. Uma das principais características do FEM é a subdivisão de um meio contínuo numa série de elementos discretos, estando esses elementos ligados por um mapa topológico, normalmente referido como malha dos elementos finitos. O FEM é utilizado geralmente para modelar e prever o comportamento de estruturas contínuas. Contudo, surgem problemas quando o FEM é utilizado em domínios que contenham descontinuidades (tais como fendas). Neste caso, é normalmente utilizado um refinamento de malha adaptativo em torno da descontinuidade. Este processo funciona perfeitamente, mas acarreta um enorme custo computacional. Alternativamente, o Método dos Elementos Finitos Estendidos (*eXtended Finite Element Method* – XFEM) é um método numérico utilizado para modelar descontinuidades fortes e fracas, utilizando enriquecimento local. É uma generalização do FEM que permite a incorporação de enriquecimento local de aproximação de espaços. Este enriquecimento é feito através do conceito de partição de unidade, ao adicionar funções especiais à aproximação por elementos finitos. Para a modelação de uma fenda em regime linear elástico isotrópico, é utilizada a função de Heaviside para enriquecer os elementos que são completamente cortados pela fenda, e a função assintótica para enriquecer os elementos que contenham a ponta de fenda. Este processo de enriquecimento cria novos graus de liberdade que têm de ser incorporados no sistema, através de uma etapa de pós-processamento. Isto permite que o domínio possa ser modelado, sem que exista a preocupação de fazer coincidir a malha com a localização da fenda, e que seja preciso recorrer a um processo de remalhamento caso exista propagação da fenda.

Neste contexto, o presente trabalho aborda os principais conceitos de FEM e XFEM, a criação de um *software* pedagógico de XFEM (com o seu processo de implementação numérica e manual do *software*) e as principais diferenças entre a implementação padrão do FEM e um programa de XFEM. Finalmente, são apresentados alguns resultados numéricos da aplicação do XFEM.

Keywords

XFEM; FEM; Level-set method; Discontinuity; Crack

Abstract

The Finite Element Method (FEM) is one of the most efficient tools used, in computational solid mechanics, for the numerical solution of Partial Differential Equations (PDE). This numerical technique has been extensively used in the past decades for finding approximate solutions to PDE in both engineering and science fields. A main feature of the FEM is the subdivision of a continuum into a discrete set of elements, being these elements connected by a topological map, usually referred to as the finite element mesh. The FEM can generally be used to model and predict the behaviour of continuous structures. However, problems arise when FEM is used on a domain with a discontinuity (like a crack). In this case, it is usual to use adaptive mesh refinement around the discontinuity. This process works fine, but has a very high computational cost. Alternatively, the eXtended Finite Element Method (XFEM) is a numerical method for modelling strong and weak discontinuities using local enrichment. It is a FEM generalization that enables the incorporation of local enrichment of approximation spaces. This enrichment is done through the partition of unity concept by adding special functions to the finite element approximation. For crack modelling in isotropic linear elasticity, the Heaviside function is used to enrich the completely cut elements and an asymptotic function is used to enrich the crack tip elements. This enrichment creates new degrees of freedom that must be integrated into the analysis during a post-processing step. This enables the domain to be modelled without explicitly meshing the crack surfaces and without a remeshing process for the crack propagation.

In this context, this work addresses the main concepts of FEM and XFEM, the creation of a pedagogical XFEM software (with its numerical implementation process and software manual) and the differences between a standard FEM implementation and a XFEM program. Finally, some numerical results of the XFEM application are presented.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
1.3	Reading guide	2
2	Fracture Mechanics	5
2.1	Introduction	5
2.2	Modes of failure	5
2.3	Stress intensity factor	6
2.4	Stresses and displacements around the crack tip	7
2.5	J-integral	7
2.5.1	Interaction integral	8
2.6	Crack propagation	10
3	Finite Element Method	11
3.1	Introduction	11
3.2	Overview of the Finite Element Method	12
3.2.1	FEM formulation	12
3.2.2	Isoparametric elements	14
3.2.3	Stiffness matrix	15
3.2.4	Element integration	16
3.2.5	Problem solving	17
3.2.6	Strain and stress	17
3.3	Finite Element Method program	18
4	Extended Finite Element Method	21
4.1	Introduction	21
4.2	Modelling strong discontinuities	22
4.3	Modelling weak discontinuities	22
4.4	XFEM formulation	23
4.4.1	Discrete equations	24
4.4.2	Element partitioning and integration	25
4.5	Enrichment functions	26
4.5.1	Heaviside function	26
4.5.2	Asymptotic function	27
4.6	Detecting enriched elements	29
4.7	Blending elements	30

4.8	XFEM program	30
5	The XFrac-2D Program	33
5.1	Introduction	33
5.2	Selection of enriched elements	33
5.3	B operator	34
5.3.1	Heaviside enrichment	35
5.3.2	Asymptotic enrichment	35
5.3.3	Shifted enrichment	36
5.4	Stiffness matrix	37
5.4.1	Assembly	37
5.5	Numerical integration of the enriched elements	38
5.5.1	Subdomain	39
5.5.2	Gauss points	39
5.6	Calculation of SIF	40
5.6.1	Domain of the interaction integral	40
5.6.2	Interaction integral in FEM	41
5.6.3	Parameters for real state	42
5.6.4	Parameters of the auxiliary state for mode I	43
5.6.5	Parameters of the auxiliary state for mode II	44
5.7	Crack propagation	45
6	Numerical Examples	47
6.1	FEM validation	47
6.1.1	Tensile test	47
6.1.2	Bending test	48
6.1.3	FEM remarks	49
6.2	Modification of the detecting process	49
6.2.1	Crack tip condition	50
6.2.2	Cracks with multiple vectors	50
6.3	XFrac-2D validation	50
6.3.1	Plate with a central crack under uniaxial tension	50
6.3.2	Plate with an edge crack	52
6.3.3	Plate with a slanted crack	56
6.3.4	Crack propagation	60
7	Final Remarks	63
7.1	Future work	64
	Bibliography	67
	Appendices	73
A	XFrac-2D manual	75
A.1	Pre-processing	75
A.2	Processing	76
A.3	Post-processing	76
A.4	Modelling examples	77

A.5	Subroutines list	81
A.6	XFrac-2D flowchart	84
B	List of Tables	87
C	List of Figures	91

This page was intentionally left blank.

Chapter 1

Introduction

Nowadays, the Finite Element Method (FEM) is one of the most efficient tools to get a numerical solution on problems using differential equations. It is one of the most used methods in numerical simulation software on account of its reliable results and efficiency. However, problems arise when a FEM software is used on a domain with a discontinuity (such as a crack). In this case, it is necessary to use a very careful mesh around the discontinuity and a constant remeshing for each iteration (when there is a propagation of the discontinuity). These processes work fine, but have a very high computational cost. The eXtended Finite Element Method (XFEM) is a recent approach to this problem. It is based on Finite Element Method, but without all the mesh specification.

Before someone tries to understand how a specific XFEM software works, there is the need to know what is the XFEM and how it works, and since the XFEM core is based on the FEM, it is important to know some of the main FEM concepts too. Lastly, since the XFEM uses enrichment functions for a crack modelling that represent the fields around the crack tip, it is also necessary to understand some concepts and methods from Fracture Mechanics.

1.1 Motivation

The main motivation behind this work is the opportunity of doing something different in the University of Aveiro. The author already had learned about the theory of FEM and how it worked in Abaqus. However, he never understood how FEM was implemented and what happens inside a commercial software like Abaqus. So he wanted to merge the knowledge learned in the subjects of computational mechanics and simulation of technological processes in order to create his own software.

However the FEM has some issues when dealing with discontinuities like a crack. The XFEM is presented as something that could be added to a standard FEM program, acting like a subprogram or a patch, solving some of these issues and improving its performance and accuracy. Another motivations behind XFEM are the fact that it was a very recent method; it is a completely new field for the author, in which he expects to learn a lot; the opportunity to establish the baseline for a new research field in the university and the opportunity to create a work that will be very useful to further researchers.

1.2 Objectives

The main task of this work is the development of a pedagogical XFEM software and the elaboration of a document explaining all the process. As already said, the XFEM is a recent method, meaning there isn't available a lot of information about its implementation process. That is why the primary objective of this work is the creation of a tool that can be useful to any researcher starting in the XFEM field. For this reason, it is better to keep the software simple, organized and intuitive, instead of optimizing memory, speed and number of code lines. With these objectives in mind, the author opted to create a software that can only model 2-D domains with quadrilateral elements and didn't merge FEM and XFEM subroutines.

Nowadays, it isn't hard to find information about the theoretical part. There are some books explaining the theory behind XFEM [1; 2; 3]. However, since this is a recent method it is very difficult to find documents describing completely the implementation process and its little variants. That is why it is very important that this document explains not only the theory behind the method but also its implementation process.

In summary, the four main objectives of this work are the:

- Development of a pedagogical eXtended Finite Element Method software (Xfrac-2D); it should be easy for anyone to use it and to understand what each subroutine does;
- Usage of an open-source language for the software development;
- Validation of the eXtended Finite Element Method software;
- Development of a program manual, in order to help future researchers and users, explaining how the software works and what happens in each subprogram.

A FEM software is to be created and then the XFEM implemented into the software. The program is to be developed in Octave. This fact is very important for the spread of this work, since it is an open-source software very identical to Matlab, meaning that any person can access and use the Xfrac-2D software for free.

1.3 Reading guide

This work puts together the main aspects of Fracture Mechanics, FEM and XFEM related to the creation of a XFEM program.

Chapter 2 explains some basic concepts of Fracture Mechanics and shows how to determine the stress intensity factors. In Chapter 3 there's a brief review of the Finite Element Method and the way it works. Chapter 4 shows how to model a two dimension crack using the eXtended Finite Element Method. It shows what needs to be done to transform a normal FEM program into a XFEM program. Chapter 5 explains all the subjects related to the XFEM implementation, showing how to detect the elements, what needs to be done to enrich them, how to get K_I and K_{II} from the displacements, and some of the problems faced during the implementation process.

Chapter 6 presents the numerical results obtained by the FEM and XFEM programs for different tests. In order to validate both programs, the results are compared with

analytical solution for each test. When the results aren't accurate enough, some parameters are changed in order to try to lower the error value. Finally, Chapter 7 presents the main conclusions of this work and some proposals for future work.

This page was intentionally left blank.

Chapter 2

Fracture Mechanics

2.1 Introduction

When a structural component collapses, it doesn't always mean that it was badly projected. Sometimes, it is the presence of a defect on the material that is responsible for its weakness. This is one of the reasons why a component can collapse when subject to much lower efforts than that it was designed for. Fracture Mechanics is the branch of engineering that studies this kind of defects and the influence they have on the structure integrity. Around the defects, the stress can be much higher than on the overall body. It is this localized stress that is responsible for the collapse of the structure. Even when subject to lower efforts than those it was designed for (because the stress around the defect can be higher than the tensile strength), this can generate a local rupture that can spread and destroy the structure.

It was Griffith [4] who first compared the strength needed for breaking brittle materials. The experimental values were about ten times lower than the theoretical values (strength needed for breaking the atomic bonds). He thought these results were due to the presence of microscopic cracks/flaws on the material. He also developed a relation between the critical stress, the kind of brittle material and the crack length. However, there is a development of a plastic zone behind the crack tip on ductile materials. G. Irwin [5], around 1950, modified the Griffith criterion [4] so that it could be used on ductile materials. E. Orowan came to a similar conclusion [6], establishing that a material's fracture resistance could be split into surface energy, the elastic energy released with the crack growing and strain energy that comes from plastic energy dissipation. This plastic zone will increase with the growth of the crack, causing an increment of the plastic energy dissipation and the fracture resistance of the material [7].

2.2 Modes of failure

There are three modes of failure in Fracture Mechanics. The types of stresses around the crack tip is very different for each mode. The three modes are shown in Figure 2.1. A material can suffer from only one mode or a combination of two or even three modes:

- Mode I: uniaxial load (traction or compression) normal to the crack plane;
- Mode II: shear stress parallel to the crack plane and perpendicular to the crack front;

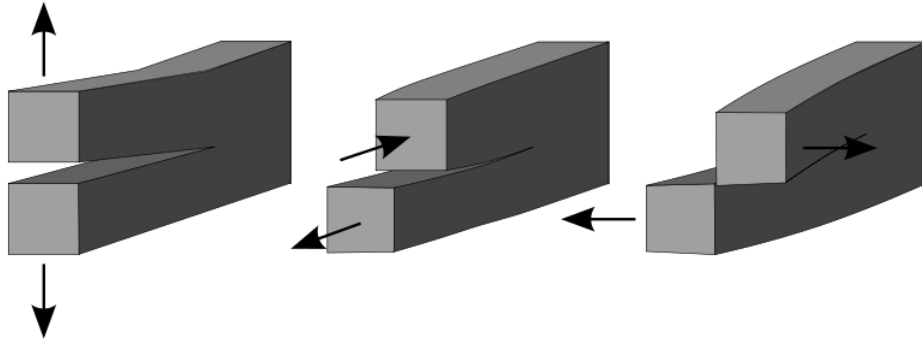


Figure 2.1: Modes of failure in Fracture Mechanics [8].

- Mode III: shear stress parallel to the crack plane and to the crack front.

2.3 Stress intensity factor

The relation between the local and global stress is called the stress intensity factor (SIF) and is represented by K . This concept was introduced by Inglis [9], in the beginning of the 20th century. It can be used to find what is the maximum global stress a structure can handle before it collapses or what is the critical crack size (knowing how big K can be).

The stress intensity factor can be determined by

$$K_c = \sqrt{GE} \quad (2.1)$$

and

$$K_c = \sqrt{\frac{GE}{1-\nu^2}}, \quad (2.2)$$

where equation 2.1 is for a plane stress state and equation 2.2 is for a plane strain state, K is the stress intensity factor and c is the mode of failure. G is the energy release rate, E is the Young's modulus and ν is the Poisson's ratio.

There are some analytical solutions to determine K_c in a fracture problem. However, it only works for the simplest cases and loads. In most situations, the only way to determine K_c is with a numerical method.

2.4 Stresses and displacements around the crack tip

Irwin continued Griffith's work and established that around the crack tip the stresses in mode I+II are given by [10]

$$\begin{aligned} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{bmatrix} &= \frac{K_I \cos\left(\frac{\theta}{2}\right)}{\sqrt{2\pi r}} \begin{bmatrix} 1 - \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right) \\ 1 + \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right) \end{bmatrix} \\ &+ \frac{K_{II}}{\sqrt{2\pi r}} \begin{bmatrix} -\sin\left(\frac{\theta}{2}\right) [2 + \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right)] \\ \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right) \\ \cos\left(\frac{\theta}{2}\right) [1 + \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{3\theta}{2}\right)] \end{bmatrix}, \end{aligned} \quad (2.3)$$

where r is the distance between the crack tip and the point in analyses, and θ is the angle between the r vector and the crack tip vector. The u_x and u_y displacements, around the crack tip, are given by

$$\begin{aligned} \begin{bmatrix} u_x \\ u_y \end{bmatrix} &= \frac{1}{4G} \sqrt{\frac{r}{2\pi}} \left(K_I \begin{bmatrix} (2k-1) \cos\left(\frac{\theta}{2}\right) - \cos\left(\frac{3\theta}{2}\right) \\ (2k+1) \sin\left(\frac{\theta}{2}\right) - \sin\left(\frac{3\theta}{2}\right) \end{bmatrix} \right. \\ &\left. + K_{II} \begin{bmatrix} -(2k+3) \sin\left(\frac{\theta}{2}\right) - \sin\left(\frac{3\theta}{2}\right) \\ (2k-3) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right) \end{bmatrix} \right), \end{aligned} \quad (2.4)$$

where r is the distance between the crack tip and the point in analysis, θ is the angle between the r vector, and the crack tip vector, and k is the Kolsov constant which is given by

$$k = 3 - 4\nu \quad (2.5)$$

for a plane strain state, and by

$$k = \frac{3 - \nu}{1 + \nu}, \quad (2.6)$$

for a plane stress state, which is related with the Poisson's ratio (ν).

2.5 J-integral

The J-integral was introduced by Cherapanov and Rice [11] in the end of the 1960s. The J-integral is a way to measure the strain energy release rate for the material when the crack tip deformation doesn't obey the linear elastic laws. The path of the J-integral must surround the crack tip and be closed. The path is independent once it respects these conditions. Figure 2.2 represents a crack tip surrounded by a J-integral path. This means the near-tip deformations can be related with a far field around the crack. That field obeys the elastic laws because it doesn't have plastic deformations. In this way, the

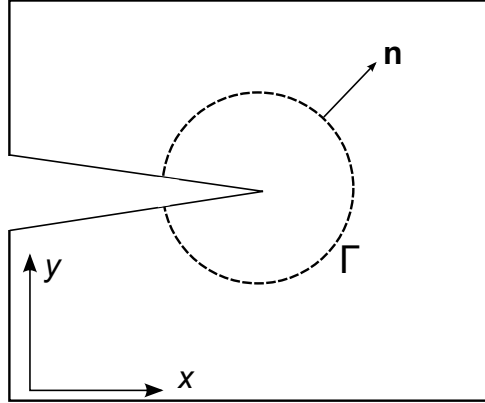


Figure 2.2: The J-integral path around a crack.

energy release rate can be determined using a domain that avoids the crack tip and its problems.

This method is very important, since the value of J is equal to G (see equations 2.1 and 2.2). This means that the J-integral can be used to determine the values of K_c . Equations 2.1 and 2.2 can be combined in

$$K_c = \sqrt{JE^*}, \quad (2.7)$$

where E^* is equal to E for a plane stress state and to $\frac{E}{1-\nu^2}$ for a plane strain state. The value of the J is given by

$$J = \int_{\Gamma} \left(W dy - \mathbf{T} \cdot \frac{\partial \mathbf{u}}{\partial x} d\Gamma \right), \quad (2.8)$$

where \mathbf{T} is the traction vector, W is the strain energy density, $\frac{\partial \mathbf{u}}{\partial x}$ is the gradient of the displacement field on the x -direction and Γ is the curve surrounding the crack. The strain energy density is given by

$$W = \sigma \varepsilon, \quad (2.9)$$

where σ and ε are, respectively, the stress and the strain.

2.5.1 Interaction integral

On a multiple mode fracture it is impossible to obtain the stress intensity factor independently for each individual mode. If it is a I+II mode, the interaction integral can be used to obtain K_I and K_{II} [12]. In order to get it, there is the need to define two equilibrium states on the body. State 1 is the actual state of the body and state 2 is an auxiliary state. $J^{(1+2)}$ is the new equilibrium state and can be calculated with the J-integral equation (equation 2.8), rewritten in [12]

$$J^{(1+2)} = \int_{\Gamma} \left[\frac{1}{2} (\sigma_{ij}^{(1)} + \sigma_{ij}^{(2)}) (\varepsilon_{ij}^{(1)} + \varepsilon_{ij}^{(2)}) \delta_{1j} - (\sigma_{ij}^{(1)} + \sigma_{ij}^{(2)}) \frac{\partial (u_i^{(1)} + u_i^{(2)})}{\partial x_1} \right] n_j d\Gamma \quad (2.10)$$

and

$$J^{(1+2)} = J^{(1)} + J^{(2)} + I^{(1+2)}. \quad (2.11)$$

After simplifying the equation 2.10, there is the new term $I^{(1+2)}$ – the interaction integral between the state 1 and 2 – given by [12]

$$I^{(1+2)} = \int_{\Gamma} \left[W^{(1+2)} \delta_{1j} - \sigma_{ij}^{(1)} \frac{\partial u_i^{(2)}}{\partial x_1} - \sigma_{ij}^{(2)} \frac{\partial u_i^{(1)}}{\partial x_1} \right] n_j d\Gamma, \quad (2.12)$$

where $W^{(1+2)}$ is the body interaction strain energy

$$W^{(1+2)} = \sigma_{ij}^{(1)} \varepsilon_{ij}^{(2)} = \sigma_{ij}^{(2)} \varepsilon_{ij}^{(1)}. \quad (2.13)$$

For a mode I+II, J corresponds to [12]

$$J = \frac{K_I^2}{E^*} + \frac{K_{II}^2}{E^*}, \quad (2.14)$$

and equation 2.11 can be rewritten as [12]

$$J^{(1+2)} = J^{(1)} + J^{(2)} + \frac{2}{E^*} \left(K_I^{(1)} K_I^{(2)} + K_{II}^{(1)} K_{II}^{(2)} \right), \quad (2.15)$$

meaning that

$$I^{(1+2)} = \frac{2}{E^*} \left(K_I^{(1)} K_I^{(2)} + K_{II}^{(1)} K_{II}^{(2)} \right). \quad (2.16)$$

The I-integral has only the interaction terms and can be used to calculate each stress intensity factor. Like the J-integral, it has an independent path, meaning that the same techniques can be applied.

To get K_I and K_{II} there is the need to take into account the following considerations for the auxiliary state:

- considering state 2 as pure I mode, meaning that $K_I^{(2)} = 1$ and $K_{II}^{(2)} = 0$:

$$K_I^{(1)} = \frac{E^*}{2} I^{(1, \text{Mode I})}. \quad (2.17)$$

- considering state 2 as pure II mode, meaning that $K_I^{(2)} = 0$ and $K_{II}^{(2)} = 1$:

$$K_{II}^{(1)} = \frac{E^*}{2} I^{(1, \text{Mode II})}. \quad (2.18)$$

In equation 2.17, the value of $K_{II}^{(1)}$ is irrelevant since it is associated with $K_{II}^{(2)} = 0$ and, in the previous equation, the value of $K_I^{(1)}$ is irrelevant since it is associated with $K_I^{(2)} = 0$.

2.6 Crack propagation

In each iteration, after the the stresses around the crack tip reach a critical level, the crack segment must grow with a certain angle and length. The crack propagation angle (β) is a very important parameter for the accuracy of the problem, since an incorrect crack path can lead to wrong SIF results. There are several criteria to calculate the crack propagation angle, giving some of them better results in particular situations [7]. Two of the most commonly used criteria are the maximum energy release rate and the maximum hoop stress. The maximum energy release rate assumes that crack propagation will start when the maximum energy release rate reaches a critical value (G_{cr}), where G_{cr} is a materiel property [13]. The crack will grow in a radial direction, from the crack tip to where the energy release rate is maximum. The crack propagation angle, for this criterion, is given by

$$\beta = 2 \arctan \left(\frac{K_I}{K_{II}} \right), \quad (2.19)$$

where β is the crack propagation angle in the local coordinate system, K_I and K_{II} are the stress intensity factor for mode I and II, respectively. The critical maximum energy release rate is a function of the J-integral and is given by [14]

$$G_{cr} = J_1 \cos(\beta) + J_2 \sin(\beta), \quad (2.20)$$

where β is the crack propagation angle in the local coordinate system. J_1 and J_2 are the values of the J-integral for modes I and II, respectively.

The maximum hoop stress criterion is the most commonly used in Linear Elastic Fracture Mechanics [15]. This criterion is based on the evaluation of the mixed mode SIF (K_I and K_{II}). Like in the previous criteria, it assumes that the crack initiation will start when the maximum hoop stress reaches a critical value. The crack propagation angle for this criterion is given by

$$\beta = 2 \arctan \left[\frac{1}{4} \left(\frac{K_I}{K_{II}} \pm \sqrt{\left(\frac{K_I}{K_{II}} + 8 \right)^2} \right) \right]. \quad (2.21)$$

This criterion presents good results for traction-free crack surfaces. However, its maximum angle is only 70.5° , meaning it has some limitations when facing pure mode II problems [7].

The following equation was proposed by Liang as being more efficient than equation 2.21 [16]:

$$\beta = 2 \arctan \frac{-2 \frac{K_{II}}{K_I}}{1 + \sqrt{1 + 8 \left(\frac{K_{II}}{K_I} \right)^2}}. \quad (2.22)$$

Depending on the problem in study, the crack propagation angle (β) can be always the same or assumed a different value for each interaction.

Chapter 3

Finite Element Method

3.1 Introduction

Nowadays, the Finite Element Method (FEM) is one of the most widely used methods on numerical simulation programs. It provides a reliable and efficient resolution for problems based on differential equations. The Finite Element Method divides the main problem (domain) into subproblems (elements). It analyses each element individually and assembles the elements in the global problem. These elements are much simpler and easier to analyse and solve than the global problem. There are points on the domain, called nodes. They must have the properties of the surrounding elements. Having the main problem a continuous domain, the elements must be continuous too. FEM uses continuous polynomial functions to link the nodes and ensure the continuity of the domain. This polynomial functions depend on the type of element used in the analysis [17; 18; 19].

In 1943, Courant developed a methodology for the discretization of continuous domains by triangular elements [20]. In 1969, Turner *et al.* created and described the first finite elements [21]. In their work, they developed the first triangular, quadrilateral and rectangular elements. The triangular element had six degrees of freedom that which the displacement in their nodes. The quadrilateral element was obtained by the inclusion of a virtual internal node and then it was split into four triangular elements. In 1959, Greenstadt discretized some mathematical functions into continuous subdomains, in order to determine their solution [22].

During the 1950s, John Argyris and his research group published several articles about matrix analysis of linear systems with discrete elements. Their work established the concepts of stiffness and flexibility matrix and how to obtain them using the matrix operators [17].

Ray Clough analysed some plane state problems in 1960 and established the systematic procedure used today [23]. In 1965 and 1967, Zienkiewicz and Cheung generalized the Finite Element Method so it could be used for any variational problem. These works expanded the use of FEM for more fields of engineering than just the structural problems [24; 25]. FEM can be used to determine the stress intensity factor of a crack but difficulties arise when using it in Fracture Mechanics problems. However, since the crack is a discontinuity, the resolution isn't as simple as in its usual applications. In traditional FEM, introducing a discontinuity in the mesh requires a remeshing process to ensure that the element edges are aligned with the discontinuity [26].

3.2 Overview of the Finite Element Method

As already said, the Finite Element Method transforms the domain into a mesh of elements. These elements can have very different shapes. For example, on 2-D problems, they can be square, rectangular or triangular, while on 3-D problems can be tetrahedral, pentahedral or hexahedral.

The nodal functions, also called shape functions, must have the value 1 in their node and 0 in all the other nodes [17]. In any point inside each element (and in its boundary), the sum of all their shape functions must be 1. The value of a shape function outside of his element must be 0. This means that a shape function has no influence outside its element. These shape functions preserve the continuity of the body, because they are a partition of unity method. Inside each element, the properties are a weighted average between its position and the value in each node. The weight of each node is equal to the value of its shape function in the point in study. This means that for an 1-D element with two nodes, the polynomial degree must be 1. If the element has three nodes, the polynomial degree must be 2. For an 1-D element with N nodes, the polynomial degree must be $N - 1$.

Figure 3.1 shows a body with 4 elements, each element with 2 nodes, and their shape functions. Since this is a partition of unity method, the sum of all the functions is 1 in all the points of all the elements. For an element, the value of a shape function varies inside it and is 0 outside of it. For example, the second node belongs to the element 1 and 2. Its shape function $\eta_{\Omega 2}$ is 1 in node 2 and changes inside element 1 and element 2, being its value 0 in the first node, in the third node and in all the other elements.

Since the Finite Element Method is based on continuous functions, it carries a great amount of limitations when facing a discontinuity such as a crack [26]. There are some approaches to this problem. The most common are:

- mesh refinement around the crack;
- pay special attention to nodes position around the crack, placing the nodes in such a way that the crack won't cut any element. The discontinuity is created by the presence of nodes in each side of the crack;
- remesh on each iteration, so that the crack propagation doesn't cut any element.

However, most of the time, these processes are laborious and difficult, requiring a lot of time and computational resources [19].

Another approach is to incorporate a meshless method on the finite element program. The eXtended Finite Element Method (XFEM) detects where the discontinuity is and adds this information to the surrounding nodes. It adds discontinuous functions, which simulate the discontinuity inside the element. Using this type of method, there is no need to worry about the mesh or the type of discontinuity on the problem.

3.2.1 FEM formulation

Considering a body in an equilibrium state, subjected to an external force and with a displacement boundary condition as shown in Figure 3.2. The strong form of the equilibrium equation is given by

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \text{ in } \Omega, \quad (3.1)$$

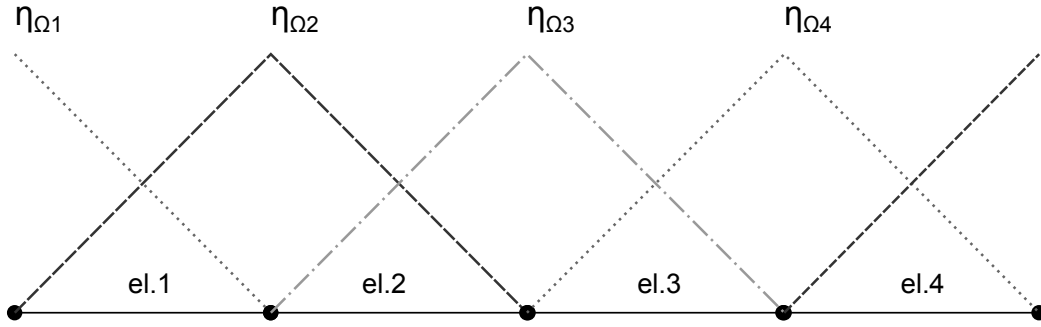


Figure 3.1: Shape functions for four elements with two nodes each.

where $\boldsymbol{\sigma}$ is the stress tensor, \mathbf{b} is the external body force and Ω is the open set of the domain. The body has the following boundary conditions:

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \text{ on } \Gamma_t \quad (3.2)$$

and

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u, \quad (3.3)$$

where \mathbf{n} is the outside normal unitary vector, Γ_t and Γ_u are, respectively, traction and displacement boundaries, and $\bar{\mathbf{t}}$ is the external traction vector. Considering the material to be linear elastic, the relation between strain and displacement is given by

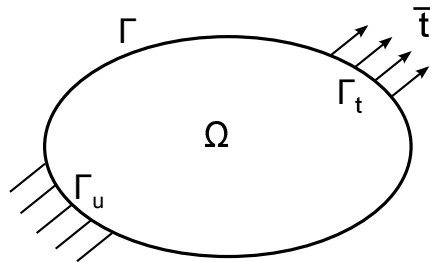
$$\boldsymbol{\varepsilon} = \nabla \mathbf{u}, \quad (3.4)$$

where $\boldsymbol{\varepsilon}$ is the linear strain tensor, ∇ is the symmetric part of the displacement gradient and \mathbf{u} the displacement. The multiplication of the strong form of the equilibrium equation (equation 3.1) by test functions is given by [27]

$$\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}) \cdot \delta \mathbf{u} d\Omega + \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} d\Omega = 0, \quad (3.5)$$

where δ is the symbol of the first variation vector. The above equation can be rewritten in

$$\int_{\Omega} \nabla \cdot (\boldsymbol{\sigma} \cdot \delta \mathbf{u}) d\Omega - \int_{\Omega} \boldsymbol{\sigma} : \nabla(\delta \mathbf{u}) d\Omega + \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} d\Omega = 0. \quad (3.6)$$

Figure 3.2: Body Ω .

Using the divergence theorem in the Ω domain, with the symmetry of $\boldsymbol{\sigma}$, one gets [27]

$$\int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta \mathbf{u} d\Gamma - \int_{\Omega} \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} d\Omega + \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} d\Omega = 0. \quad (3.7)$$

3.2.2 Isoparametric elements

The concept of isoparametric element is very important in FEM. It was created to solve the problems caused by the development of shape functions for complex elements, such as elements with several nodes or with curved boundaries. It provides an easy process to create the shape functions for any type of element and enables the use of Gauss integration [28].

Even when dealing with simple elements, such as triangles or rectangles, the isoparametric concept is used because it is much simpler than analysing each element individually. In the isoparametric system, any quadrilateral is represented as a square between the points $(-1, -1)$ and $(+1, +1)$. Figure 3.3 shows how a four nodes quadrilateral element is represented in the isoparametric system.

As said before, all the calculations are done in the same way for any element. In the end, there is just the need to transform the results from the isoparametric element into the original element, using the Jacobian matrix.

In the local system, the ξ and η coordinates vary from -1 to $+1$, taking the value zero on the quadrilateral center. This particular variation (instead of, for example, 0 to 1), was chosen by Irons to facilitate use of the standard Gauss integration formulas [29].

The four-node quadrilateral shown in Figure 3.3 is the simplest element on the quadrilateral family. Its shape functions are:

$$\begin{cases} N_1 = \frac{1}{4}(1 - \xi)(1 - \eta) \\ N_2 = \frac{1}{4}(1 + \xi)(1 - \eta) \\ N_3 = \frac{1}{4}(1 + \xi)(1 + \eta) \\ N_4 = \frac{1}{4}(1 - \xi)(1 + \eta) \end{cases} \quad (3.8)$$

Since this corresponds to a partition of unity method, the sum of its shape functions, at any point inside the element (or in its boundary) must be equal to 1. The value of the displacement at any point inside the element e is given by

$$\mathbf{u}^e = \sum_{i=1}^4 N_i \mathbf{u}_i, \quad (3.9)$$

where i is the node number. The same can be done for the strains and the stresses, resulting in

$$\boldsymbol{\varepsilon}^e = \sum_{i=1}^4 N_i \boldsymbol{\varepsilon}_i, \quad (3.10)$$

which defines the strain at any point inside the element, and

$$\boldsymbol{\sigma}^e = \sum_{i=1}^4 N_i \boldsymbol{\sigma}_i, \quad (3.11)$$

defining the stress at any point of the element.

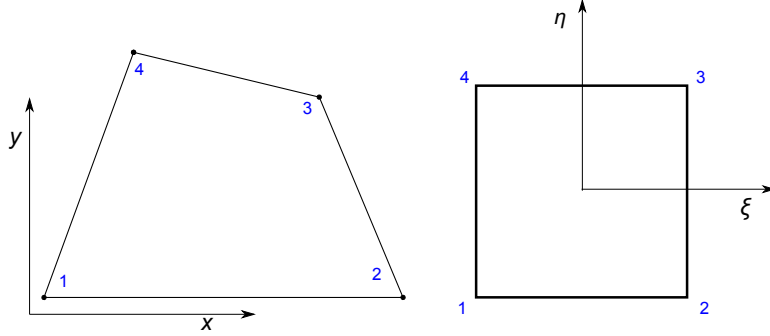


Figure 3.3: Isoparametric representation (right) of a four nodes quadrilateral element defined in Oxy (left).

Jacobian matrix

Partial derivatives of shape functions with respect to the global coordinates x and y are required for the elementary stiffness matrix, strain and stress calculations. Since the shape functions for isoparametric elements depend on the natural coordinates (ξ and η) and not on the global coordinates (x and y), the determination of the cartesian partial derivatives may not be an easy task. The Jacobian matrix may be used to find the relation between the local and global derivatives. The relationship between the global and the local coordinate system corresponds to

$$\begin{bmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial N}{\partial \xi} \\ \frac{\partial N}{\partial \eta} \end{bmatrix} \quad (3.12)$$

and

$$dxdy = \det \mathbf{J} d\xi d\eta, \quad (3.13)$$

where \mathbf{J} is the Jacobian matrix, defined by

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \quad (3.14)$$

Considering the interpolation of the Cartesian coordinates using the shape functions for a four node quadrilateral, one gets [17]

$$\mathbf{J} = \sum_{i=1}^4 \begin{bmatrix} \frac{\partial N_i}{\partial \xi} x_i & \frac{\partial N_i}{\partial \xi} y_i \\ \frac{\partial N_i}{\partial \eta} x_i & \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}. \quad (3.15)$$

3.2.3 Stiffness matrix

One of the main steps in FEM is to calculate the stiffness matrices of the mesh elements. These matrices can be obtained from the strain-displacement matrix (\mathbf{B} operator). It is

a matrix with the x and y derivatives of all the shape functions of the element. For a four node quadrilateral element, it is defined by

$$\mathbf{B} = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix}. \quad (3.16)$$

For an isotropic linear elastic an homogeneous material, the elasticity matrix (\mathbf{D}) is given by

$$\mathbf{D}^{\text{stress}} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \quad (3.17)$$

for a plane stress state, and

$$\mathbf{D}^{\text{strain}} = \frac{E}{(1 + \nu)(1 - 2\nu)} \begin{bmatrix} 1 - \nu & \nu & 0 \\ \nu & 1 - \nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}, \quad (3.18)$$

for a plane strain state.

The equation

$$\mathbf{K}^e = \int_{A^e} \mathbf{B}^T \mathbf{D} \mathbf{B} t dA^e \quad (3.19)$$

corresponds to the integral form of the elementary stiffness matrix, where \mathbf{D} is the elasticity matrix and t the thickness of the material. Normally, on 2-D problems, value t is considered as being unitary. This means the whole problem must be modelled *per* thickness unity. The elasticity matrix is equal to matrix 3.17 or 3.18, depending if the problem corresponds to a plane stress or a plane strain state.

3.2.4 Element integration

From the previous section, in order to obtain the element stiffness matrix, there is the need to compute an area integral. Normally, the computation of integrals can be cumbersome. In this way, one of the most common solutions to overcome this problem, in computational mechanic is the use the Gauss-Legendre quadrature. This means that the integral defined in equation 3.19 can be transformed into a sum. This is very important because it transforms a continuous integral in a sum of weighted functions, making the computation much simpler. This approximation for the local coordinate system corresponds to [17]

$$\int_{-1}^{+1} \int_{-1}^{+1} f(\xi, \eta) d\xi d\eta \cong \sum_r^{n_r} \sum_s^{n_s} f(\xi_r, \eta_s) w_r w_s, \quad (3.20)$$

where n_r and n_s are the number of integration points to be used in the $O\xi$ -direction and in the $O\eta$ -direction, respectively. $f(\xi_r, \eta_s)$ is a generic function defined in the local coordinates (ξ, η) . Finally, w_r and w_s are the weights related with the chosen point. For

the case of simple geometries, like linear squares or rectangles, four Gauss points are enough to get precise results. Table 3.1 shows the local coordinates and weights of the Gauss points used in the integration of the stiffness matrices.

With the combination of equations 3.13, 3.19 and 3.20, the following equation for the element stiffness matrix is obtained [17]:

$$\mathbf{K}^e \cong \sum_{s=1}^{n_s} (\mathbf{B}^T \mathbf{D} \mathbf{B} t \det \mathbf{J})_s^e w_s, \quad (3.21)$$

where n_s is the number of gauss points, $\det \mathbf{J}$ is the determinant of the element Jacobian matrix and w_s is the weight for each Gauss point.

3.2.5 Problem solving

Normally, the main task of a FEM program is to obtain the nodal displacement and external force vectors. This task is performed, considering the system stiffness, by the relation

$$\mathbf{K} \mathbf{u} = \mathbf{f}. \quad (3.22)$$

In order to solve this system of linear equations, boundary conditions must be applied. Thus, one must defined not only which nodes are fixed in the x -direction and in the y -direction but also the external forces. Then the global system can be reduced, and, with the inversion of the reduced global stiffness matrix, the problem can be solved.

3.2.6 Strain and stress

In elastic materials, stresses are directly related to strains at each point through

$$\boldsymbol{\sigma} = \mathbf{D} \boldsymbol{\varepsilon}. \quad (3.23)$$

This means that the stress computation procedure begins with the strain computation and that the accuracy of stresses is related to the accuracy of strains. The strains can only be obtained after the resolution of the global equation system which provides the nodal displacements (equation 3.22). That is why this is considered a post-processing step since it can only be done after the calculation of the nodal displacements. In order to calculate strains and stresses a loop over all the elements is performed. The strain at any point inside each element is obtained from the nodal displacements by

$$\boldsymbol{\varepsilon} = \mathbf{B} \mathbf{u}^e, \quad (3.24)$$

Point	ξ	η	w
1	$-\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	1
2	$\frac{\sqrt{3}}{3}$	$-\frac{\sqrt{3}}{3}$	1
3	$\frac{\sqrt{3}}{3}$	$\frac{\sqrt{3}}{3}$	1
4	$-\frac{\sqrt{3}}{3}$	$\frac{\sqrt{3}}{3}$	1

Table 3.1: Coordinates and weights of the 4 Gauss points in a square element.

where $\boldsymbol{\varepsilon}$ is the point strain, \mathbf{B} is the strain-displacement matrix and \mathbf{u}^e is the vector with the nodal displacements of the element. All the coordinates used for the \mathbf{B} matrix are the isoparametric coordinates of the point in study. The corresponding stresses are given by

$$\boldsymbol{\sigma} = \mathbf{DBu}^e. \quad (3.25)$$

It is important to note that the strains and stresses computed at adjacent elements will not generally be continuous. For this reason, some form of stress and strain average should be used to improve the accuracy of the results, which can be performed by two different ways:

1. directly on the nodes of the element using their isoparametric coordinates on the shape functions;
2. on the Gauss integration points used in the element stiffness integration and then extrapolated to the element node points.

Both ways work fine for rectangular and triangular elements. However, for quadrilateral elements which geometry departs substantially from the rectangular shape, the second approach gives better results [30]. In this work, the stress is calculated directly on the nodes, since the program only uses square elements. The first method is used because it requires one less step to get the same results. If, for some reason, one decides to use another kind of element, the stress and strain must be determined on the Gauss points. Only after that step, their values can be extrapolated to the nodes of the element.

3.3 Finite Element Method program

Nowadays, there is a quite large amount of commercial and personal software that use FEM to perform numerical simulation. They can have different interfaces and be made for different fields of engineering, but the main steps inside each FEM program are almost the same.

The first thing a FEM program does is the discretization of the domain and build the mesh connectivity of the domain. In this step, nodes and elements are created according to the type of mesh. The relationship between nodes, elements and their positions is also created. Afterwards, there is the assembly, where the global stiffness matrix \mathbf{K} is created. Firstly, the stiffness matrix is created for each element. Then, all the element stiffness matrices are assembled in the global stiffness matrix. This is done based on the relations defined by the connectivity for each element.

After the creation of the global stiffness matrix, the loads and displacement boundary conditions are applied. At this point, the global system is reduced so it can be solved and it is possible to get the displacements and therefore strains and stresses for each node. Figure 3.4 shows a flowchart of a typical FEM program:

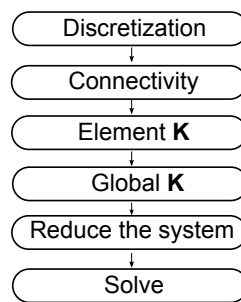


Figure 3.4: Typical flowchart of a FEM software.

This page was intentionally left blank.

Chapter 4

Extended Finite Element Method

4.1 Introduction

The eXtended Finite Element Method (XFEM) is a numerical method for approximation of solutions that involve either strong or weak discontinuities, such as jumps, kinks or high gradients in some part of the domain. It is a FEM generalization that enables the incorporation of local enrichment functions which are an approximation of the field. The XFEM is normally used in the simulation of cracks, shear bands, dislocations, multiphase problems, two-fluid flows and fluid-structure interaction [31]. In these cases, the usual Finite Element Method (FEM) relies on a specific mesh and on a constant remeshing process to model the discontinuity, meaning that, during the crack growth, a new mesh is created for each iteration. This results in the creation of new shape functions and the loss of data of the previous iterations, causing an high computational cost. However, in the XFEM simulation, there is no need to manipulate the mesh in order to get the results. These are achieved with the enrichment of some of the elements of the usual FEM [32].

In 1999, Belytschko and Black introduced the concept of local enrichment, using the partition of unity concept, for modelling cracks [33]. This enrichment could be done with any generic function that correctly modelled the field and was multiplied by the standard FEM function. This method used only a crack tip function to enrich the field through the crack length, eliminating the use of a remesh technique present in FEM. However, it still needed a minimal remesh when dealing with cracks with complex geometries. In the same year, Moës proposed the use of the Heaviside function as a way to model the crack when it is far away from the crack tip [34], which was a progress in the modulation of crack growth without the use of a remesh technique.

In 2000, Daux modelled holes and intersecting cracks using XFEM [35] and Sukumar developed the method for three-dimensional crack problems [36].

In 2001, Belytscho modelled tangential discontinuities [37]; Stolarska incorporated the use of level-sets and fast marching for two-dimensional evolving cracks [38]; Sukumar introduced the use of the level-set technique to model weak discontinuities like material interfaces (usually known as bimetals) [39] and Wells and Sluys applied the Heaviside enrichment to cohesive crack problems [40].

In the following years, some improvements were made in the modelling of multiple cracks in 3 dimensions. Moës used the level-set method to model the crack growth of non-planar cracks [41]; Wells used the partition of unity enrichment to overcome the

volumetric locking in low-order finite elements [42]; Zi and Belytschko used a different enrichment function that could be applied to all the elements (crack tip and completely cut) and introduced the shifted enrichment [43]; at the same time, Sukumar, on one side, and Chop, on another one, developed, in their works, a way of using the fast marching method to model the growth of multiple coplanar cracks [44; 45].

With his work, in 2003, Sukumar studied the crack growth in different materials microstructure, using the XFEM [46].

In 2007, Duffot presented an overview of the level-set method for two and three-dimensional crack propagation problems [47]; Asferg showed that the enrichment function developed by Zi and Belytschko wasn't capable to model equal stresses in both sides of the crack, proposing an alternative with his work [48]; Meschke and Dumstoft used a method based on global energy within the XFEM framework to model cohesive cracks in brittle materials [49].

Although the first researches had been related to structural mechanics (mainly fracture), XFEM can also be used in other fields of engineering like fluids mechanics and materials science [27]. In 2001, Wagner proposed a computational model for rigid particles in stokes flow [50]; in 2002, Merle, Dolbow and Chessa modelled moving phase bounding problems using the XFEM and the level-set method [51; 52].

Recently, the XFEM has gained an enormous attention and it has been introduced in recent commercial software.

4.2 Modelling strong discontinuities

In the eXtended Finite Element Method, any generic function can be used, as an enrichment function, if it represents the behaviour of the approximation field.

A strong discontinuity has a jump in its field, meaning that, in such cases, a step function like the Heaviside function can be used as an enrichment function of the approximation spaces [53]. The enriched function results in a multiplication of the Heaviside function with the standard FEM shape function.

For a one-dimensional case, the nodal XFEM approximation of the field variable u_{sd} is given by

$$u_{sd} = \sum_{i=1}^N N_i u_i + \sum_{j=1}^{n^e} N_j H(x) a_j, \quad (4.1)$$

where the first sum belongs to the standard FEM and the second one is the XFEM part. N is the number of nodes in the element, n^e is the number of enriched nodes in the element, N_i is the nodal shape function of node i , u_i is the nodal variable, N_j is the nodal shape function of the enriched j node, $H(x)$ is the value of the nodal Heaviside function and a_j is a new variable that doesn't have any physical meaning.

Contrary to the standard FEM method, u_i doesn't correspond to the real displacement value of an enriched node i .

4.3 Modelling weak discontinuities

A weak discontinuity is like a kink in the field. This time, the enrichment function is continuous and the discontinuity is presented only in its derivative [53]. Normally, when

modelling a field with a weak discontinuity, a ramp function, like the following absolute distance function, is used:

$$\chi(\mathbf{x}) = |\Delta(\mathbf{x})|, \quad (4.2)$$

where $\Delta(\mathbf{x})$ is a distance function between any point of the domain and the location of the discontinuity.

Again, for a one-dimensional case, the nodal XFEM approximation of the field variable u is given by

$$u_{\text{wd}} = \sum_{i=1}^N N_i u_i + \sum_{j=1}^{n^e} N_j \chi(x) a_j, \quad (4.3)$$

where, again, the first sum belongs to the standard FEM and the second one is the XFEM part. N is the number of nodes of the element, n^e is the number of enriched nodes in the element, N_i is the nodal shape function of node i , u_i is the nodal variable, N_j is the nodal shape function of the enriched j node, $\chi(x)$ is the value of the absolute distance function and a_j is a new variable that doesn't have any physical meaning.

4.4 XFEM formulation

Considering a body used in Subsection 3.2.1, but now with a crack in its interior as shown by Figure 4.1, with the following boundary conditions

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \text{ on } \Gamma_t \cup \Gamma_c \quad (4.4)$$

and

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u, \quad (4.5)$$

where \mathbf{n} is normal unitary vector, Γ_t , Γ_c and Γ_u are, respectively, traction, crack and displacement boundaries and $\bar{\mathbf{t}}$ is the external traction vector.

Using the divergence theorem in the Ω domain of equation 3.6 of the FEM formulation and with the symmetry of $\boldsymbol{\sigma}$, a new term appears:

$$\int_{\Gamma_c^+ \cup \Gamma_c^-} \bar{\mathbf{t}} \cdot \delta \mathbf{u} d\Gamma, \quad (4.6)$$

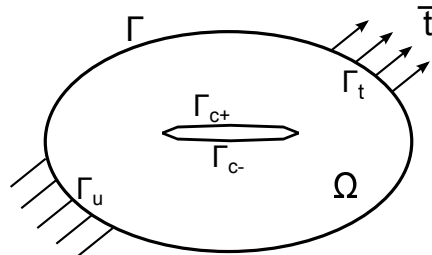


Figure 4.1: Body with an internal crack and corresponding boundary conditions.

which wasn't present in equation 3.7, meaning that this equation now corresponds to

$$\int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta \mathbf{u} d\Gamma + \int_{\Gamma_t^+ \cup \Gamma_t^-} \bar{\mathbf{t}} \cdot \delta \mathbf{u} d\Gamma - \int_{\Omega} \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} d\Omega + \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} d\Omega = 0. \quad (4.7)$$

Nevertheless, since this new term results in a zero integral, the weak form (principle of virtual work) also corresponds, in XFEM, to [27]

$$\int_{\Omega} \boldsymbol{\sigma} : \delta \boldsymbol{\varepsilon} d\Omega = \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta \mathbf{u} d\Gamma + \int_{\Omega} \mathbf{b} \cdot \delta \mathbf{u} d\Omega, \quad (4.8)$$

The divergence theorem was used in the previous equation for a continuous domain, meaning that the crack needs to be defined as an internal boundary of the integration domain. In the weak form of the finite element, the domain must be divided into several subdomains that obey to the same requirements of the continuous problem, providing, this way, the equivalence between the strong and weak form of the problem [27].

4.4.1 Discrete equations

The discretization of equation 4.8 results in the following discrete system of linear equilibrium equations [1; 27]

$$\mathbf{K} \mathbf{u} = \mathbf{f}, \quad (4.9)$$

where \mathbf{K} is the global stiffness matrix, \mathbf{u} is the vector of the nodal degrees of freedom and \mathbf{f} is the external force vector. The elementary stiffness matrix and the force vector are calculated and assembled into their global counterparts through the usual assembly procedure. For each element, \mathbf{K}^e is defined as

$$\mathbf{K}^e = \begin{bmatrix} \mathbf{K}_{ij}^{uu} & \mathbf{K}_{ij}^{uh} & \mathbf{K}_{ij}^{ua} \\ \mathbf{K}_{ij}^{hu} & \mathbf{K}_{ij}^{hh} & \mathbf{K}_{ij}^{ha} \\ \mathbf{K}_{ij}^{au} & \mathbf{K}_{ij}^{ah} & \mathbf{K}_{ij}^{aa} \end{bmatrix}, \quad (4.10)$$

where the submatrices from the previous equation are defined as [1; 27]

$$\mathbf{K}_{ij}^e = \int_{\Omega^e} (\mathbf{B}_i^\alpha)^T \mathbf{D} (\mathbf{B}_j^\beta) d\Omega \quad (\alpha, \beta = u, a, h), \quad (4.11)$$

where, considering that the standard FEM shape functions are used, the three different gradient \mathbf{B} operators are

$$\mathbf{B}_i^u = \mathbf{B}_{std_i}^u = \begin{bmatrix} (N_{std_i})_{,x} & 0 \\ 0 & (N_{std_i})_{,y} \\ (N_{std_i})_{,y} & (N_{std_i})_{,x} \end{bmatrix}, \quad (4.12)$$

$$\mathbf{B}_i^h = \mathbf{B}_{enri_i}^h = \begin{bmatrix} (HN_{std_i})_{,x} & 0 \\ 0 & (HN_{std_i})_{,y} \\ (HN_{std_i})_{,y} & (HN_{std_i})_{,x} \end{bmatrix} \quad (4.13)$$

and

$$\mathbf{B}_i^{a_l} = \mathbf{B}_{enri_i}^{a_l} = \begin{bmatrix} (F_l N_{std_i})_{,x} & 0 \\ 0 & (F_l N_{std_i})_{,y} \\ (F_l N_{std_i})_{,y} & (F_l N_{std_i})_{,x} \end{bmatrix}, \quad (4.14)$$

where N_{std_i} is the standard FEM shape function, H is the Heaviside enrichment function, F_l is the asymptotic enrichment function and l is the function number varying from 1 to 4. The external nodal force vector is

$$f_i = \{f_i^u, f_i^h, f_i^{a_1}, f_i^{a_2}, f_i^{a_3}, f_i^{a_4}\}, \quad (4.15)$$

where the three different subvectors are [1; 27]

$$f_i^u = f_{std_i}^u = \int_{\Gamma_t} (N_{std_i})^T \bar{\mathbf{t}} d\Gamma + \int_{\Omega} (N_{std_i})^T \mathbf{b} d\Omega, \quad (4.16)$$

$$f_i^h = f_{std_i}^h = \int_{\Gamma_t} (H N_{std_i})^T \bar{\mathbf{t}} d\Gamma + \int_{\Omega} (H N_{std_i})^T \mathbf{b} d\Omega \quad (4.17)$$

and

$$f_i^{a_l} = f_{std_i}^{a_l} = \int_{\Gamma_t} (F_l N_{std_i})^T \bar{\mathbf{t}} d\Gamma + \int_{\Omega} (F_l N_{std_i})^T \mathbf{b} d\Omega. \quad (4.18)$$

4.4.2 Element partitioning and integration

In Finite Element Method, the integration process of the local stiffness matrix is done with the equation 3.20. On the other hand, XFEM enrichment is done with approximation functions that incorporate the discontinuity into the element domain. Since the Gauss quadrature can only be applied in the integration of functions that are continuous inside the quadrature domain, it can't be used on the enriched elements in the same way it is used on the non-enriched ones. However, there is an easier solution for this problem. If the elements are split by the crack, all the enrichment functions inside each subelement become continuous. This partition of the element into several subdomains is very important since, if done correctly, each subdomain becomes continuous, enabling the use of the regular Gauss quadrature in it. This partition of the integral domain is demonstrated by

$$F(x) = \int_{\Omega} f(x) dx = \int_{\Omega^+} f(x) dx + \int_{\Omega^-} f(x) dx, \quad (4.19)$$

where $f(x)$ is a discontinuous function in the Ω domain. With such condition, the element can be integrated using the usual Gauss quadrature. The integration process of an enriched element is done numerically by

$$F(x) = \sum_{\Omega_e} \sum_{\Omega_s} f(gp) gw, \quad (4.20)$$

where the standard FEM loop over the element was replaced by a loop over each subdomain, Ω_e is the entire element domain and Ω_s is the subdomain.

Attending on all the geometries the cut elements can have, it is very appealing to have an easier way to split them. For this reason, the elements are usually divided in triangles.

They can be divided into other kinds of elements, mainly rectangular and degenerated quadrilaterals. However, on the weirdest geometries, it is much simpler to divide and integrate the triangular elements than the quadrilateral ones. The previous expression is very identical to the FEM integration, but, as already said, the integration is done in the Gauss points of each triangle individually and “added” to the element stiffness matrix.

Ordinary FEM processes use a remeshing method to incorporate the crack into the elements edges. This is done as a pre-processing step and is the only way a crack can be modelled by FEM. However, it changes the original mesh and creates lots of new degrees of freedom. But that remeshing process is very different from the subdivision used by XFEM: since the subdivision is only made for the integration process, it doesn’t add any degrees of freedom to the problem unlike in FEM; the construction of the triangles in one element doesn’t interfere with the triangles of the other elements, contrary to FEM, in which all the triangles need to be linked together.

4.5 Enrichment functions

There are two different functions used to model a crack in XFEM on a 2-D domain: Heaviside and asymptotic functions.

The Heaviside function is used to enrich the completely cut elements and it creates the discontinuity.

The asymptotic function is used to enrich the elements near the crack tip that aren’t completely cut by the crack. It is composed by 4 individual functions.

Each node gets a new degree of freedom for each function, enriching it. This means that, when modelling a two-dimensional crack, each node will get 2 additional degrees of freedom, if enriched with the Heaviside function, and 8 additional degrees of freedom, if enriched with the asymptotic function.

4.5.1 Heaviside function

The Heaviside and the step function are usually used to enrich the elements that are completely cut by the crack. Those functions model properly the jump caused in the displacement field by the crack.

Originally, the first enrichment functions used were step functions defined by

$$S(\mathbf{x}^*) = \begin{cases} 1 & \text{for } (\mathbf{x} - \mathbf{x}^*) \cdot \mathbf{n} > 0 \\ 0 & \text{for } (\mathbf{x} - \mathbf{x}^*) \cdot \mathbf{n} < 0 \end{cases}, \quad (4.21)$$

where \mathbf{n} is the normal crack vector, \mathbf{x} is a point inside the crack and \mathbf{x}^* is the point that is being analysed.

Nowadays, the sign Heaviside function is normally used as the enrichment function. It is a step function with the output of 1 or -1 and it is used to enrich the nodes of the elements that are completely cut by the crack, adding the discontinuity to the computer model. The objective is to calculate if a point is above or bellow the crack, using a distance function, and giving it the correct Heaviside value. The sign Heaviside function

is given by

$$H(\mathbf{x}^*) = \begin{cases} 1 & \text{if } (\mathbf{x} - \mathbf{x}^*) \cdot \mathbf{n} > 0 \\ 0 & \text{if } (\mathbf{x} - \mathbf{x}^*) \cdot \mathbf{n} = 0, \\ -1 & \text{if } (\mathbf{x} - \mathbf{x}^*) \cdot \mathbf{n} < 0 \end{cases}, \quad (4.22)$$

where \mathbf{n} is the normal crack vector, \mathbf{x} is a point inside the crack and \mathbf{x}^* is the point that is being analysed. The previous equation shows that the enrichment is 1 if a point is above the crack and -1 if a point is under the crack. If a point is inside the crack, its enrichment value is 0 meaning the point isn't enriched.

Applying equation 4.1 to a two-dimensional domain, the displacement, for any point inside an element enriched by the Heaviside function, is given by

$$u^h(x, y) = \sum_{i=1}^N N_i(x, y)u_i + \sum_{j=1}^{n^e} H_j N_j(x, y)a_j, \quad (4.23)$$

where N is the number of nodes in the element; n^e is the number of enriched nodes in the element; $N_i(x, y)$ and N_j are the nodal shape functions, at (x, y) , of node i and j , respectively; u_i is the nodal displacement; H_j is the Heaviside enrichment value of node j and a_j is the displacement of the pseudo node j , which doesn't have any physical meaning.

4.5.2 Asymptotic function

When an element has a crack tip in its interior, only a part of it is completely cut, meaning the Heaviside function can't be used to properly enrich it. In this case, the asymptotic function is used, since it is based on the analytical results of the stress and the displacement field around the crack tip. The crack-tip enrichment consists of 4 functions which incorporate the distance (r) and the angle behaviour of the two-dimensional crack tip displacement field. The crack tip enrichment function vector is given by

$$\mathbf{F}(r, \theta) = \left\{ \sqrt{r} \cos\left(\frac{\theta}{2}\right), \sqrt{r} \sin\left(\frac{\theta}{2}\right), \sqrt{r} \sin\left(\frac{\theta}{2}\right) \sin(\theta), \sqrt{r} \cos\left(\frac{\theta}{2}\right) \sin(\theta) \right\}, \quad (4.24)$$

where $\sqrt{r} \sin\left(\frac{\theta}{2}\right)$ is the only function that creates the discontinuity and all the three other functions just improve the results near the crack tip [7]. The discontinuity is created along the radius for an angle of $\theta = \pm\pi$ with the crack tip vector. As already said, the use of the asymptotic function creates 8 new degrees of freedom, in each node that is enriched by it.

Applying equation 4.3 on a two-dimensional domain, the displacement, for any point inside an element enriched by the asymptotic function, is given by

$$u^{a1}(x, y) = \sum_{i=1}^N N_i(x, y)u_i + \sum_{j=1}^{n^e} \sum_{l=1}^4 F_{lj} N_j(x, y)a_{lj}, \quad (4.25)$$

where N is the number of nodes in the element; n^e is the number of enriched nodes in the element; $N_i(x, y)$ and N_j are the nodal shape function, at (x, y) , of node i and j ,

respectively; u_i is the nodal displacement; F_{l_j} is one of the four asymptotic enrichment values of node j and a_j is the displacement of the pseudo node j , which doesn't have any physical meaning.

Since the partial derivatives of the asymptotic function exist and are different than 0, contrary to the Heaviside derivatives, they must be calculated, so that the enrichment can be done. Those derivatives are used to create the \mathbf{B} operator belonging to the asymptotic enrichment. The partial derivatives of F_a on the global coordinate system are given by

$$\frac{\partial F_a}{\partial x} = \frac{\partial F_a}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial F_a}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (4.26)$$

and

$$\frac{\partial F_a}{\partial y} = \frac{\partial F_a}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial F_a}{\partial \eta} \frac{\partial \eta}{\partial y}, \quad (4.27)$$

where equation 4.26 is the partial derivative in x , equation 4.27 is the partial derivative in y . The derivative of the isoparametric coordinates, according to the global coordinate system, are given by

$$\frac{\partial \xi}{\partial x} = \cos \alpha, \quad \frac{\partial \xi}{\partial y} = \sin \alpha \quad (4.28)$$

and

$$\frac{\partial \eta}{\partial x} = -\sin \alpha, \quad \frac{\partial \eta}{\partial y} = \cos \alpha, \quad (4.29)$$

where α is the angle between the tangential crack tip vector and the vector defined by the crack tip and the point in analysis.

The partial derivatives of the four enrichment functions are given by [1]

$$\frac{\partial F_1}{\partial \xi} = -\frac{1}{2\sqrt{r}} \sin\left(\frac{\theta}{2}\right), \quad (4.30)$$

$$\frac{\partial F_1}{\partial \eta} = \frac{1}{2\sqrt{r}} \cos\left(\frac{\theta}{2}\right), \quad (4.31)$$

$$\frac{\partial F_2}{\partial \xi} = \frac{1}{2\sqrt{r}} \cos\left(\frac{\theta}{2}\right), \quad (4.32)$$

$$\frac{\partial F_2}{\partial \eta} = \frac{1}{2\sqrt{r}} \sin\left(\frac{\theta}{2}\right), \quad (4.33)$$

$$\frac{\partial F_3}{\partial \xi} = -\frac{1}{2\sqrt{r}} \sin\left(\frac{3\theta}{2}\right) \sin(\theta), \quad (4.34)$$

$$\frac{\partial F_3}{\partial \eta} = \frac{1}{2\sqrt{r}} \left[\sin\left(\frac{\theta}{2}\right) + \sin\left(\frac{3\theta}{2}\right) \cos(\theta) \right], \quad (4.35)$$

$$\frac{\partial F_4}{\partial \xi} = \frac{1}{2\sqrt{r}} \cos\left(\frac{3\theta}{2}\right) \sin(\theta) \quad (4.36)$$

and

$$\frac{\partial F_4}{\partial \eta} = \frac{1}{2\sqrt{r}} \left[\cos\left(\frac{\theta}{2}\right) + \cos\left(\frac{3\theta}{2}\right) \cos(\theta) \right]. \quad (4.37)$$

4.6 Detecting enriched elements

Since XFEM uses two types of enrichment functions, a way to detect the nodes around the crack must be developed. The program also needs to detect which elements are completely cut by the crack and which have a crack tip inside, that is, if an element needs to be enriched with either the Heaviside function or the asymptotic function. One solution is to use a level-set function to detect the elements.

The level-set method was developed to track moving interfaces [54]. It represents the discontinuity as a zero level. One side of the interface has a positive level-set and the other a negative level-set.

Since a crack normally divides only a portion of the domain, rather than dividing the whole domain into two distinct parts, it is necessary to use two level-set functions to model the crack [7; 55]:

- A normal level-set function (Φ) – the orientation of this level-set is normal to the crack. This means the value of the normal level-set function is 0 under the crack, positive on one side and negative on the other side;
- A tangential level-set function (Ψ) – this level-set is tangential to the crack tip, creating a border between two distinct domains (the domain with the crack, where the value of the tangential level-set function is negative, and the domain without the crack, where the value is positive). Like the normal level-set, the value is positive on one side, negative on the other and 0 on the border. However, in this case, there is no crack on the border. It is just an imaginary discontinuity between the domain with the crack and the domain without it. In this way, the domain with the crack will be isolated. If it is an interior crack, it will be necessary to define two tangential level-set functions (one for each crack tip).

Figure 4.2 shows the values of the normal and tangential level-set functions around the crack. The crack and the crack tip are defined by

$$\left\{ \begin{array}{l} \text{Crack : } \quad \Phi = 0 \wedge \Psi \leq 0 \\ \text{Crack tip : } \quad \Phi = 0 \wedge \Psi = 0 \end{array} \right. \quad (4.38)$$

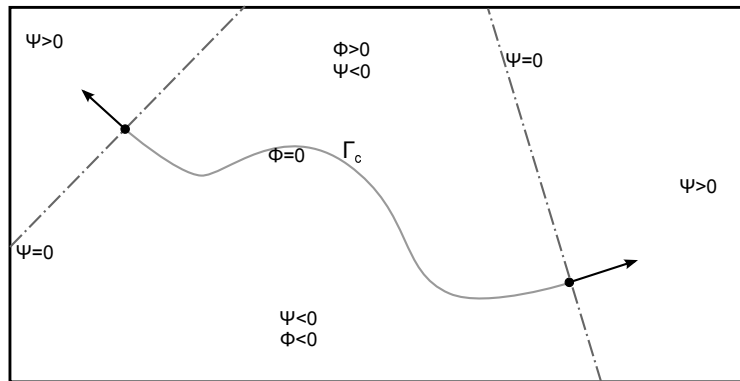


Figure 4.2: Values of the level-set functions.

Knowing these properties, an element that is completely cut by a crack can be found by

$$\Phi_{\min} \Phi_{\max} \leq 0 \wedge \Psi_{\max} < 0, \quad (4.39)$$

and the elements that contain a crack tip can be found by

$$\Phi_{\min} \Phi_{\max} \leq 0 \wedge \Psi_{\min} \Psi_{\max} \leq 0. \quad (4.40)$$

4.7 Blending elements

In XFEM, there are the following three types of elements:

- Non-enriched elements: when no node is enriched. These are the standard FEM elements;
- Enriched elements: when all the nodes are enriched. These elements have additional degrees of freedom created by the enrichment functions;
- Blending elements: these are composed by enriched and non enriched nodes. The enriched nodes have the properties of XFEM elements and the non enriched nodes have the properties of those based in standard FEM analysis.

The blending elements are a consequence of the transition between enriched and non-enriched domains, representing the link between them. Since, in these elements, not all the nodes are enriched, the concept of partition of unity isn't satisfied, meaning that

$$\sum_j N_j(x, y) \neq 1 \quad (4.41)$$

for most of the points inside a blending element. This property isn't very important, since the enrichment displacement is calculated in the enriched element, where the partition of unity is satisfied. However, their stiffness matrix is similar to the stiffness matrix present in the enriched elements, only with less enriched parts because they have less enriched nodes. For example, for a four node quadrilateral element where only two nodes are enriched, the displacement for any point inside the element is given by

$$u_{\text{bld}} = \sum_{i=1}^4 N_i u_i + \sum_{j=1}^2 N_j \chi(\mathbf{x}) a_j, \quad (4.42)$$

where N_i is the nodal shape function and u_i is the nodal displacement of node i , N_j is the nodal shape function of the enriched j node, $\chi(\mathbf{x})$ is the value of the enrichment function and a_j is the displacement of pseudonode j .

4.8 XFEM program

Like already referred, an XFEM program is very similar to a FEM program. However, XFEM has the advantage that the mesh nodes don't need to incorporate the discontinuities, like cracks. After the discretization and the creation of the domain process, the

XFEM searches for a crack, using two different types of level-set functions to find it. After finding the crack, it is necessary to detect which elements will be enriched. The elements completely cut by the crack are enriched with the Heaviside function and the elements that have a crack tip inside are enriched with the asymptotic function. If there is no crack, the procedure is exactly the same as in a FEM program.

When building the stiffness matrix and before the integration process, the enriched elements must be remeshed with triangles that match the crack geometry. In this way, the field becomes regular, continue and can be integrated [7; 55]. The remesh of the elements is done only for the integration process and doesn't add more degrees of freedom (unlike in FEM).

These are the main characteristics of an XFEM program. Everything else is similar to those of a FEM program.

Figure 4.3 shows a diagram of a standard XFEM program. The black-coloured steps are the ones used in both FEM and XFEM programs, while the orange-coloured ones are those used only in a XFEM program.

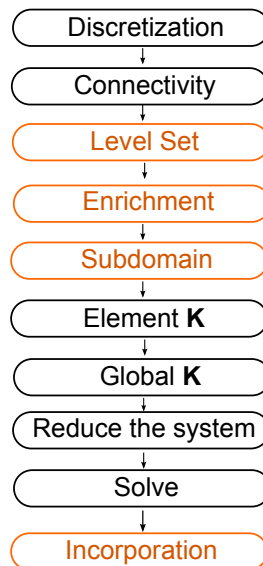


Figure 4.3: Flowchart of a standard XFEM software.

This page was intentionally left blank.

Chapter 5

The XFrac-2D Program

5.1 Introduction

In this chapter, it will be explained the way the eXtended Finite Element Method (XFEM) was implemented. It starts by the explanation on how on an element with a crack is detected, then how the stiffness matrix is created and the numerical integration is performed, and finishing with the way Stress Intensity Factors (SIFs) are determined.

The result of this implementation is the XFrac-2D, which is a software capable of modelling 2-D cracks, using the XFEM, developed in Octave language. The XFrac-2D only works with 2-D quadrilateral domains and it performs the detection of cracks and the calculation of displacements, stresses and SIFs. It can also be used as a standard Finite Element Method (FEM) software. The XFrac-2D manual is presented in Appendix A.

5.2 Selection of enriched elements

The selection of the enriched elements is made with the help of two different level-set functions. Like already referred in the previous chapter, a normal level-set function Φ is used to detect where the crack is. However, this process isn't that simple since the normal level-set Φ function can't detect where the crack ends. The solution adopted is the use of three level-set functions to evaluate all the nodes of each element. Two tangential level-set functions Ψ are introduced in the detecting process. Each one detects where one of the crack tips are and detects if an element is inside the crack domain. This means that it is only possible to know if an element is enriched, and which functions should be used on the enrichment, if the three level-set values are known in each node that belongs to the element. Without this combination of different level-set functions, a large number of elements that aren't cut by the crack would be enriched.

As stated in the previous chapter, the elements whose nodes are according to the criteria of equation 4.39 are detected as completely cut elements and the elements which nodes are according to criteria 4.40 are detected as crack tip elements.

Both level-set functions are determined based on the crack tip coordinates. This means that for each crack segment there are two different values for the tangential level-set functions Ψ , one calculated with a vector tangent to the first crack tip and the other calculated with the vector tangent to the second crack tip. The normal level-set function Φ is also affected by these different vectors. However, since this level-set function is only

important for the creation of the gradient, the only consequence is in its signal (which is almost irrelevant). That is why it isn't necessary to be calculated twice.

The three level-set functions are calculated for each node of each element. The first thing to do, before the calculation of all the level-set functions, is to get the two unitary vectors tangent to each crack tip and the third unitary vector normal to the crack segment. Let's call the tangent vectors \mathbf{t}_1 and \mathbf{t}_2 and the normal vector \mathbf{n} . That direction vectors are obtained by the ratio between their coordinates and their norm, like it is shown by

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|}, \quad (5.1)$$

where \mathbf{v} is the direction vector of vector \mathbf{u} .

The next step, to calculate the level-set value in a node, is to get the distance vector (x, y) between each crack tip and that node. The two tangential level-set Ψ values for any node can be obtained by

$$\Psi_a = \mathbf{t}_a \cdot \mathbf{d}_a, \quad (5.2)$$

where a can have the value of 1 or 2, and \mathbf{d} is the distance vector between the node and one of the crack tips. On the other hand, the normal level-set Φ value can be obtained by

$$\Phi = \mathbf{n} \cdot \mathbf{d}_2. \quad (5.3)$$

This equation is very similar to equation 5.2. The main difference is that \mathbf{n} is the crack normal vector, which results from a 90° rotation of \mathbf{t}_2 . \mathbf{t}_2 was used instead of \mathbf{t}_1 so that the Φ level-set function has always a positive value above the crack. That relation is expressed by

$$\mathbf{n} = \begin{bmatrix} \cos(90) & -\sin(90) \\ \sin(90) & \cos(90) \end{bmatrix} \mathbf{t}_2^T = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{t}_2^T. \quad (5.4)$$

The Ψ and Φ level-set values defined in equations 5.2 and 5.3, respectively, are just the scalar product of two vectors, in which one of them is the distance vector and the other the direction vector of each level-set.

All of this can only be used for a straight crack composed by just two points. Most of the time, it doesn't work for problems where the crack goes through more than two points because it has more than one normal level-set Φ function and those functions start to mess with each other. The resolution adopted for these type of cases is the split of the crack vector in simpler vectors between only two points and analyse each vector individually. This process is explained with more detail a bit further in this document.

5.3 B operator

The \mathbf{B} operator is where the enrichment is done. In the XFEM analysis, it is composed by two parts: one is the standard \mathbf{B} operator of the FEM and the other is the enriched \mathbf{B} operator. The enriched part, depending on which discontinuities are being modelled, can have several different enrichments. In a crack case, it has two kinds of enrichment, as it can be seen in

$$\mathbf{B} = [\mathbf{B}_{\text{std}} \ \mathbf{B}_{\text{enri}}] = [\mathbf{B}_{\text{std}} \ \mathbf{B}_H \ \mathbf{B}_a], \quad (5.5)$$

where \mathbf{B}_{std} is the standard \mathbf{B} operator from the FEM analyses and \mathbf{B}_H and \mathbf{B}_a are the \mathbf{B} operators with the Heaviside and asymptotic enrichment, respectively. The following subsections show how to get \mathbf{B}_H and \mathbf{B}_a from the enrichment functions.

5.3.1 Heaviside enrichment

Since the Heaviside function can only have the value of 1 or -1 , its partial derivative, on the positive and negative domain, is always 0. Knowing this, the partial derivative of the enrichment function, $F = HN$, for x and y corresponds to

$$\frac{\partial F}{\partial a} = H \frac{\partial N}{\partial a}, \quad (5.6)$$

where a represent either x or y variables in the global coordinate system. The assembly, in the Heaviside enriched \mathbf{B} operator (\mathbf{B}_H), is done exactly in the same way as in FEM, being defined by

$$\mathbf{B}_H = \begin{bmatrix} H \frac{\partial N_i}{\partial x} & 0 \\ 0 & H \frac{\partial N_i}{\partial y} \\ H \frac{\partial N_i}{\partial y} & H \frac{\partial N_i}{\partial x} \end{bmatrix}, \quad (5.7)$$

where i is the number of the node that is being enriched. This means that this process must be repeated for all the nodes enriched by the Heaviside function, making the size of \mathbf{B}_H as big as 3×8 , when all the four nodes are enriched. \mathbf{B}_H has the same values of the corresponding function in \mathbf{B} , but some of them have the opposite sign because of the Heaviside function.

5.3.2 Asymptotic enrichment

The asymptotic enrichment is done in the same way as the previous one; however, it is a little more complicated. Firstly there is the need to get the partial derivatives of the four enrichment functions. Starting with the equations 4.26 and 4.27, and using equations 4.28 to 4.37, one gets the following derivatives for each asymptotic function:

$$\frac{\partial F_1}{\partial x} = \frac{1}{2\sqrt{r}} \left[-\sin\left(\frac{\theta}{2}\right) \cos(\alpha) - \cos\left(\frac{\theta}{2}\right) \sin(\alpha) \right], \quad (5.8)$$

$$\frac{\partial F_1}{\partial y} = \frac{1}{2\sqrt{r}} \left[-\sin\left(\frac{\theta}{2}\right) \sin(\alpha) + \cos\left(\frac{\theta}{2}\right) \cos(\alpha) \right], \quad (5.9)$$

$$\frac{\partial F_2}{\partial x} = \frac{1}{2\sqrt{r}} \left[\cos\left(\frac{\theta}{2}\right) \cos(\alpha) - \sin\left(\frac{\theta}{2}\right) \sin(\alpha) \right], \quad (5.10)$$

$$\frac{\partial F_2}{\partial y} = \frac{1}{2\sqrt{r}} \left[\cos\left(\frac{\theta}{2}\right) \sin(\alpha) + \sin\left(\frac{\theta}{2}\right) \cos(\alpha) \right], \quad (5.11)$$

$$\frac{\partial F_3}{\partial x} = \frac{1}{2\sqrt{r}} \left[-\sin\left(\frac{3\theta}{2}\right) \sin(\theta) \cos(\alpha) - \left[\sin\left(\frac{\theta}{2}\right) + \sin\left(\frac{3\theta}{2}\right) \cos(\theta) \right] \sin(\alpha) \right], \quad (5.12)$$

$$\frac{\partial F_3}{\partial y} = \frac{1}{2\sqrt{r}} \left[-\sin\left(\frac{3\theta}{2}\right) \sin(\theta) \sin(\alpha) + \left[\sin\left(\frac{\theta}{2}\right) + \sin\left(\frac{3\theta}{2}\right) \cos(\theta) \right] \cos(\alpha) \right], \quad (5.13)$$

$$\frac{\partial F_4}{\partial x} = \frac{1}{2\sqrt{r}} \left[-\cos\left(\frac{3\theta}{2}\right) \sin(\theta) \cos(\alpha) - \left[\cos\left(\frac{\theta}{2}\right) + \cos\left(\frac{3\theta}{2}\right) \cos(\theta) \right] \sin(\alpha) \right] \quad (5.14)$$

and

$$\frac{\partial F_4}{\partial y} = \frac{1}{2\sqrt{r}} \left[-\cos\left(\frac{3\theta}{2}\right) \sin(\theta) \sin(\alpha) + \left[\cos\left(\frac{\theta}{2}\right) + \cos\left(\frac{3\theta}{2}\right) \cos(\theta) \right] \cos(\alpha) \right], \quad (5.15)$$

where α is the angle between the crack tip vector and the x axis, r is the radius from the crack tip until the Gauss point, and θ is the angle between the r vector and the crack tip vector.

All this partial derivatives are extremely important since, unlike for the Heaviside function, they are different than 0. This means that the calculation of the \mathbf{B} operator is much more complex than it was for the Heaviside enrichment. It is calculated, for any node using any of the four functions, by

$$\mathbf{B}_a = \begin{bmatrix} F_a \frac{\partial N_i}{\partial x} + \frac{\partial F_a}{\partial x} N_i & 0 \\ 0 & F_a \frac{\partial N_i}{\partial y} + \frac{\partial F_a}{\partial y} N_i \\ F_a \frac{\partial N_i}{\partial y} + \frac{\partial F_a}{\partial y} N_i & F_a \frac{\partial N_i}{\partial x} + \frac{\partial F_a}{\partial x} N_i \end{bmatrix}, \quad (5.16)$$

where i and a can have any value between 1 and 4. N_i is the shape function for the enriched node, a is the enrichment function number, being F_a the enrichment function. In each enriched node, there is always the need to get the \mathbf{B}_a for all the four enrichment functions. This means that, for each enriched node in the element, the \mathbf{B}_a size is 3×8 . If all the four nodes of an element are enriched with the asymptotic function, \mathbf{B}_a size is as big as 3×32 .

5.3.3 Shifted enrichment

The standard XFEM approximation doesn't fulfil the Kronecker property. The shifting of the enrichment functions causes it to have the value zero at each enriched node. This is a desired property that helps applying the Dirichlet boundary condition, meaning the computed unknowns q_i are the values of the displacement at node i . The shifted value of the enrichment functions is calculated by

$$\Psi_{\text{shift}}(x) = \Psi(x) - \Psi(x_j), \quad (5.17)$$

where $\Psi_{\text{shift}}(x)$ is the shifted enrichment value, $\Psi(x)$ is the Gauss point enrichment value, $\Psi(x_j)$ is the enrichment value for the node in analysis and Ψ is the enrichment function. $\Psi(x)$ and $\Psi(x_j)$ enrichment values must be firstly converted into the crack tip coordinate system. Only the $\Psi_{\text{shift}}(x)$ value is used to perform the enrichment of the \mathbf{B} operator. This shifting method doesn't affect the partial derivatives of the enrichment functions.

5.4 Stiffness matrix

In the eXtended Finite Element Method (XFEM), the stiffness matrix is built exactly in the same way as in the Finite Element Method (FEM). Both methods use

$$\mathbf{K}^e = \sum_{s=1}^{n_s} (\mathbf{B}^T \mathbf{D} \mathbf{B} t \det \mathbf{J})_s^e w_s \quad (5.18)$$

to build the stiffness matrix of each element. The major difference between them relies on the size of the \mathbf{B} operator. As discussed in the previous section, the size of the \mathbf{B} operator is equal in FEM and XFEM for unenriched elements (3×8). However, for the enriched elements, its size can vary between 3×10 and 3×40 . As shown in equation 5.5, the \mathbf{B} operator has always a part from the standard finite element method and a second part from the enrichment. Depending on the type of enrichment in the element, the size of \mathbf{K}^e can be between 8×8 , for the unenriched elements, up to 40×40 , for the crack tip enriched elements.

In the XFEM stiffness matrix, there is always a part that belongs to the standard FEM analysis and another part that is created by the new degrees of freedom from XFEM. Figure 5.1 shows how the information is stored inside the stiffness matrix. There, matrix \mathbf{A} is the stiffness matrix from standard FEM, matrices \mathbf{B} and \mathbf{C} are the stiffness matrices with the relation between the FEM degrees of freedom and the XFEM degrees of freedom, and matrix \mathbf{D} is the stiffness matrix with the relation between the XFEM degrees of freedom. This kind of organization is only achieved when the vector of the degrees of freedom has first the real and, only after, the auxiliary degrees of freedom. If that doesn't happens, the stiffness matrix is exactly the same but the order and position are changed. This organization is very useful, since it helps when one needs to check the data inside one of the matrices. In this context, in this work, it was adopted for the elementary and the global stiffness matrices.

5.4.1 Assembly

After the creation of the elementary stiffness matrices, it is necessary to assemble the in the global stiffness matrix. The assembly process is done in the same way as explained in Chapter 3, except that the enriched elements have a much bigger stiffness matrix and

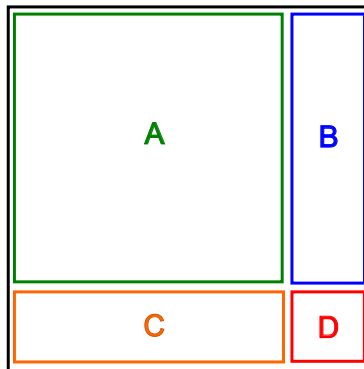


Figure 5.1: Stiffness matrix configuration.

much more degrees of freedom. Any node has two additional degrees of freedom (x and y) for each of its enriched function. This means that, for an element:

- Any node enriched with the Heaviside function has two additional degrees of freedom;
- Any node enriched with the asymptotic function has eight additional degrees of freedom (on account of the four crack tip enrichment functions).

In order to do the assembly, there is the need to first get the vector with the degrees of freedom (\mathbf{df}).

The following code lines represent all the assembly process in the developed XFEM code:

```
for a=1:length(df)
    ka=df(1,a);
    for b=1:length(df)
        kb=df(1,b);
        K(ka,kb)=K(ka,kb)+Ke(a,b);
    end
end
```

where $K(ka,kb)$ is an element of the global stiffness matrix and $Ke(a,b)$ is an element of the elementary stiffness matrix.

5.5 Numerical integration of the enriched elements

As already referred in Subsection 4.4.2, Gauss quadrature can only be used as an integration process of continuous functions. However, the XFEM crack enrichment is performed with the addition of discontinuous approximation functions, meaning that, in the enriched elements, the standard Gauss quadrature can't be directly used.

A simple solution is the creation of triangular subdomains and proceed to their integration process. If the triangles are created around the crack, they don't incorporate the discontinuity of the enrichment function, meaning that each subdomain is continuous and that the Gauss quadrature can be used to integrate them. The contribution of each triangle of the subdomain to the element stiffness matrix corresponds to its area, meaning that

$$\mathbf{K}^e = \sum_{i=1}^{N_t} (A_i \mathbf{K}_i \det \mathbf{J})^e, \quad (5.19)$$

where N_t is the number of triangles, A_i is a triangle area, \mathbf{K}_i is a triangle stiffness matrix and, since the isoparametric coordinate system is used, $\det \mathbf{J}$ is the determinant of the element Jacobian matrix.

Since the subdivision is only made for the integration process, it doesn't add any degrees of freedom to the problem.

5.5.1 Subdomain

In Octave, the triangles belonging to the subdomain are generated by the `delaunay` command. This command uses the nodes and the intersection points of the crack with the edges of the element to generate the triangular mesh between them. It works with the isoparametric coordinates of the points and gives back a matrix with three columns and a number of lines equal to the number of generated triangles. Each line represents a triangle and has the number of points belonging to the triangle. To construct the triangles, these points numbers need to be converted into their isoparametric coordinates.

The real trick here is that the triangular elements are integrated with the shape functions and natural coordinates of the isoparametric quadrilateral. Such means that the coordinates of a Gauss point need to be converted from the isoparametric triangle into the isoparametric quadrilateral element. These coordinates are used to get the value of the shape functions in the integration points. This conversion is made with

$$[\xi_{\text{quad}} \quad \eta_{\text{quad}}] = \mathbf{n}\mathbf{C}^{\text{tri}}, \quad (5.20)$$

where ξ_{quad} and η_{quad} are the local coordinates of the Gauss point inside the isoparametric quadrilateral. This is done with

$$\mathbf{n} = [1 - \xi_{\text{gp}} - \eta_{\text{gp}} \quad \xi_{\text{gp}} \quad \eta_{\text{gp}}], \quad (5.21)$$

where \mathbf{n} is the vector with the three shape functions of the triangle. ξ_{gp} and η_{gp} are coordinates of the Gauss points. These coordinates are expressed on the isoparametric triangle referential by

$$\mathbf{C}^{\text{tri}} = \begin{bmatrix} \xi_{1\text{tri}} & \eta_{1\text{tri}} \\ \xi_{2\text{tri}} & \eta_{2\text{tri}} \\ \xi_{3\text{tri}} & \eta_{3\text{tri}} \end{bmatrix}, \quad (5.22)$$

where \mathbf{C}^{tri} is the matrix with the coordinates of the nodes of the triangle in the quadrilateral isoparametric referential.

Figure 5.2 shows an example of a division of the subelements into triangles. The domain generated by these triangles is affected only by continuous functions, so it can be integrated.

5.5.2 Gauss points

In non-enriched elements, only four Gauss points are needed in order to get an accurate resolution. However, since the enriched elements have functions with a much higher gradient, by using only four Gauss points, the results won't be very accurate. Knowing this, and after some trial and error attempts, it was found the lowest number of Gauss points in order to get results convergence.

Enriched elements

In the enriched elements, it is necessary to use 9 Gauss points for each triangle, of the subdomain, inside the quadrilateral element. This means that the number of Gauss points inside an enriched element varies from 36 to 54 points, depending on the crack geometry.

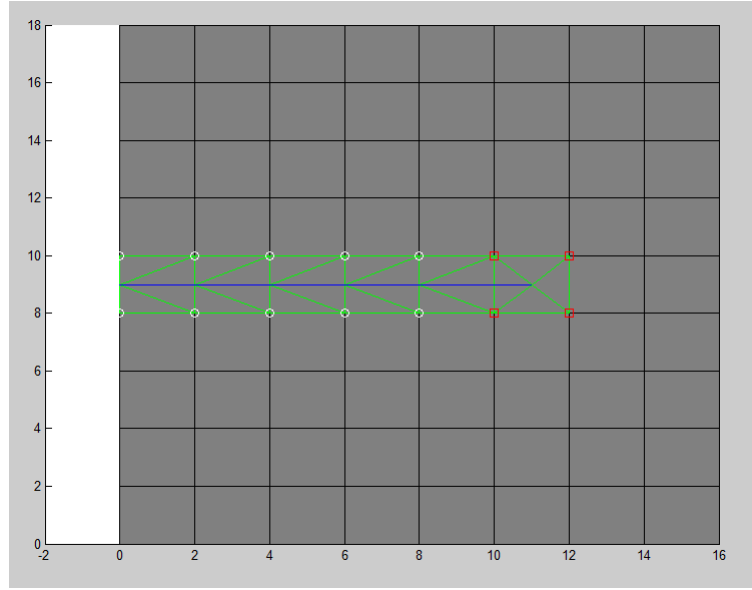


Figure 5.2: Division of the quadrilateral enriched elements into triangles.

Bending elements

In the bending elements, it is necessary to use 36 Gauss points for a square element. Since these elements aren't divided into triangles, they are integrated in the same way as any unenriched element. More points are used only in order to get better results because of the high gradient related to the crack tip enrichment functions. With less points, like 4 or 9, the results weren't as accurate as with 36.

5.6 Calculation of SIF

As already seen in Chapter 2, the stress intensity factor (SIF) is given by

$$K_I = \frac{E^*}{2} M^I \quad (5.23)$$

and

$$K_{II} = \frac{E^*}{2} M^{II}, \quad (5.24)$$

where E^* is related to the Young's modulus and the state of the problem, and M is the interaction integral for the mode in study.

5.6.1 Domain of the interaction integral

In order to compute the interaction integral, a domain must be selected. That domain is circular, around the crack tip, and has a radius r_d that is given by [56]

$$r_d = R\sqrt{A_e}, \quad (5.25)$$

where R is a constant and A_e is the area of the crack tip element. The R value usually is between 2 and 3 [57].

A weight function (q) is then defined and its value is 1 inside the domain and 0 outside. All the nodes are evaluated with this q function. The nodes which are inside the integral domain have $q=1$ and the ones outside the domain have $q=0$. The interaction integral is only evaluated when $\frac{\partial q}{\partial j} \neq 0$, meaning that only elements around the contour contribute for the SIF calculation, since they are the only ones that have nodes with $q=1$ and nodes with $q=0$.

However, when the crack tip is close to a border, the domain of the integral doesn't lie completely inside the body, meaning that the domain becomes open. This integral doesn't represent the crack tip energy with a good accuracy and turns into wrong SIF values. The integral path must be closed, creating a modified domain, in order to solve this problem.

Such task requires an extra step between the creation of the q function and the evaluation for each element. This extra step is very simple, but needs to be done after the assignment of the value 1 to all the nodes inside the domain. After that, it just gives the value 0 for the q function of all the nodes that are on the edges of the plates. This automatically creates a border around the plate, closing the domain.

5.6.2 Interaction integral in FEM

Using the Green's theorem in the interaction integral of equation 2.12, one gets the interaction integral for an area domain. This is a very important step, since, in FEM, it is much easier to compute an area integral than a line integral. The following equation shows the way the interaction integral can be determined with the domain of the previous subsection [58]:

$$M = \sum_{e=1}^{\text{ned}} \sum_{p=1}^{\text{ngp}} \sum_{i=1}^2 \sum_{j=1}^2 \left[\left(\sigma_{ij}^{(\text{Rs})} \frac{\partial u_{i,1}^{(\text{As})}}{\partial x} + \sigma_{ij}^{(\text{As})} \frac{\partial u_{i,1}^{(\text{Rs})}}{\partial x} - W^{(1,2)} \delta_{ij} \right) \frac{\partial q}{\partial j} \det \left(\frac{\partial x_k}{\partial \xi_k} \right) \right] (w_p), \quad (5.26)$$

where e is an element inside the integration domain, ned is the number of elements inside the integration domain, gp is the Gauss point in analysis, ngp is the number of Gauss points, i and j can be 1 or 2 (which means x or y on the crack tip coordinate system), Rs is the real state, As is the auxiliary state, q is the weight function, w_p is the weight of each Gauss point and $\det \left(\frac{\partial x_k}{\partial \xi_k} \right)$ is the determinant of the Jacobian matrix. $W^{(1,2)} \delta_{ij}$ is the strain energy and is given by

$$W^{(1,2)} \delta_{ij} = \sigma_{ij}^{(\text{Rs})} \varepsilon_{ij}^{(\text{As})}, \quad (5.27)$$

where $\sigma_{ij}^{(\text{Rs})}$ is the stress in the real state and $\varepsilon_{ij}^{(\text{As})}$ is the strain in the auxiliary state.

For a rectangular element with four Gauss points, the Gauss weight (w_p), in each point, is equal to 1. The partial derivative of the weight function q is calculated with the x and y partial derivatives of the shape function N_i , using

$$\frac{\partial q}{\partial a} = \sum_{i=1}^4 N_{i,a} q_i, \quad (5.28)$$

where q_i has the value of 0 or 1 and a represents the x or y partial derivative of the weight (q) and shape (N) functions.

5.6.3 Parameters for real state

The $\sigma_{ij}^{(R_s)}$, $\varepsilon_{ij}^{(R_s)}$ and $u_{i,1}^{(R_s)}$ parameters are obtained by XFEM analyses. However, they need to be converted into the crack tip coordinate system. In order to do it, one needs to use the following rotation matrix:

$$\mathbf{R} = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}, \quad (5.29)$$

where α is the angle between the crack tip vector and the x axis. The partial derivatives of the weight function q are converted to the crack tip coordinate system ($\frac{\partial q}{\partial 1}$ and $\frac{\partial q}{\partial 2}$) by

$$\begin{bmatrix} \frac{\partial q}{\partial 1} \\ \frac{\partial q}{\partial 2} \end{bmatrix} = \mathbf{R} \begin{bmatrix} \frac{\partial q}{\partial x} \\ \frac{\partial q}{\partial y} \end{bmatrix}. \quad (5.30)$$

Stresses

The stresses tensor is converted into the crack tip coordinate system by

$$\boldsymbol{\sigma}^{(R_s)} = \mathbf{R}\boldsymbol{\sigma}\mathbf{R}^T, \quad (5.31)$$

where $\boldsymbol{\sigma}$ is the stress tensor in the Cartesian coordinate system and $\boldsymbol{\sigma}^{(R_s)}$ is the stress tensor in the crack tip coordinate system.

Displacement gradient

The displacement gradient is converted into the crack tip coordinate system by

$$\mathbf{u}_G^{(R_s)} = \mathbf{R}\mathbf{u}_G\mathbf{R}^T, \quad (5.32)$$

where

$$\mathbf{u}_G = \begin{bmatrix} u_{x,x} & u_{x,y} \\ u_{y,x} & u_{y,y} \end{bmatrix} \quad (5.33)$$

is the displacement gradient in the Cartesian coordinate system and

$$\mathbf{u}_G^{(R_s)} = \begin{bmatrix} u_{x,x}^{(R_s)} & u_{x,y}^{(R_s)} \\ u_{y,x}^{(R_s)} & u_{y,y}^{(R_s)} \end{bmatrix} \quad (5.34)$$

is the displacement gradient in the crack tip coordinate system.

Strains

The strain tensor is converted into the crack tip coordinate system by

$$\boldsymbol{\varepsilon}^{(R_s)} = \frac{1}{2} \begin{bmatrix} 2u_{x,x}^{(R_s)} & u_{x,y}^{(R_s)} + u_{y,x}^{(R_s)} \\ u_{y,x}^{(R_s)} + u_{x,y}^{(R_s)} & 2u_{y,y}^{(R_s)} \end{bmatrix}, \quad (5.35)$$

where $\boldsymbol{\varepsilon}^{(R_s)}$ is the strain tensor in the crack tip coordinate system.

5.6.4 Parameters of the auxiliary state for mode I

In order to obtain the K_I value, it is necessary to get the values for all the auxiliary stresses, strains and displacement gradients. These parameters don't need to be converted to the crack tip coordinate system because they all are functions of θ .

Stress

The auxiliary stress values for mode I are calculated by

$$\begin{aligned}\sigma_{xx}^{(A_s)} &= K_I \frac{1}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right) \left[1 - \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\theta}{2}\right)\right], \\ \sigma_{yy}^{(A_s)} &= K_I \frac{1}{\sqrt{2\pi r}} \sin\left(\frac{\theta}{2}\right) \left[1 + \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right)\right] \quad \text{and} \\ \sigma_{xy}^{(A_s)} &= K_I \frac{1}{\sqrt{2\pi r}} \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right),\end{aligned}\tag{5.36}$$

where r is the radius and θ the angle between the r vector and the crack tip vector.

Displacement gradient

In order to know the displacement gradient, it is necessary to first get the displacement derivatives for r and θ . For the auxiliary state, they are obtained using the following equations:

$$\begin{aligned}u_{x,r}^{(A_s)} &= K_I \frac{1}{4G\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right) [k - \cos(\theta)], \\ u_{x,\theta}^{(A_s)} &= K_I \frac{1}{2G} \sqrt{\frac{r}{2\pi}} \left[-\frac{1}{2} \sin\left(\frac{\theta}{2}\right) [k - \cos(\theta)] + \cos\left(\frac{\theta}{2}\right) \sin(\theta)\right], \\ u_{y,r}^{(A_s)} &= K_I \frac{1}{4G\sqrt{2\pi r}} \sin\left(\frac{\theta}{2}\right) [k - \cos(\theta)] \quad \text{and} \\ u_{y,\theta}^{(A_s)} &= K_I \frac{1}{2G} \sqrt{\frac{r}{2\pi}} \left[\frac{1}{2} \cos\left(\frac{\theta}{2}\right) [k - \cos(\theta)] + \sin\left(\frac{\theta}{2}\right) \sin(\theta)\right],\end{aligned}\tag{5.37}$$

where G is the shear modulus, r is the radius, θ is the angle and k is the Kolsov constant.

The gradient of displacements on the local Cartesian crack tip coordinate system corresponds to

$$u_{x,x}^{(A_s)} = u_{x,r}^{(A_s)} \frac{\partial r}{\partial x} + u_{x,\theta}^{(A_s)} \frac{\partial \theta}{\partial x},\tag{5.38}$$

$$u_{y,y}^{(A_s)} = u_{y,r}^{(A_s)} \frac{\partial r}{\partial y} + u_{y,\theta}^{(A_s)} \frac{\partial \theta}{\partial y},\tag{5.39}$$

$$u_{x,y}^{(A_s)} = u_{x,r}^{(A_s)} \frac{\partial r}{\partial y} + u_{x,\theta}^{(A_s)} \frac{\partial \theta}{\partial y}\tag{5.40}$$

and

$$u_{y,x}^{(A_s)} = u_{y,r}^{(A_s)} \frac{\partial r}{\partial x} + u_{y,\theta}^{(A_s)} \frac{\partial \theta}{\partial x},\tag{5.41}$$

where $\frac{\partial r}{\partial x}$, $\frac{\partial r}{\partial y}$, $\frac{\partial \theta}{\partial y}$ and $\frac{\partial \theta}{\partial x}$ are obtained by

$$\frac{\partial r}{\partial x} = \cos(\theta) \quad ; \quad \frac{\partial \theta}{\partial x} = \frac{-\sin(\theta)}{r}; \quad (5.42)$$

$$\frac{\partial r}{\partial y} = \sin(\theta) \quad ; \quad \frac{\partial \theta}{\partial y} = \frac{-\cos(\theta)}{r}. \quad (5.43)$$

Strain

Strain values for the mode I auxiliary state can be obtained by

$$\epsilon^{(A_s)} = \frac{1}{2} \begin{bmatrix} 2u_{x,x}^{(A_s)} & u_{x,y}^{(A_s)} + u_{y,x}^{(A_s)} \\ u_{y,x}^{(A_s)} + u_{x,y}^{(A_s)} & 2u_{y,y}^{(A_s)} \end{bmatrix}, \quad (5.44)$$

where the parameters $u_{x,x}^{(A_s)}$, $u_{y,y}^{(A_s)}$, $u_{x,y}^{(A_s)}$ and $u_{y,x}^{(A_s)}$ can be obtained from the previous equations.

5.6.5 Parameters of the auxiliary state for mode II

Stress

The auxiliary stress values for mode II can be calculated by

$$\begin{aligned} \sigma_{xx}^{(A_s)} &= -K_{II} \frac{1}{\sqrt{2\pi r}} \sin\left(\frac{\theta}{2}\right) \left[2 + \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right) \right], \\ \sigma_{yy}^{(A_s)} &= K_{II} \frac{1}{\sqrt{2\pi r}} \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{3\theta}{2}\right) \end{aligned} \quad (5.45)$$

and

$$\sigma_{xy}^{(A_s)} = K_{II} \frac{1}{\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right) \left[1 - \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\theta}{2}\right) \right],$$

where r is the radius, θ the angle between the r vector and the crack tip vector.

Displacement gradient

In order to know the displacement gradient, it is necessary to first get the displacement derivatives for r and θ . For the auxiliary state, they can be obtained by

$$\begin{aligned} u_{x,r}^{(A_s)} &= K_{II} \frac{1}{4G\sqrt{2\pi r}} \sin\left(\frac{\theta}{2}\right) [k + 2 + \cos(\theta)], \\ u_{x,\theta}^{(A_s)} &= K_{II} \frac{1}{2G} \sqrt{\frac{r}{2\pi}} \left[\frac{1}{2} \cos\left(\frac{\theta}{2}\right) [k + 2 + \cos(\theta)] - \sin\left(\frac{\theta}{2}\right) \sin(\theta) \right], \\ u_{y,r}^{(A_s)} &= -K_{II} \frac{1}{4G\sqrt{2\pi r}} \cos\left(\frac{\theta}{2}\right) [k - 2 + \cos(\theta)] \quad \text{and} \\ u_{y,\theta}^{(A_s)} &= -K_{II} \frac{1}{2G} \sqrt{\frac{r}{2\pi}} \left[-\frac{1}{2} \sin\left(\frac{\theta}{2}\right) [k - 2 + \cos(\theta)] - \cos\left(\frac{\theta}{2}\right) \sin(\theta) \right], \end{aligned} \quad (5.46)$$

where G is the shear modulus, r is the radius, θ is the angle between the r vector and the crack tip vector and k is the Kolsov constant. The gradient of displacements on the local Cartesian crack tip coordinate system corresponds to

$$u_{x,x}^{(A_s)} = u_{x,r}^{(A_s)} \frac{\partial r}{\partial x} + u_{x,\theta}^{(A_s)} \frac{\partial \theta}{\partial x}, \quad (5.47)$$

$$u_{y,y}^{(A_s)} = u_{y,r}^{(A_s)} \frac{\partial r}{\partial y} + u_{y,\theta}^{(A_s)} \frac{\partial \theta}{\partial y}, \quad (5.48)$$

$$u_{x,y}^{(A_s)} = u_{x,r}^{(A_s)} \frac{\partial r}{\partial y} + u_{x,\theta}^{(A_s)} \frac{\partial \theta}{\partial y} \quad (5.49)$$

and

$$u_{y,x}^{(A_s)} = u_{y,r}^{(A_s)} \frac{\partial r}{\partial x} + u_{y,\theta}^{(A_s)} \frac{\partial \theta}{\partial x}, \quad (5.50)$$

where $\frac{\partial r}{\partial x}$, $\frac{\partial r}{\partial y}$, $\frac{\partial \theta}{\partial x}$ and $\frac{\partial \theta}{\partial y}$ are given by equation 5.42 and 5.43.

Strain

Strain values for the mode II imaginary state can be obtained by

$$\boldsymbol{\epsilon}^{(A_s)} = \frac{1}{2} \begin{bmatrix} 2u_{x,x}^{(A_s)} & u_{x,y}^{(A_s)} + u_{y,x}^{(A_s)} \\ u_{y,x}^{(A_s)} + u_{x,y}^{(A_s)} & 2u_{y,y}^{(A_s)} \end{bmatrix}, \quad (5.51)$$

which is the same equation used for the mode I; however, they differ since the displacement gradients aren't the same.

5.7 Crack propagation

In this work, the maximum release energy rate criterion for crack propagation was used. As already referred in Chapter 2, the crack propagation will start when the maximum release energy rate, obtained by equation 2.20, reaches a critical level. For each iteration, when the critical value is achieved, the crack will propagate a predefined distance in the direction obtained by equation 2.19. Since in XFEM the asymptotic enrichment is done based on the crack tip direction, it is irrelevant for an element to have several crack segments inside. However, if that happens, it results into a large number of iterations, leading to a very high computational cost. This means that it is undesirable to have a propagated distance value that is too small compared with the elements size.

This page was intentionally left blank.

Chapter 6

Numerical Examples

In this chapter some numerical results are presented in order to validate the XFrac-2D software. Firstly, a FEM validation process was made with 2 different tests: a tensile test and a bending test. The displacements, reactions in the nodes with boundary conditions and stress results were compared with the theoretical values. Then, a XFEM validation was done with 5 tests. The first one using a plate with a crack in its center, with an angle between the crack tip vector and the x axis (α) equal to 0° . The second one using a plate with an edge crack, again with $\alpha = 0^\circ$. The third one using a plate with a crack in its center and considering an α angle variation from 0° to 90° . Stress intensity factors were computed for these three tests and compared with theoretical ones. The last two tests correspond to crack growth problems. The first for a center crack and the second for an edge crack, being their main objective to analyse the crack propagation results.

Another thing discussed in this chapter is the crack detection process. In fact, since the implementation of the level-set method didn't produce accurate results under every circumstances, the problematic cases were identified and two specific resolution processes were adopted so that the crack detection process could work correctly.

6.1 FEM validation

6.1.1 Tensile test

The first test is the simulation of a tensile test for an aluminium sheet. Only a quarter of the sheet was modelled because of the existence of symmetry on the x and y axes. A 5×5 element mesh was used and the problem was modelled as a plane strain state. The material is AlMgSi1 alloy, with $E=70$ GPa and $\nu=0.32$. The size of the quarter of the sheet is 5×10 mm² (xy). Figure 6.1 shows the geometry and boundary conditions for this problem, being the applied load of 1750N.

The numerical results of this test were the same as the theoretical ones, being the maximum displacement of 0.05 mm in the y -direction and -0.008 mm in the x -direction. The relation between strain and displacement is given by

$$\varepsilon = \frac{\Delta L}{L_0}, \quad (6.1)$$

where ε is the strain, L_0 is the initial length and ΔL is the difference between the final and initial length. For this test, equation 6.1 gives a strain of 0.005 in the force direction.

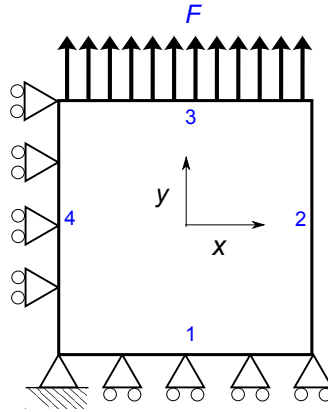


Figure 6.1: Geometry and boundary conditions for the tensile test.

The stress obtained by the FEM program was 350 MPa in the y -direction, which is in accordance with the problem's data.

6.1.2 Bending test

For the bending test, a whole beam is modelled and discretized with three different meshes (20×2 , 40×4 and 100×10). Figure 6.2 shows the geometry and boundary conditions in this test. The material is the same used in the previous test ($E=70$ GPa and $\nu=0.32$). Again, the problem was modelled as a plane strain state. The size of the beam is 80×10 mm² and it is pinned on edge number 4. The load is applied on the edge number 2 and its value is -50 N on the y -direction. The theoretical displacement for this case is 2.857 mm, obtained by [59]

$$d = \frac{F_y L^3}{3EI_z}. \quad (6.2)$$

In this work, the absolute values for the error and the relative error has been adopted. Table 6.1 shows the results and errors for each mesh. As expected, the error in the 20×2 mesh is the highest. This can be explained by the fact of the mesh being the less refined one, which may create some approximation problems. However, for most refined mesh, the results are accurate, with a relative error minor than 1%.

Mesh	20×2	40×4	100×10
Displacement [mm]	2.5518	2.7742	2.8589
Error [mm]	0.3052	0.0828	0.0019
Error [%]	10.68	2.900	0.67

Table 6.1: Displacement and errors for the considered meshes.

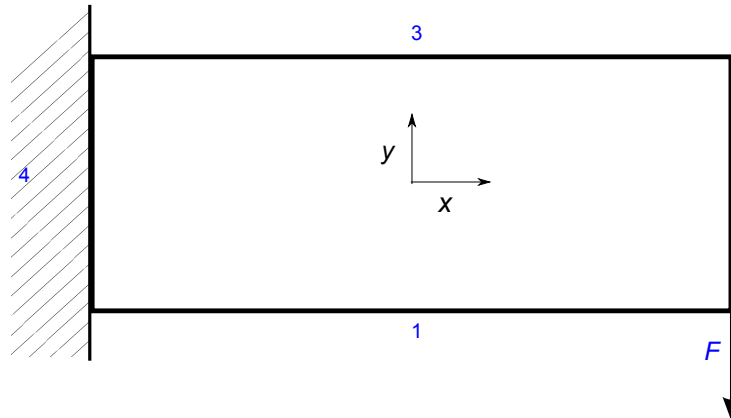


Figure 6.2: Geometry and boundary conditions for the bending test.

6.1.3 FEM remarks

The numerical results were in accordance with the analytical values and therefore, in this context, the FEM program seems to be validated.

6.2 Modification of the detecting process

In the initial versions of the XFrac-2D, the implementation of the level-set criteria didn't work well under some circumstances. Sometimes, the program detected elements that didn't have a crack as crack tip elements. This occurred with certain crack geometries and meshes, mainly when the mesh was unrefined and the crack was oblique. In these cases, an element without the crack tip had positive and negative Φ and Ψ level-set values in its nodes. The same problem occurred when the crack geometry was complex, like when it was composed by multiple vectors. In these cases, the two Ψ level-set functions weren't enough to restrict the crack domain and, in most cases, the crack wasn't properly modelled by only one Φ level-set function. This made XFrac-2D incapable of correctly tracking the crack, since it detected regular non-enriched elements as crack tip or completely cut elements. These problems are shown in Figure 6.3. As explained in

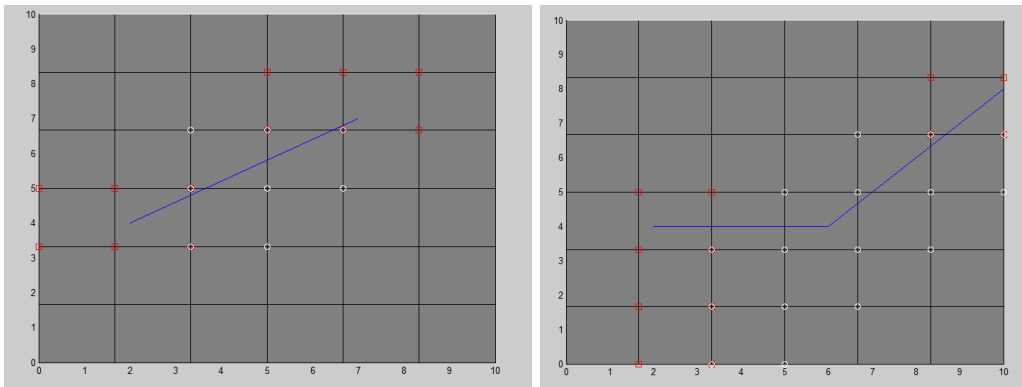


Figure 6.3: Incorrect detection of the enriched elements.

the following subsections, the first problem was solved with the incorporation of a crack tip condition, the second one was solved by splitting the complex crack geometries into simpler vectors and analysing each one individually.

6.2.1 Crack tip condition

In accordance to the reasons presented before, the criteria shown in equation 4.40 weren't enough to identify properly a crack tip element. In order to solve this problem, a position condition was created in order to ensure that the crack tip is inside the element, which means that an element can only be detected as a crack tip element if the crack tip's x and y are between the minimum and maximum x and y of the element.

6.2.2 Cracks with multiple vectors

The criteria presented in Equation 4.39 also weren't accurate in some cases. Such situations occurred when the mesh was unrefined and when the crack was composed by at least 2 vectors. In these cases, some elements around the crack kinks were detected as completely cut elements. This happened because software only used the two crack tips to create the Ψ level-set function, meaning that the Φ level-set detected elements around the kink that weren't crossed by the crack. Figure 6.3 shows the problem.

The adopted procedure was to split of the crack in all its vectors and applying the level-set function to all of them separately. After that, the software ended up with a lot of crack tip elements. However, most of them were completely cut elements and needed to be converted. This was achieved with the following condition: any element detected as a crack tip element for two vectors is really a completely cut element.

6.3 XFrac-2D validation

This subsection contains the numerical results of the stress intensity factor, for different static cases. The numerical results are compared with analytical solutions. For the plate with an edge crack, a study of how the K_I results change with the mesh refinement and the M-integral domain was done. The slanted crack was used to study the evolution of the SIF with the crack angle. After the correct stress intensity factors were identified, some crack growth problems were analysed.

6.3.1 Plate with a central crack under uniaxial tension

The first test was a plate, with a crack in its center, tractioned by a constant remote stress σ , that was perpendicular to the crack direction. The plate had 50 mm height, 20 mm length and $a=4$ mm. The problem geometry and the boundary conditions are shown in Figure 6.4 and Table 6.2 shows all the data of the problem. The problem mesh and enriched elements are show by Figure 6.5. The mesh used had 22×55 elements.

The modified level set function, from Section 5.2, was used to identify the enriched elements. Figure 6.5 shows the enriched nodes and elements. All the crack tip enriched nodes are assigned with red squares and all the Heaviside enriched nodes are assigned with white circles.

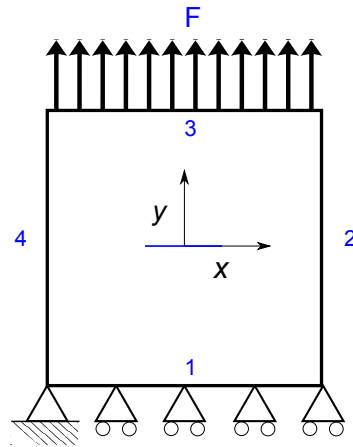


Figure 6.4: Geometry and boundary conditions for the plate with a central crack.

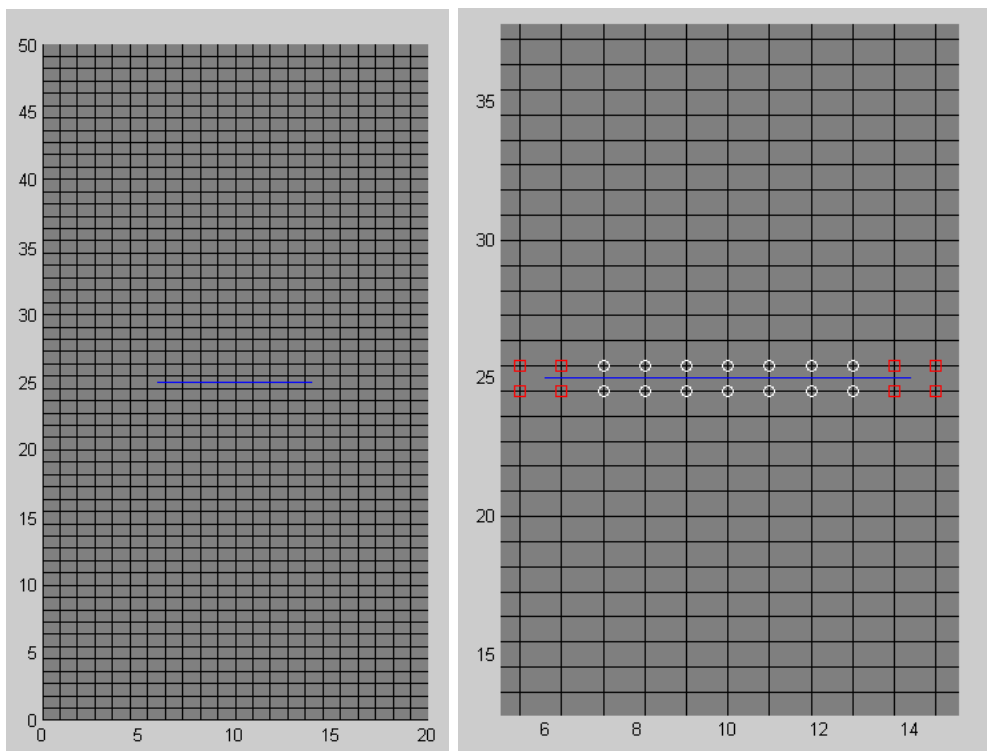


Figure 6.5: Representation of the mesh and the crack in the plate.

The stress intensity factor was evaluated using the XFEM analysis and compared with a theoretical solution. Since this was a pure mode I load, the theoretical value of K_I was given by the following equation [56]

$$K_I = \sigma F(a/l) \sqrt{\pi a}, \quad (6.3)$$

where a is half the crack length (because of the symmetry of the problem), σ is the remote stress and $F(a/L)$ is a correction factor for problems using finite plates.

Since $a/l = 0.4$, $F(a/l) = 1.1094$ and the theoretical $K_I = 393.27 \text{ MPa}\cdot\text{mm}^{1/2}$ [60].

The numerical result was $K_I = 389.63 \text{ MPa}\cdot\text{mm}^{1/2}$. This gave an error of $3.64 \text{ MPa}\cdot\text{mm}^{1/2}$ or 0.93% .

As expected, the remote stress in the y -direction was equal to the problem remote stress and there was a formation of a zone without any stresses, around the completely cut element. These characteristics are represented in Figure 6.6.

6.3.2 Plate with an edge crack

The second test was a plate, with a lateral crack, again under uniaxial stress. The plate had 50 mm height, 10 mm length and the crack size was 4.1 mm. The crack was located in the left side of the plate, in the middle of y -direction. The state and the material were the same as in the previous test and it was plane strain analyses. The force magnitude needed to be half the value from the previous test because it was used the same remote stress but now the plate has only half the length. Figure 6.7 shows the problem geometry and boundary conditions and Table 6.3 shows all the test properties.

For a lateral crack, $F(a/l)$ was determined by [60]

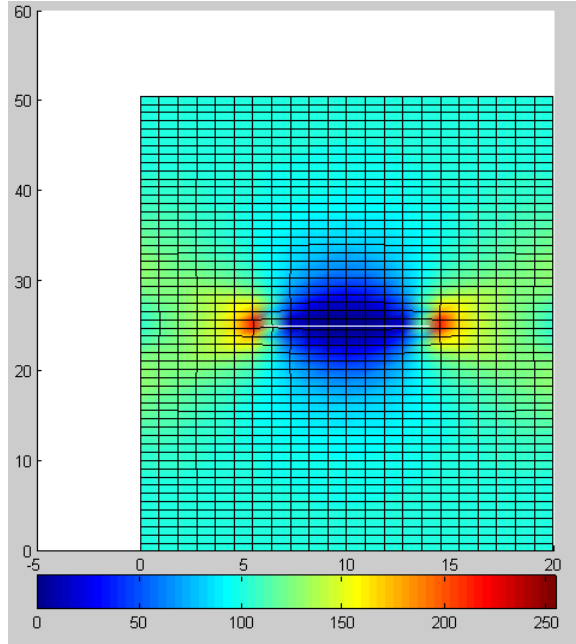
$$F(a/l) = 1.12 - 0.231c + 10.55c^2 - 21.72c^3 + 30.39c^4. \quad (6.4)$$

Since $c = a/l = 0.41$, $F(a/l) = 2.1605$ and theoretical $K_I = 775.40 \text{ MPa}\cdot\text{mm}^{1/2}$. This was the reference value used to calculate the error and accuracy, for all the meshes and M-domain's radiuses.

The first test done used a 9×49 mesh and a M-domain radius $R=3$. The K_I value obtained was $712.4 \text{ MPa}\cdot\text{mm}^{1/2}$, with an error of $63 \text{ mm}^{1/2}$ or 8.12% , which is very high. Remember that the previous problem only had an error of 0.93% . To improve the results,

Property	Value
Crack size, a [mm]	4
Length, L [mm]	20
Height, H [mm]	50
Force, F [N]	2000
Remote stress, σ [MPa]	100
Young's modulus, E [MPa]	10^4
Poisson's ratio, ν [-]	0.3
State	Plane Strain
Mesh	22×55

Table 6.2: Properties of the plate with a central crack.

Figure 6.6: Stress in y -direction [MPa].

a study about the impact of the mesh, the M-domain radius and the kind of stress used in the calculation of K_I was done.

The first thing to do was to understand the impact of the use of the average nodal stress, instead of the elementary stress, in the calculation of the M-integral. For that, two tests were performed. One, using the nodal stress obtained directly from the element, while, the other, using the average stress of all the elements containing that node. Both tests used the same mesh and M-domain radiuses. Each test was performed with two different meshes, so that the impact of the used type of stress in different meshes could be understood. Table 6.4 shows the values of K_I and the respective errors for the two different meshes and for two different types of stress value. It shows that when the element stress was used K_I had lower errors. For the less refined mesh, the use of element stress had a bigger impact in the final result. The type of used stress in M-integral could reduce

Property	Value
Crack size, a [mm]	4.1
Length, L [mm]	10
Height, H [mm]	50
Force, F [N]	1000
Remote stress, σ [MPa]	100
Young's modulus, E [MPa]	10^4
Poisson's ratio, ν [-]	0.3
State	Plane Strain

Table 6.3: Properties of the plate with an edge crack.

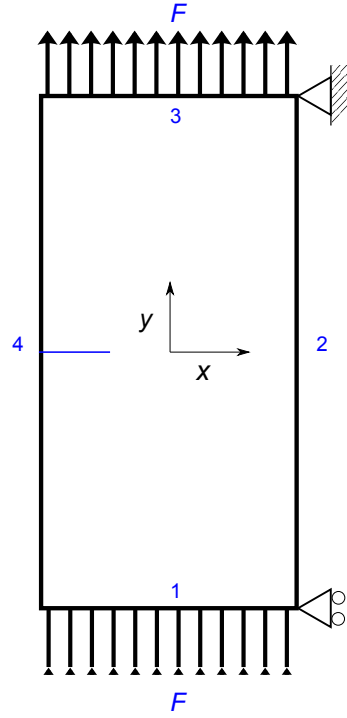


Figure 6.7: Geometry and boundary conditions of the plate with an edge crack.

considerably the error of K_I in rough mesh. However, its effect was almost insignificant for the most refined mesh. This was shown by the difference of only 0.06% for the 19×99 mesh. It seems that for this type of problems it is necessary to use the most refined mesh. In fact, the error was reduced to almost a third of its original value by the mesh refinement.

Mesh	9x49		19x99	
Type of used stress	Element	Average	Element	Average
Stress value [MPa]	741.4	738.4	763.9	763.5
Error [$\text{mm}^{1/2}$]	33.0	37.0	11.5	11.9
Error [%]	4.26	4.77	1.48	1.54

Table 6.4: Influence of the type of stress and mesh refinement for a plate with a lateral crack.

Since the type of used mesh had a high influence on the value of K_I , another analysis was done in order to try to understand the effect of the mesh refinement and the M domain on the sensibility of the XFEM results. So, considering the properties from Table 6.3, a test was performed for 5 different types of mesh (9×49 , 11×55 , 15×65 , 15×75 and 19×99) and 5 different radiuses of the M-integral domain (1.5, 2, 2.5, 3 and 3.5). The results and relative error were calculated and are represented in Table 6.5 and Table 6.6, respectively. Based on the tables, the most consistent results were obtained for R between 2.5 and 3.5. This result was expected since the objective of using the M-integral is to have an accurate evaluation of the crack tip on a more regular field,

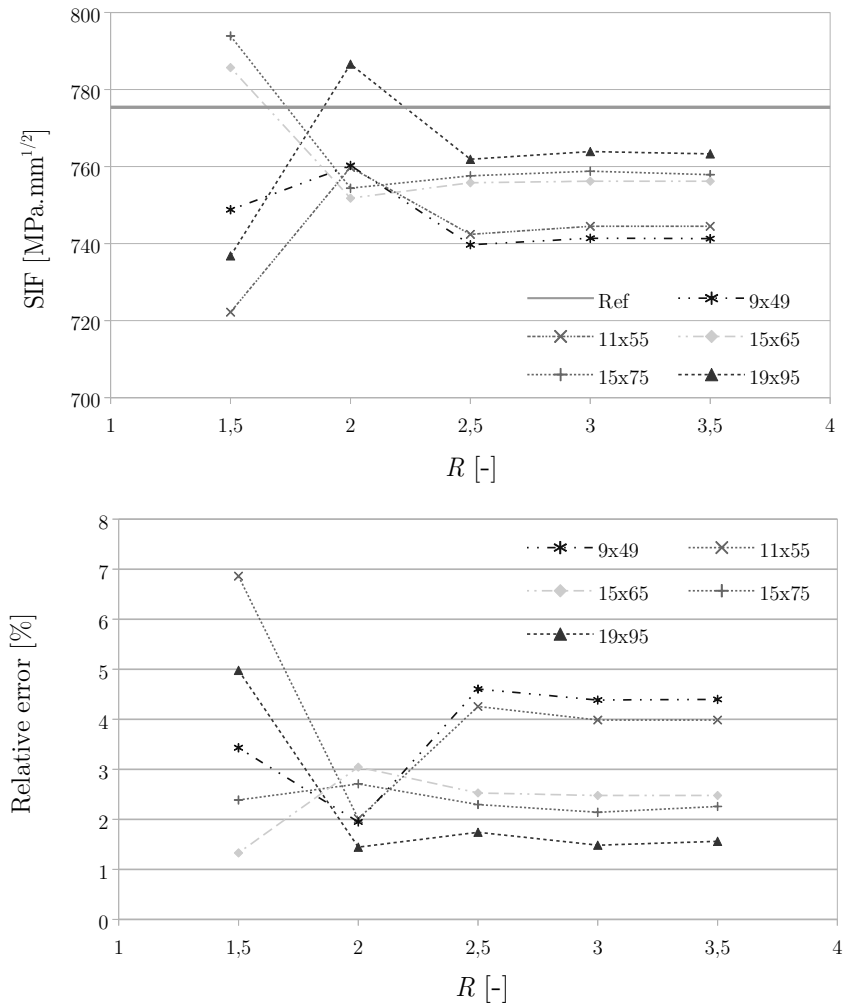


Figure 6.8: SIF and relative error for the mesh refinement analysis.

		Mesh				
		9×49	11×55	15×65	15×75	19×95
R	1.5	748.8	722.2	785.7	793.9	736.8
	2	760.3	759.8	751.8	754.4	786.6
	2.5	739.7	742.4	755.8	757.6	761.9
	3	741.4	744.5	756.2	758.8	763.9
	3.5	741.3	744.5	756.2	757.9	763.3

Table 6.5: K_I [$\text{MPa}\cdot\text{mm}^{1/2}$] for all the different meshes and radiuses.

		Mesh				
		9×49	11×55	15×65	15×75	19×95
R	1.5	3.43	6.86	1.33	2.39	4.98
	2	1.95	2.01	3.04	2.70	1.44
	2.5	4.60	4.26	2.53	2.30	1.74
	3	4.39	3.99	2.48	2.14	1.48
	3.5	4.39	3.99	2.48	2.26	1.56

Table 6.6: Relative error [%] for all the different meshes and radiuses.

which is far away from the crack tip. For all meshes, the K_I value converged with those radiuses. As expected, more refined meshes gave rise to lower errors.

It was impossible to perform a test with a more refined mesh, since the Octave program didn't had enough memory. Another way to improve the results is with the enrichment of more elements, around the crack tip, with the asymptotic function.

6.3.3 Plate with a slanted crack

All the previous tests correspond only to pure mode I problems ($K_{II}=0$). This third test is related to a slanted crack, with length $2a$ and passing through the center of a square plate, as presented in Figure 6.9. This test was used to evaluate the accuracy of the program for a mixed mode fracture problem (I+II) and to analyse the evolution of the two stress intensity factors with the angle of the crack.

The exact stress intensity factors for an infinite plate with a slanted crack are given by [61]

$$K_I = \sigma\sqrt{\pi a} \cos^2(\alpha) \quad (6.5)$$

and

$$K_{II} = \sigma\sqrt{\pi a} \cos(\alpha) \sin(\alpha), \quad (6.6)$$

where a is half the crack length and α is the angle between the crack tip vector and the x axis. The length and with of the plate are 20 mm and the crack length ($2a$) is 8 mm. Since the ratio of the plate and the crack length was only 2.5, it can't be considered as an infinite plate. Such a lower ratio was chosen in order to lower the computational effort by reducing the number of elements in the mesh. The mesh density needed to be high so that the crack had elements enriched with the Heaviside function and that the M-integral's domain doesn't have elements that are affected by the other half of the crack. Since it couldn't be analysed as an infinite plate, a correction factor was added to the theoretical values. Since $a/l=0.4$, the correction factor is $F(a/l) = 1.1094$, which is the same value as that used in Subsection 6.3.1 [60]. The plate is 20 mm high and long, with a mesh of 19×19 elements and the crack length a is 4. The crack is introduced in the center of the plate with and angle α . The material is the same as in the previous tests and a plane strain analysis is adopted. The remote stress is 50 MPa and is applied on edges 1 and 3. The problem geometry and the boundary conditions are shown in Figure 6.9 and Table 6.7 shows all the properties for this test.

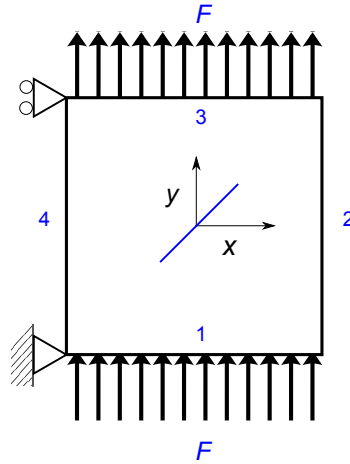


Figure 6.9: Geometry and boundary conditions of the plate with a slanted crack.

Property	Value
Crack size, a [mm]	4
Length, L [mm]	20
Height, H [mm]	20
Force, F [N]	1000
Remote stress, σ [MPa]	50
Young's modulus, E [MPa]	10^4
Poisson's ratio, ν [-]	0.3
State	Plane Strain

Table 6.7: Properties for the plate with a slanted crack.

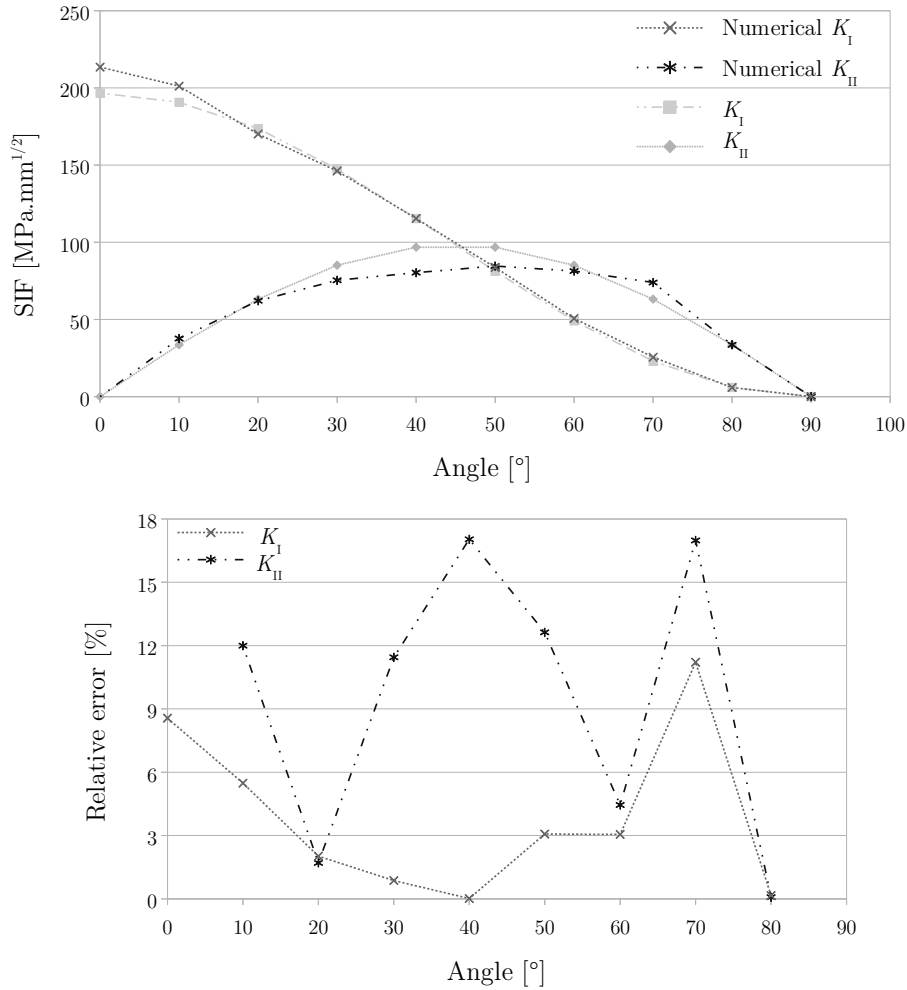
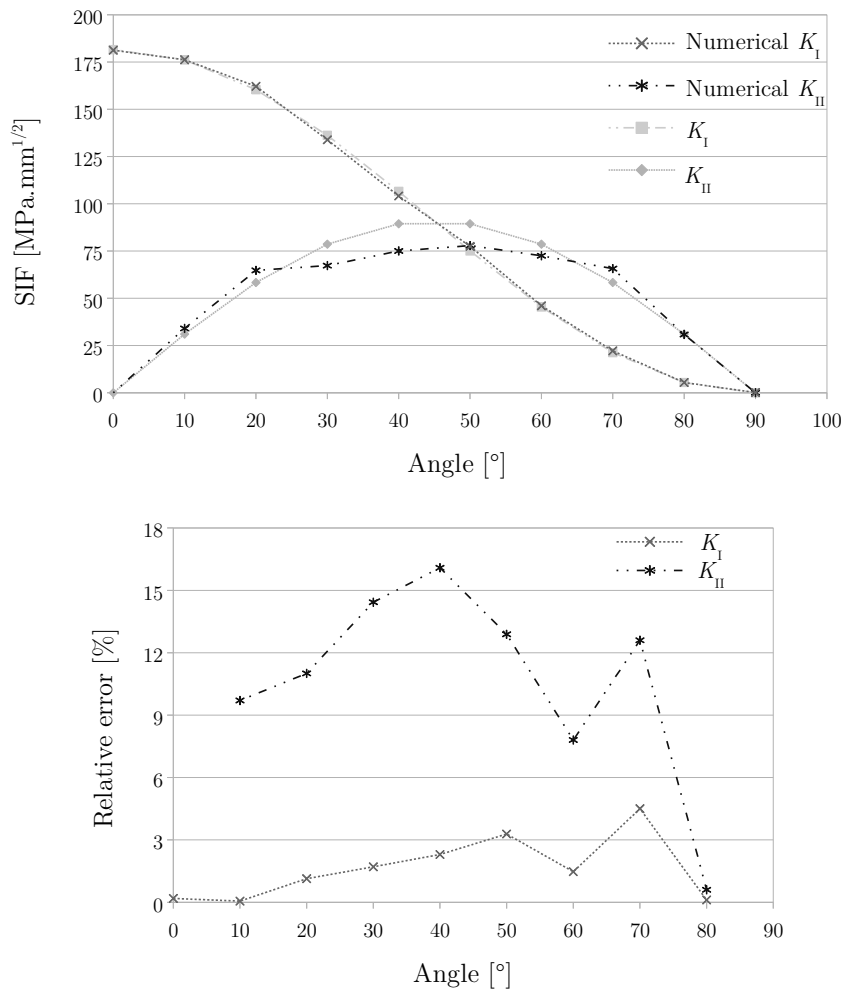


Figure 6.10: SIF and relative error for the $20 \times 20 \text{ mm}^2$ plate.

Figure 6.10 shows the values of SIF and relative error for the test. It can be seen that the errors were very high for both modes. The highest error for K_I was around 11% and, for most of the crack angles, K_{II} has a very high error. Because of these high errors, the test was repeated with a 39×39 mesh, however the results didn't improve. Only the use of bigger plates improved the results. Thus a plate with $40 \times 40 \text{ mm}^2$ was used with the same crack length and the mesh density was kept the same with a 39×39 mesh. With this change, the $F(a/b)$ factor had to be updated to 1.0246 [60]. This time, as can be seen in Figure 6.11, the results were much better for K_I . However, the results of K_{II} didn't improve very much (it continues with errors up to 16%). This kind of errors can be related to an incorrect enrichment of the elements. A solution adopted by some authors is the enrichment of some elements around the crack tip element with the asymptotic function. Another method consists to not enrich the nodes in the completely cut elements if they are far away from the crack [7; 12; 58].

Figure 6.11: SIF and relative error for the $40 \times 40 \text{ mm}^2$ plate.

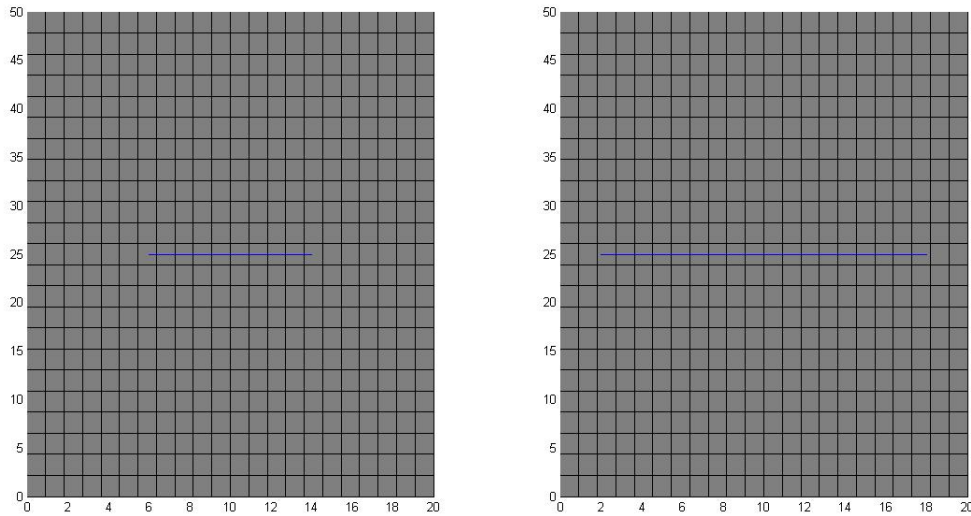


Figure 6.12: Initial and final crack path after the crack growth analyses.

6.3.4 Crack propagation

The main objective of the previous subsection was the evaluation of stress intensity factors for a static crack. Those stress intensity factors are very important to know when the crack starts to grow and its correct path. The following tests are made in order to see if the XFEM program can predict correctly the crack propagation path and the evolution of the mode I stress intensity factor.

Crack growth in a plate with a central crack

In this case, a plate with a crack in its center and being tractioned by a remote stress σ is modelled. All the problem data correspond to those considered in the previous analysis presented in Subsection 6.3.1 and the tests are performed with 4 crack growth iterations. Each iteration has the following properties: mesh with 22 by 55 elements, the M-integral radius (R) is 3 and the crack increment is 1 mm per iteration.

As already referred in Subsection 6.3.1, the problem corresponds to a pure mode I load. Due to symmetry the crack propagation angle (β) must be 0° or 180° with the x axis (each crack tip propagates in opposite directions). As expected, the crack growth followed the predicted path. Figure 6.12 shows the original crack and the propagated crack after four iterations.

Crack growth in a plate with edge crack

This test is performed in order to analyse the evolution of the stress intensity factor with the increasing of the crack length, in a pure mode I load. All the problem data came from the previous 6.3.2 analysis. The mesh corresponds to 19 by 95 elements, the M-integral radius (R) is 3 and the crack increment is 0.5 mm per iteration. This time, the test is performed by six iterations. Since this is again a pure mode I load, a crack growth with

$\beta=0^\circ$ is expected, which is similar to that of the previous test.

Figure 6.13 shows the crack path evolution during the six iterations process. The results were the expected since the crack path goes straight to the end of the body. The first picture illustrates the original crack, the second one illustrates the crack after four iterations and the third one illustrates the final crack after six growth iterations. Figure 6.14 shows the evolution of the mode I SIF and the relative error of the numerical K_I values. It can be seen that the relative error of the numerical K_I increases with the crack dimension.

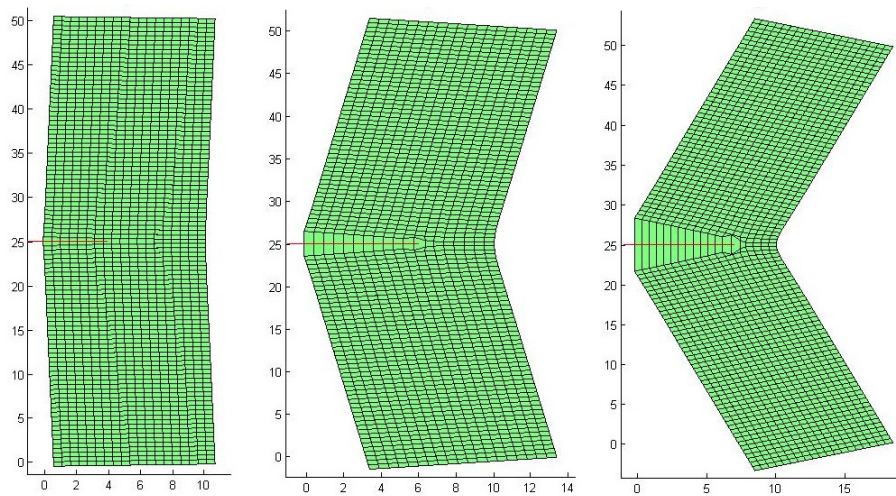


Figure 6.13: Initial, after 4 iterations and final crack path after the crack growth analysis.

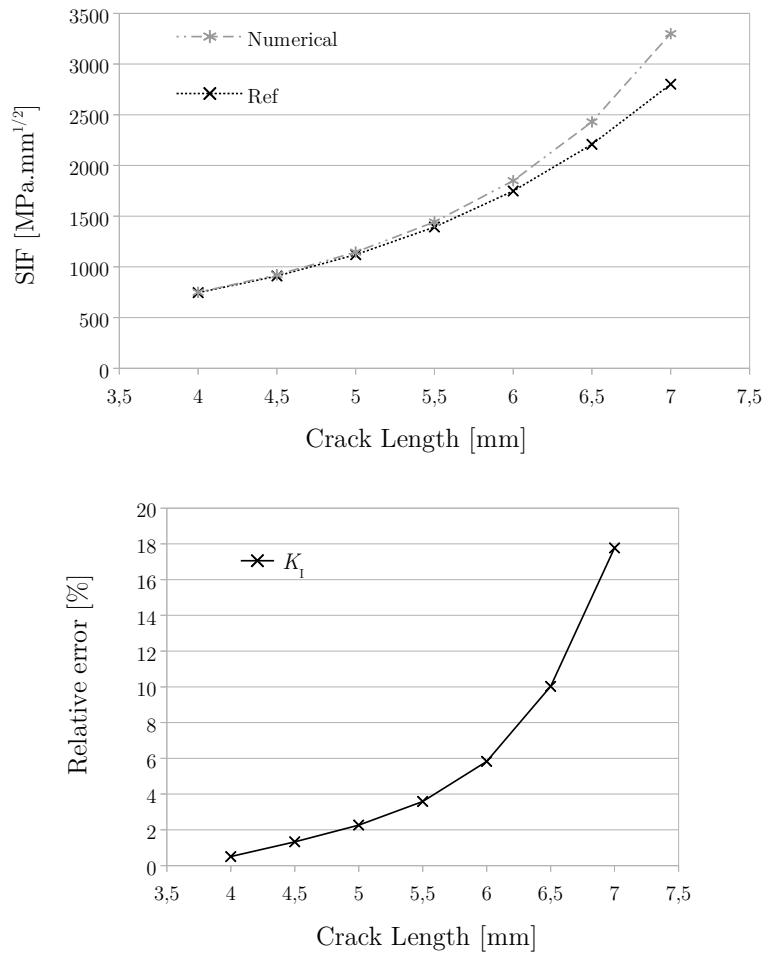


Figure 6.14: SIF and error of K_I for each iteration during the crack propagation.

Chapter 7

Final Remarks

The eXtended Finite Element Method (XFEM) is a partition of unity method that is able to model strong and weak discontinuities using local enrichment. It is a Finite Element Method (FEM) generalization that enables the incorporation of local enrichment of approximation spaces. This allows the domain to be modelled without some general issues that come with the FEM. It is important to mention that any function that describes the general behaviour of the solution can be used as an enrichment function. When there is no discontinuity on the domain, the XFEM software works exactly like a FEM software. The XFEM acts like a subprogram of the standard FEM program.

Unfortunately, since it is a recent method, it is very hard to find information about its implementation process and how an XFEM program works. That creates serious issues to anyone who tries to develop his/her own XFEM program. This work tries to fill a part of this lack of information with the explanation of how the implementation process is done, fulfilled by presenting the program flowchart and the program manual. The implementation part, in this dissertation, aims to explain almost everything the user needs to know to implement the base method with success. It shows little details, like the partial derivatives of the enrichment functions, how the stiffness matrices are stored, the subdomain coordinates change in the integration process or how to calculate the M-integral values. The program flowchart and program manual are also important to understand how an XFEM program works, where the XFEM subroutines are located inside the program, when the subroutines are executed and in which order this occurs.

The XFEM program was implemented and incorporated successfully in a standard FEM program created by the author at the beginning of this work. The program was developed using Octave. The use of the Octave program was one of the main objectives of this work, since it is an open-source software, meaning that any researcher can use it freely.

In this work, the implementation of the level-set functions weren't enough to detect the elements that needed to be enriched. Those cases were studied and the solutions were the crack tip condition and the analyses of complex cracks as multiple vectors. The crack tip condition ensured that an element can only be detected as a crack tip element if the crack tip is inside itself. For a complex crack, its path is divided into vectors and only after that the level-set method is applied. Each vector is analysed individually and then the final solution is created with the detected elements.

The enriched elements needed much more integration points than the non-enriched ones. In the beginning, the enriched elements were integrated with only 4 Gauss points

and the results weren't accurate enough. This occurred because the enrichment function introduced high gradients of stress and displacement fields in the element. The results just started to converge when 6 Gauss points for the triangular elements and 16 Gauss points for the quadrilateral blending elements were adopted. It was decided to use 9 Gauss points for the quadrilateral elements and 36 Gauss points for the quadrilateral blending elements.

The numerical results were very accurate for the pure mode I loads, with relative errors lower than 1%. In these cases, the numerical values of K_I and K_{II} were very accurate, even without using refined meshes. A study was performed for a plate with a lateral crack, showing that results converged with mesh refinement for values of the M-integral radius (R) between 2.5 and 3.5. In the mixed (I+II) mode test, the K_I results were good, since most of the relative errors were below 2%. However, the K_{II} results weren't satisfactory, with most of the errors above 10%. The crack path in the propagation problems was the expected one for a pure mode I load, with a propagation angle (β)=0°. During the propagation process, the mesh was always the same and never needed to be changed, contrarily to what occurs with adaptive meshing in the FEM, being this one of the main characteristics of this method. The errors related to the values of K_I were very low for the first iterations, but they increased with the crack growth. This appears to be the same problem that occurred in the 20×20 mm slanted crack test: the crack started to have a considerable length in relation to the problem domain, the displacements in the FEM analysis became very high and the enrichment ceased to be accurate.

In summary, all the objectives, proposed in the beginning of this work, were achieved. The numerical results shown that the XFrac-2D was developed for Octave with success and all the documentation was created. The software was developed purposely with lots of subroutines to kept them simple and easy to understand by any user. To kept the simplicity, the author decided to not use FEM and XFEM procedure in the same subroutine. This decision was taken since the objective was the development of a pedagogical software instead of an optimized one. The program manual was created and explains how the XFrac-2D works: how to insert the data, where the data is stored inside the software, what each subroutine does and which ones belong to the pre-processing, processing and post-processing step of the software; has a list of all the subroutines with a little description; has some examples of modelled cases; in the end, it has the program flowchart. This manual is very important since it may save lots of time to a new user that is learning how to work with the XFrac-2D.

7.1 Future work

The XFEM program, developed in this work, was created with the objective of being a pedagogical tool for people trying to learn about FEM and XFEM. For this reason, the program has some subroutines that are repeated and it is far away from being optimised. An optimisation process would improve its processing speed and the amount of memory used by the system. Both of these kinds of optimisation are very important and would enable the use of more refined meshes.

The enrichment process must be improved to decrease the errors in some of the mixed (I+II) mode cases. Some authors opt to enrich all the elements around the crack

tip element, creating more degrees of freedom, but improving the convergence of the displacement field. Others opt, when the crack tip becomes close to an edge, to just enrich the next element. Another variant of the enrichment could be applied to the completely cut elements. In this variant, one doesn't enrich a node inside a completely cut element if the crack is really close to it. All the other nodes of the element that don't meet this condition, are enriched with the Heaviside function. In these cases, the enrichment of the further nodes is enough and should produce a more accurate solution.

Another step that can be performed in this program is to make it capable of modelling other discontinuities, such as voids, inclusions and multiple materials. The process for modelling these discontinuities is much similar to modelling cracks. However, each discontinuity needs its own detecting process and has its own specific enrichment functions. Adding the methodology and the proper enrichment functions to these discontinuities would open the program to new research fields and make it much more general.

Another way to generalize the program is the incorporation of new boundary conditions, such as symmetry or the choice of which elements should be restricted, and the incorporation of new types of loads, such as concentrated or distributed forces in only a part of an edge. All these changes are done exactly in the same way as they are done for a FEM program.

General XFEM programs work in three dimensions. It would be a really big improvement evolving from two to three-dimensional simulations. In this case, the level-set method needs to be updated for a three-dimensional application, by the consideration of two extra tangential level-set functions, and all the enrichment functions must be updated to the three-dimensional coordinate system.

This page was intentionally left blank.

Bibliography

- [1] Mohammadi, S.; *Extended Finite Element Method for Fracture Analysis of Structures*, Blackwell Publishing Ltd., Tehran, Iran, 2008.
- [2] Zhuang, Z., Liu, Z., Cheng, B., Liao, J.; *Extended Finite Element Method*, Academic Press, Tsinghua, China, 2014.
- [3] Pommier, S., Gravouil, A., Moës, N., Combescure, A.; *Extended Finite Element Method for Crack Propagation*, ISTE, London, United Kingdom, 2011.
- [4] Griffith, A.; The Phenomena of Rupture and Flow in Solids, *Transactions of the Royal Society of London*, Series A, London, United Kingdom, 1921.
- [5] Costa, A.; *Método das Forças para a Determinação Numérica dos Factores de Intensidade de Tensão* (in Portuguese), Master dissertation in Mechanical Engineering, Department of Mechanical Engineering, University of Aveiro, Aveiro, Portugal, 2008.
- [6] Orowan, E.; Energy Criteria of Fracture, *Welding Journal Research Supplement* **20**, 1955.
- [7] Ahmed, A.; *Extended Finite Element Method (XFEM) – Modeling Arbitrary Discontinuities and Failure Analysis*, Master dissertation in Earthquake Engineering, Istituto Universitario di Studi Superiori di Pavia, University of Pavia, Pavia, Italy, 2009.
- [8] Eaton, D., Forouhideh, F.; *Microseismic Moment Tensors: the Good, the Bad and the Ugly*, Department of Geoscience, University of Calgary, Calgary, Alberta, Canada, 2010, <<http://csegrecorder.com/articles/view/microseismic-moment-tensors-the-good-the-bad-and-the-ugly>> (accessed in 10/29/2014).
- [9] Inglis, C.E.; Stresses in a Plate Due to the Presence of Cracks and Sharp Corners, *Transactions from Royal Institute of Naval Architects*, London, United Kingdom, 1913.
- [10] Irwin, G.R.; Analysis of Stresses and Strains Near the End of a Crack Traversing a Plate, *Journal of Applied Mechanics* **24**:361-364, 1957.
- [11] Melo, F.; *Introdução à Mecânica da Fratura* (in Portuguese), Department of Mechanical Engineering, University of Aveiro, Aveiro, Portugal, 2013.
- [12] Seabra, M.; *Modelo Computacional para a Fratura Baseado no Conceito de Elementos Finitos Generalizados* (in Portuguese), Master dissertation in Mechanical Engineering, Faculty of Engineering of University of Porto, Porto, Portugal, 2008.

- [13] Nuismer, R.; An Energy Release Rate Criterion for Mixed Fracture, *International Journal of Fracture* **11**:245-250, 1975.
- [14] Nishioka, T.; A Numerical Study of the Use of Path Independent Integrals in Elastic-dynamic Crack Propagation, *Engineering Fracture Mechanics* **18**:23-33, 1983.
- [15] Erdogan, F., Sih, G.; On the Crack Extension in Plates Under Plane Loading and Transverse Shear, *Journal of Basic Engineering* **85**:519-527, 1963.
- [16] Liang, J., Huang, R., Prevost, J., Suo, Z.; Evolving Crack Patterns in Thin Films With the Extended Finite Element Method, *International Journal of Solids and Structures* **40**:2343-2354, 2003.
- [17] Teixeira-Dias, F., Pinho-da-Cruz, J., Fontes Valente, R.A., Alves de Sousa, R.J.; *Método dos Elementos Finitos: Técnicas de Simulação Numérica em Engenharia* (in Portuguese), ETEP, Lisbon, Portugal, 2010.
- [18] Bathe, K.-J.; *Finite Element Procedures*, Englewood Cliffs, Prentice-Hall, New Jersey, United States of America, 1996.
- [19] Zienkiewicz, O.C., Taylor, R.L., Zhu, J.Z.; *The Finite Element Method: Its Basics and Fundamentals*, 5th edition, Elsevier/Butterworth-Heinemann, Oxford, United Kingdom, 2000.
- [20] Courant, R.; Variational Methods for the Solution of Problems of Equilibrium and Vibration, *Bulletin of the American Mathematical Society* **49**:1-23, 1943.
- [21] Turner, M.J., Clough, R.W., Martin, H.C., Topp, L.J; Stiffness and Deflection Analysis of Complex Structures, *Journal of the Aeronautical Sciences* **23**:805-8823, 1956.
- [22] Greenstadt, J; On the Reduction of Continuous Problems to Discrete Form, *IBM Journal of Research and Development* **3**(4):355-363, 1959.
- [23] Clough, R.W.; *The Finite Element Method in Plane Stress Analysis*, Proceedings of the 2nd ASCE Conference on Electronic Computation 345-378, Pittsburgh, USA, 1966.
- [24] Zienkiewicz, O.C., Cheung, Y.K.; Finite Elements in the Solution of Field Problems, *The Engineer* **220**:507-510, 1965.
- [25] Zienkiewicz, O.C., Cheung, Y.K.; *The Finite Element Method in Structural and Continuum Mechanics*, McGraw-Hill, London, United Kingdom, 1967.
- [26] Abdelaziz, Y., Nabbou, A., Hamouine, A.; A Survey of the Extended Finite Element, *Computers & Structures* **86**(11):1141-1151, 2008.
- [27] Sukumar, N., Prévost, J.-H.; *Modeling Quasi-Static Crack Growth With the Extended Finite Element Method Part I: Computer Implementation*, Princeton University, New Jersey, United States of America, 2003
- [28] Kwon, Y.W., Bang, H.; *Finite Element Method Using Matlab*, 2nd Ed., CRC, Florida, United States of America, 2000.

- [29] Irons, B.M., Draper, K.; Inadequacy of Nodal Connections in a Stiffness Solution for Plate Bending, *AIAA Journal* **3**:965-966, 1965.
- [30] Felippa, C.A.; *Introduction to Finite Element Methods*, Department of Aerospace Engineering Sciences and Center for Aerospace Structures, University of Colorado, USA, 2004.
- [31] Fries, T.P.; The Extended Finite Element Method, Project Overview, <http://www.xfem.rwth-aachen.de/Project/Overview/PRJ_Overview.php> (accessed in 10/29/2014).
- [32] Fries, T.P., Belytschko, T.; The Extended/Generalized Finite Element Method: An Overview of the Method and Its Applications, *International Journal for Numerical Methods in Engineering* **84**(3):253-304, 2010.
- [33] Belytschko, T., Black, T.; Elastic Crack Growth in Finite Elements With Minimal Remeshing, *International Journal for Numerical Methods in Engineering* **45**(5):601-620, 1999.
- [34] Moës, N., Dolbow, J., Belytschko, T.; A Finite Element Method for Crack Growth Without Remeshing, *International Journal for Numerical Methods in Engineering* **46**(1):131-150, 1999.
- [35] Daux, C., Moës, N., Dolbow, J., Sukumar, N., Belytschko, T.; Arbitrary Cracks and Holes With the Extended Finite Element Method, *International Journal for Numerical Methods in Engineering* **48**(12):1741-1760, 2000.
- [36] Sukumar, N., Moës, N., Moran, B., Belytschko, T.; Extended Finite Element Method for Three-Dimensional Crack Modeling, *International Journal for Numerical Methods in Engineering* **48**(11):1549-1570, 2000.
- [37] Belytschko, T., Moës, N., Usui, S., Parimi, C.; Arbitrary Discontinuities in Finite Elements, *International Journal for Numerical Methods in Engineering* **50**(4):993-1013, 2001.
- [38] Stolarska, M., Chopp, D.L., Moës, N., Belytschko, T.; Modeling Crack Growth by Level Sets and the Extended Finite Element Method, *International Journal for Numerical Methods in Engineering* **51**(8):943-960, 2001.
- [39] Sukumar, N., Chopp, D.L., Moës, N., Belytschko, T.; Modeling Holes and Inclusions by Level Sets in the Extended Finite-Element Method, *Computer Methods in Applied Mechanics and Engineering* **190**(46-47):6183-6200, 2001.
- [40] Wells, G.N., Sluys, L.J.; A New Method for Modelling Cohesive Cracks Using Finite Elements, *International Journal for Numerical Methods in Engineering* **50**(12):2667-2682, 2001.
- [41] Moës, N., Belytschko, T.; Extended Finite Element Method for Cohesive Crack Growth, *Engineering Fracture Mechanics* **69**(7):813-833, 2002.

- [42] Wells, G.N., Sluys, L.J., de Borst, R.; A p-Adaptive Scheme for Overcoming Volumetric Locking During Plastic Flow, *Computer Methods in Applied Mechanics and Engineering* **191**(29-30):3153-3164, 2002.
- [43] Zi, G., Belytschko, T.; New Crack Tip Element for XFEM and Applications to Cohesive Cracks, *International Journal for Numerical Methods in Engineering* **57**:2221-2240, 2003.
- [44] Sukumar, N., Chopp, D.L.; Moran, B.; Extended Finite Element Method and Fast Marching Method for Three Dimensional Fatigue Crack Propagation, *Engineering Fracture Mechanics* **70**(1):29-48, 2003.
- [45] Chopp, D.L., Sukumar, N.; Fatigue Crack Propagation of Multiple Coplanar Cracks With the Coupled Extended Finite Element Method/Fast Marching Method, *International Journal of Engineering Science* **41**(8):845-869, 2003.
- [46] Sukumar, N., Srolovitz, D.J., Baker, T.J., Prévost, J.-H.; Brittle Fracture in Polycrystalline Microstructures With the Extended Finite Element Method, *International Journal for Numerical Methods in Engineering* **56**(14):2015-2037, 2003.
- [47] Duflot, M.; A Study of Representation of Crack With Level Sets, *International Journal for Numerical Methods in Engineering* **70**:1261-1302, 2007.
- [48] Asferg, J., Poulsen, P., Nielson, L.; A Consistent Partly Cracked XFEM Element for Cohesive Crack Growth, *International Journal for Numerical Methods in Engineering* **72**:464-485, 2007.
- [49] Meschke, G., Dumstorff, P.; Energy Based Modeling of Cohesive and Cohesionless Cracks via XFEM, *Computer Methods in Applied Mechanics and Engineering* **196**:2339-2357, 2007.
- [50] Wagner, G., Moës, N., Liu, W.K., Belytschko, T.; The Extended Finite Element Method for Rigid Particles in Stokes Flow, *International Journal for Numerical Methods in Engineering* **51**(3):293-313, 2001.
- [51] Merle, R., Dolbow, J.; Solving Thermal and Phase Change Problems With the Extended Finite Element Method, *Computational Mechanics* **28**(5):339-350, 2002.
- [52] Chessa, J., Smolinski, P., Belytschko, T.; The Extended Finite Element Method (XFEM) for Solidification Problems, *International Journal for Numerical Methods in Engineering* **53**(8):1959-1977, 2002.
- [53] Hansbo, A., Hansbo, P.; A Finite Element Method for the Simulation of Strong and Weak Discontinuities in Solid Mechanics, *Computer Methods in Applied Mechanics and Engineering* **193**(33-35):3523-3540, 2004.
- [54] Osher, S., Sethian, J.; Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations, *Journal of Computational Physics* **79**:12-49, 1988.

- [55] Villanueva, C.H., Yu, K.; *A Complete Methodology for the Implementation of XFEM Inclusive Models*, Department of Mechanical Engineering, University of Colorado Boulder, Colorado, United States of America 2003.
- [56] Liang, L.Z., Rui, Z.C., Liang, Z.Y., Bo, Z.H.; *M-Integral for Stress Intensity Factor Base on XFEM*, College of Civil Engineering, Henan Polytechnic University, Jiao Zuo, Henan, China, 2010.
- [57] Shih, C., Asaro, R.; Elastic-plastic Analysis of Cracks on Bimaterial Interfaces: Part I – Small Scale Yielding, *Journal of Applied Mechanics* **55**:299-316, 1988.
- [58] Pais, M.; *2D Extended Finite Element Method Matlab Code*, v1.2, <<http://www.matthempais.com>> (accessed in 10/29/2014).
- [59] Silva, V.D.; *Mecânica e Resistência dos Materiais* (in Portuguese), 2nd Ed., ZUARI, Coimbra, Portugal, 1999.
- [60] Tada, H., Paris, P., Irwin, G.; *The Stress Analysis of Cracks Handbook*, 3rd Ed., ASME Press, New York, United States of America, 2000.
- [61] Yau, J., Wang, S., Corten H.; A Mixed Mode Crack Analysis of Isotropic Solids Using Conservation Laws of Elasticity, *Journal of Applied Mechanics* **47**:335-341, 1980.
- [62] Ferreira, A.M.; *Problemas de Elementos Finitos em Matlab* (in Portuguese), Calouste Gulbenkian Foundation, Lisbon, Portugal, 2010.
- [63] Oliveira, J.; *Modelação Micromecânica do Comportamento de Materiais Compósitos de Matriz de Alumínio* (in Portuguese), Master dissertation in Mechanical Engineering, Department of Mechanical Engineering, University of Aveiro, Aveiro, Portugal, 2006.
- [64] Sigmund O.; A 99 Line Topology Optimization Code Written in Matlab, *Structural and Multidisciplinary Optimization* **21**:120-127, Springer-Verlag, 2001.
- [65] Smith, I.M., Griffiths, D.V.; *Programming the Finite Element Method*, 3rd Ed., John Wiley & Sons, Chichester, United Kingdom, 1997.

This page was intentionally left blank.

Appendices

Appendix A

XFrac-2D manual

XFrac-2D is a 2-D eXtended Finite Element Method (XFEM) software developed for Octave. It is capable of modelling 2-D cracks, in quadrilateral plates, when subjected to several different boundary conditions, and enables the determination of the crack growth path. The XFrac-2D software was done in a way that the user only needs to do the pre-processing step and then run the `programaXFEM.m` file. The following subsections present a brief explanation of what each subroutine does, a detailed description of the input variables in the `inputData.m` file and, finally, the program flowchart.

A.1 Pre-processing

The pre-processing step is done by the modification of the `inputData.m` file. In this file, the user chooses the input parameters: size of the plate, mesh density, problem properties, loads and displacement boundary conditions, number of iterations and the graphical representation.

The software also uses this file to create the problem connectivity. This step is done through 4 subroutines. It starts by determining the FEM connectivity, using the subroutine `conectividade.m`. At this time, the software creates the FEM connectivity table with the nodes belonging to each elements and the coordinates table, where all the nodes' initial coordinates are. Then it applies the level-set method to all the elements and detects the enriched ones; this is done by the `levelSet.m` subroutine. This process not only detects the enriched elements but also assigns the nodes to their enrichment properties. After that, the software calculates and stores the crack coordinates for each enriched element using the subroutine `coordfenda.m`. This process is very important, since it incorporates the crack into the enriched elements, enabling the creation of the subdomain. Before this process, the crack is just a connection between two points in the `inputData.m` file. Finally, the subroutine `conectividadePE` gathers all the previous created information and generates the XFEM connectivity. This process creates and enumerates the new degrees of freedom from the pseudo-elements and creates the connectivity between the standard FEM nodes and the XFEM pseudonodes.

The resulting connectivity is stored in the `Conectividade` matrix. Each row represents an element, where the first column is the element number and the 4 others have the number of the nodes belonging to that element. Here, the order is very important. The nodes are numbered according to Figure 3.3, where the 1st node number corresponds to

the node number 1 of the element, and so on.

The XFEM connectivity and enrichment values are stored in the `ConectividadeX` matrix. Each row represents the XFEM properties in each FEM node. The 1st column is the FEM node number, the 2nd one is the Heaviside pseudonode number (it is 0 if the element isn't enriched with the Heaviside function), the 3rd, 4th, 5th and 6th columns have the four asymptotic pseudonode numbers (their value is 0 if the element isn't enriched with the asymptotic function), the 7th column is the nodal Heaviside value, the 8th and 9th columns have the radius and angle between the node and the crack tip, respectively (they will be used in the asymptotic enrichment of the node).

The initial coordinates of each node are stored in the `Coordeandas` matrix. Here, once again, each row represents a node and the first 3 columns have, respectively, the x , y and z coordinates. The 4th column has the number of nodes that are connected to this node.

A.2 Processing

During the processing step, each element stiffness matrix is built and assembled into the global stiffness matrix (called `KG`), the force vector is created and the equations system is reduced and solved. First, the software creates the element stiffness matrices and assembles them into `KG` (the global stiffness matrix). Then, it creates the XFEM enriched stiffness matrices, for each pseudonode, and assembles them into the same `KG`. This process is done, respectively, with the `assembly.m` and `EnriAssembly.m` subroutines. Inside `KG`, the data is stored by the way described in Section 5.4. After that, the subroutine `loads.m` creates the external force vector (`Loads`), which is the FEM load vector, and the subroutine `updateLoads.m` adds the new degrees of freedom created by the XFEM. Since the external force in the pseudonodes is always zero, this updating process only adds new rows to the original force vector, being zero their values. The application of the displacement boundary conditions is done by the `BoundaryConditions.m` subroutine and is exactly the same as in a FEM software. The matrix in `KG` and the vector `Loads` are reduced, creating the `KR` matrix and the `LoadsR` vector and, then, the equation system is solved by the `solve.m` subroutine, leading to the reduced nodal displacement vector `DeslocamentosR`. Then, the expansion of the reduced nodal displacement vector results in the global displacement vector `Deslocamentos`.

A.3 Post-processing

The displacement vector, `Deslocamentos`, is initially composed by the FEM and the XFEM displacements. The vector `DeslocamentosT` is created with only the FEM displacements and the XFEM displacements are incorporated into it. After that, the strains and stresses can be determined for each element. Since the strains and stresses are usually discontinuous between the elements, the nodal average strain/stress is used as a smoothing process. All these processes are done by the `juncaocorrecta.m` subroutine. Then, the mode I and II Stress Intensity Factor (SIF) values are determined, for each crack tip, with the calculation of the interaction integral, in subroutine `IntegralJ.m`, and stored in the `KI` and `KII` vectors, respectively. With the values of K_I and K_{II} , the new crack coordinates can be determined by subroutine `CrescimentoFenda.m` and added

to the original matrix with the crack coordinates (**Crack**). The final subroutine, **plots.m**, creates the graphical representations of the mesh, the crack, the enriched element, the M-integral radius, the strains and the stresses. During the pre-processing step, the user defines the graphical representation that will be created in the **plots.m** subroutine.

A.4 Modelling examples

This section presents the modelling of some cases in XFrac-2D. The cases are the same ones analysed in Chapter 6. As written before, all the process is done with the modification of the **inputData.m** file.

All the edges numbers are according to the numeration done in Figure A.1.

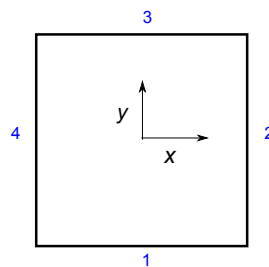


Figure A.1: XFrac-2D nodal and edge numeration.

The properties are equal in all the problems and are modelled by the following code lines:

```
E=10000;           %Young's modulus [PA]
nu=0.3;           %Poisson's ratio
RaioJ=3;         %M-integral Radius [R]
Estado=2;       %Plane strain state
```

Central crack

Figure A.2 shows the geometry and boundary conditions for this problem.

The domain of the central crack problem has 20×50 mm² and 22×55 elements and it is modelled by the following code lines:

```
NelmX=22; %number of elements in x
NelmY=55; %number of elements in y

comprimentoX=20; %length in x [mm]
comprimentoY=50; %length in y [mm]
```

A 2000 N load is applied, in the y -direction, to the edge number 3, and the edge number 1 is simply supported. The load, the boundary condition, the crack dimension/location and the plots are modelled by the following code lines:

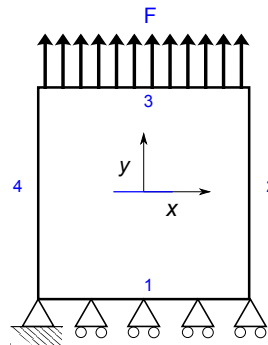


Figure A.2: XFrac-2D nodal and edge numeration.

```

Forca=[0 2000 3];    %2000N in Y direction applied in edge number 3
BC=[1 1];           %Boundary condition (edge number 1 simply supported)
Crack=[6 25
       14 25];      %Coordinates of the two points composing the crack
PLOTS=zeros(3,5);   %No plots

```

If graphical representations are wanted, the change of the respective value in the `PLOTS` matrix must be done, according to the `inputData.m` part presented in the subroutines list.

Edge crack

Figure A.3 shows the geometry and boundary conditions for this problem. The domain

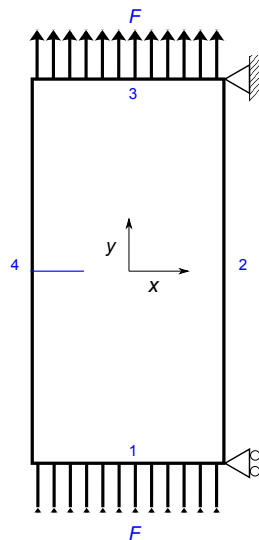


Figure A.3: XFrac-2D nodal and edge numeration.

of the edge crack problem has $10 \times 50 \text{ mm}^2$ and 19×99 elements, and it is modelled by

the following code lines:

```
NelmX=19; %number of elements in x
NelmY=99; %number of elements in y

comprimentoX=10; %length in x [mm]
comprimentoY=50; %length in y [mm]
```

1000 N and -1000 N loads are applied, in the y -direction, to the edges number 3 and 1, respectively, being the boundary conditions those shown in Figure 6.7. The loads, the boundary condition, the crack dimension/location and plots are modelled by the following code lines:

```
Forca=[0 1000 3 %1000 N in y direction applied in edge number 3
        0 -1000 1]; %-1000 N in y direction applied in edge number 1

BC=[2 4]; %Boundary condition (restriction of just 2 nodes)

Crack=[-1 25
        4.1 25]; %Coordinates of the two points composing the crack

PLOTS=zeros(3,5); %No plots
```

Again, if graphical representations are wanted, the change of the respective value in the `PLOTS` matrix must be done, according to the `inputData.m` part presented in the subroutines list.

Slanted crack

Figure A.4 shows the geometry and boundary conditions for this problem. The domain

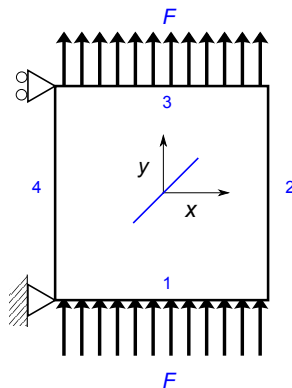


Figure A.4: XFrac-2D nodal and edge numeration.

of the slanted crack problem is a square 20×20 mm², with a mesh of 35×35 elements, and is modelled by the following code lines:

```
NelmX=35; %number of elements in x
NelmY=35; %number of elements in y
```

```
comprimentoX=20; %length in x [mm]
comprimentoY=20; %length in y [mm]
```

This time, 9 different cracks are modelled. The following code lines represent the coordinates of these different cracks. All of them have an 8 mm length, but different inclination angles (α). The software is run for each crack individually, meaning that all the remaining ones are signed as comments in the script.

```
Crack=[6 10
      14 10]; %0a crack
%-----
Crack=[6.0608 9.3054
      13.9392 10.6946]; %10a crack
%-----
Crack=[6.2412 8.6319
      13.7588 11.3681]; %20a crack
%-----
Crack=[6.5359 8
      13.4641 12]; %30a crack
%-----
Crack=[6.9358 7.4288
      13.0642 12.5712]; %40a crack
%-----
Crack=[7.4288 6.9358
      12.5712 13.0642]; %50a crack
%-----
Crack=[8 6.5359
      12 13.4641]; %60a crack
%-----
Crack=[8.6319 6.2412
      11.3681 13.7588]; %70a crack
%-----
Crack=[9.3054 6.0608
      10.6946 13.9392]; %80a crack
```

A 2000 N and -2000 N load are applied, in the y -direction, to the edges number 3 and 1, respectively, being the boundary conditions those shown in Figure 6.9. The loads, the boundary condition and the plots are modelled by the following code lines:

```
Forca=[0 2000 3 %2000 N in y direction applied in edge number 3
      0 -2000 1]; %-2000 N in y direction applied in edge number 1

BC=[4 4]; %Boundary condition (restriction of just 2 nodes)

PLOTS=zeros(3,5); %No plots
```

Once again, if graphical representations are wanted, the change of the respective value in the PLOTS matrix must be done, according to the inputData.m part presented in the subroutines list.

Crack propagation

The propagation process can be modelled by the following code lines in any of the previous examples:


```
Crescimento=1.0;           %Crack increment in each iteration [mm]
Niteracao=3                %Number of iterations performed
```

The first variable is the crack length added in each increment and the second variable is the number of increments. The previous examples were performed without the consideration of the crack propagation process, meaning that numerically only one iteration was performed. To keep the accuracy of the graphical representation, the XFrac-2D software uses the data from the previous iteration.

A.5 Subroutines list

This Section presents the a description list of all the subroutines of0 the XFrac-2D software.

assembly.m

Creates the stiffness FEM matrix for each element and assembles it in the global stiffness matrix.

BoundaryConditions.m

Deletes some specific columns and/or rows in the forces vector, the displacement vector and the global stiffness matrix. In this way, it applies the boundary conditions of the problem, in order to solve the system linear equations.

conectividade.m

Creates the FEM connectivity matrix among the nodes and the FEM elements.

conectividadePE.m

Creates the XFEM connectivity matrix among the pseudonodes and the FEM elements.

coordfenda.m

Detects which elements are completely cut by the crack and stores the coordinates

CrescimentoFenda.m

Calculates the crack propagation angle (β), determines the new crack coordinates and adds them to the Crack matrix.

EnriAssembly.m

Creates the elementary stiffness matrix with the enrichment functions for each element and assembles it in the global stiffness matrix.

EnriAssim.m

Calculates the values of the enrichment functions at the nodes enriched with the crack tip function.

EnriHeavi.m

Calculates the values of the enrichment functions at the nodes enriched with the Heaviside function.

inputData.m

Is the file related to the pre-processing data.

There, E , ν and $estado$ are material properties where,

E – is the Young's modulus [MPa].

ν – is the Poisson's ratio.

$Estado$ – is the state of the problem. $Estado=1$ for a plane strain state and $Estado=2$ for a plane stress state.

The next variables are related the kind of analysis performed.

$RaioJ$ – is the M-integral radius.

$Crescimento$ – is the length the crack grows in each iteration.

The following variables define the rectangular domain and the mesh density:

$comprimentoX$ – is the length in the x -direction.

$comprimentoY$ – is the length in the y -direction.

$NelmX$ – is the number of elements in the x -direction.

$NelmY$ – is the number of elements in the y -direction.

Forca

Is the force matrix, where each row represents a force in a different application point.

$Forca(1,1)$ – force magnitude in the x -direction.

$Forca(1,2)$ – force magnitude in the y -direction.

$Forca(1,3)$ – the application point.

BC

Is the boundary condition matrix, where each row represents a boundary condition and an application point/edge.

$BC(1,1)$ – is the edge number.

$BC(1,2)$ – is the type of boundary condition. $BC(1,2)=1$ for a simply supported edge, 2 for a pinned edge, 3 for the two middle points pinned in the selected edge and 4 for one edge with one of its vertex simply supported and the other vertex pinned.

Crack

Is the matrix with the crack coordinates, where each row represents a point belonging to the crack vector:

$Crack(1,1)$ – x coordinate of the first crack point.

$Crack(1,2)$ – y coordinate of the first crack point.

Plots

This variable controls which plots are to be output by the matlab code. For all cases, Yes = 1 and No = 0.

Plots(1,1) – plots the initial mesh.
Plots(1,2) – plots the FEM deformed mesh.
Plots(1,3) – plots the crack into the domain.
Plots(1,4) – plots the nodes inside de M-integral domain.
Plots(1,5) – plots the triangle subdomain in the elements.
Plots(2,1) – plots the FEM stresses for the x -direction.
Plots(2,2) – plots the FEM stresses for the y -direction.
Plots(2,3) – plots the FEM sheer stresses for the xy -direction.
Plots(3,1) – plots the initial mesh.
Plots(3,2) – plots the XFEM deformed mesh.
Plots(3,3) – plots the M-integral domain.
Plots(3,4) – plots the M-integral search radius.
Plots(4,1) – plots the XFEM stresses for the x -direction.
Plots(4,2) – plots the XFEM stresses for the y -direction.
Plots(4,3) – plots the XFEM sheer stresses for the xy -direction.

IntegralJ.m

Calculates the stress intensity factor for every iteration (K_I and K_{II}).

juncaocorrecta.m

Joins the XFEM with the FEM displacements in the enriched nodes.

levelSet.m

Applies the level-set functions and detects the elements that must be enriched by the XFEM enrichment function.

loads.m

Creates the FEM external load vector.

plotFenda.m

Plots the crack.

plots.m

Plots the original and deformed meshes, the enriched elements, the M-integral domain and the stresses.

plotTriangulos.m

Plots the triangle subdomain.

programaXFEM.m

Is the main script, it must be run after the input data is defined in the inputData.m file is changed with the problem data. This file calls most of the subroutines.

solve.m

Obtains the displacements, inverting the reduced stiffness matrix and multiplying it by the reduced external load vector.

triangulos.m

Creates the triangle subdomain, using the `deLaunay` function.

updateLoads.m

Updates the FEM external load vector with the new degrees of freedom created by the XFEM enrichment.

A.6 XFrac-2D flowchart

In Figure A.5, the orange subroutines belong to finite element method and the blue subroutines belong to eXtended Finite Element Method.

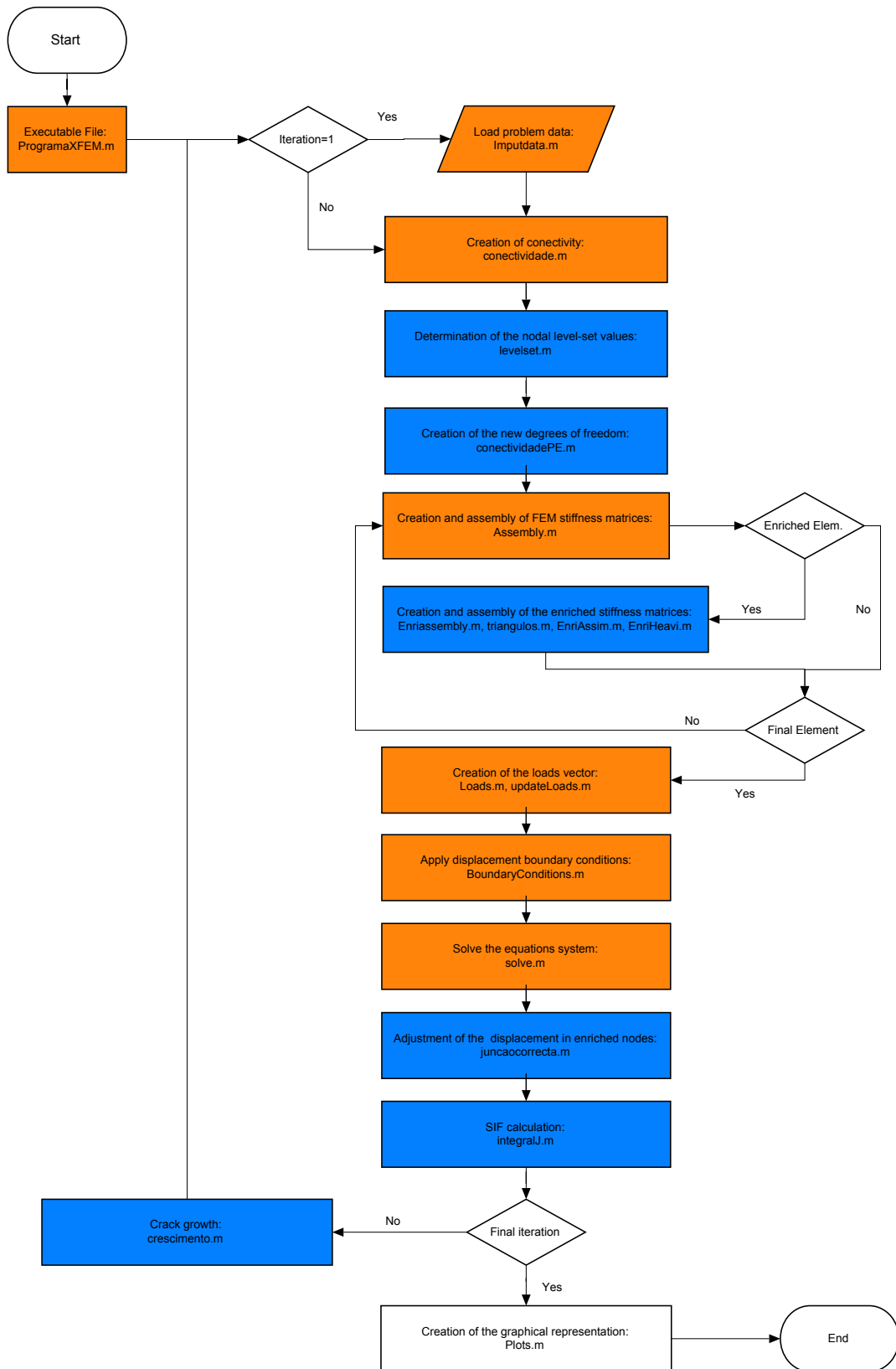


Figure A.5: Flowchart of the XFrac-2D software.

This page was intentionally left blank.

Appendix B

List of Tables

This page was intentionally left blank.

List of Tables

3.1	Coordinates and weights of the 4 Gauss points in a square element.	17
6.1	Displacement and errors for the considered meshes.	48
6.2	Properties of the plate with a central crack.	52
6.3	Properties of the plate with an edge crack.	53
6.4	Influence of the type of stress and mesh refinement for a plate with a lateral crack.	54
6.5	K_I [MPa.mm ^{1/2}] for all the different meshes and radiuses.	55
6.6	Relative error [%] for all the different meshes and radiuses.	56
6.7	Properties for the plate with a slanted crack.	57

This page was intentionally left blank.

Appendix C

List of Figures

This page was intentionally left blank.

List of Figures

2.1	Modes of failure in Fracture Mechanics [8].	6
2.2	The J-integral path around a crack.	8
3.1	Shape functions for four elements with two nodes each.	13
3.2	Body Ω	13
3.3	Isoparametric representation (right) of a four nodes quadrilateral element defined in Oxy (left).	15
3.4	Typical flowchart of a FEM software.	19
4.1	Body with an internal crack and corresponding boundary conditions.	23
4.2	Values of the level-set functions.	29
4.3	Flowchart of a standard XFEM software.	31
5.1	Stiffness matrix configuration.	37
5.2	Division of the quadrilateral enriched elements into triangles.	40
6.1	Geometry and boundary conditions for the tensile test.	48
6.2	Geometry and boundary conditions for the bending test.	49
6.3	Incorrect detection of the enriched elements.	49
6.4	Geometry and boundary conditions for the plate with a central crack.	51
6.5	Representation of the mesh and the crack in the plate.	51
6.6	Stress in y -direction [MPa].	53
6.7	Geometry and boundary conditions of the plate with an edge crack.	54
6.8	SIF and relative error for the mesh refinement analysis.	55
6.9	Geometry and boundary conditions of the plate with a slanted crack.	57
6.10	SIF and relative error for the 20×20 mm ² plate.	58
6.11	SIF and relative error for the 40×40 mm ² plate.	59
6.12	Initial and final crack path after the crack growth analyses.	60
6.13	Initial, after 4 iterations and final crack path after the crack growth analysis.	61
6.14	SIF and error of K_I for each iteration during the crack propagation.	62
A.1	XFrac-2D nodal and edge numeration.	77
A.2	XFrac-2D nodal and edge numeration.	78
A.3	XFrac-2D nodal and edge numeration.	78
A.4	XFrac-2D nodal and edge numeration.	79
A.5	Flowchart of the XFrac-2D software.	85