UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MECÁNICA

FRACTURE MODELLING DIRECTLY FROM COMPUTER-AIDED DESIGN (CAD) BY THE EXTENDED ISOGEOMETRIC FINITE ELEMENT METHOD (X-IGA FEM) WITH TRIMMED NURBS

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MECÁNICO

JAVIER ANDRÉS VIDELA MARIÓ

PROFESOR GUÍA:
ELENA ATROSHCHENKO

MIEMBROS DE LA COMISIÓN:
ALEJANDRO ORTIZ BERNARDIN
VIVIANA MERUANE NARANJO

SANTIAGO DE CHILE
2017

FRACTURE MODELLING DIRECTLY FROM COMPUTER-AIDED DESIGN (CAD) BY THE EXTENDED ISOGEOMETRIC FINITE ELEMENT METHOD (X-IGA FEM) WITH TRIMMED NURBS

Tanto los software de Diseño Asistido por Computadora (CAD) como las herramientas de análisis por medio del Método de Elementos Finitos (FEM) han tenido un enorme impacto en la actividad ingenieril en las últimas décadas. Aun así, tienen la desventaja de que las geometrías CAD no son directamente compatibles con las geometrías utilizas en FEM, lo cual resulta en la necesidad de re-mallar la geometría varias veces durante un ciclo de soluciones FEM. Para solucionar esto, el Análisis Isogeométrico (IGA) ha sido propuesto como una metodología capaz de generar un vínculo directo entre el diseño mediante CAD y el análisis FEM. La principal idea dentro de IGA es sustituir las funciones de forma utilizadas en FEM por las funciones de base que utiliza el software CAD, conocidas como NURBS.

Por otro lado, los problemas de mecánica de fractura presentan dificultades extras para los métodos numéricos debido a las irregularidades que presenta la geometría de la grieta y el campo de esfuerzos singular que se origina alrededor de la punta de la grieta. Para solucionar esto, una nueva generación de métodos numéricos ha sido desarrollada, la cual incluye el renombrado XFEM.

La misma idea del XFEM ha sido formulada dentro del contexto de IGA, conocido como Análisis Isogeométrico Extendido (XIGA) y ha sido aplicado exitosamente en grietas rectas y curvas en problemas de elasticidad 2-D. Sin embargo, los trabajos publicados en este tema se han limitado a problemas simples.

El objetivo de este trabajo es aplicar el XIGA en problemas de mecánica de fractura con grietas en geometrías complejas dadas por curvas CAD. El trabajo consiste en implementar la metodología de enriquecimiento del XFEM dentro de un código IGA para solucionar problemas de mecánica de fractura 2D.

La primera simulación consiste en el problema de una placa infinita con una grieta recta. Esta es resuelta utilizando enriquecimiento Heaviside y Heaviside con punta de grieta. La segunda simulación consiste en emplear NURBS cuadráticos y cúbicos para resolver el problema de un agujero circular con una grieta. En ambas simulaciones se calculan las normas de error L^2, H^1 y energética, y el factor de intensidad de esfuerzos (SIF) K_I para evaluar los resultados.

Ambos problemas exhiben convergencia sub-óptima al momento de refinar la malla y el K_I está en concordancia con la solución analítica.

Abstract

Both the CAD software and FEM software have a huge impact on engineering nowadays. Even though both are powerful tools for design and analysis, the main drawback is that CAD geometries and Finite Element models do not completely match, which results in the necessity to re-parametrize the geometry many times during the solution cycle in FEM. Isogeometric Analysis (IGA) was proposed to fulfill this gap and create the direct link between the CAD design and FEM analysis. The main idea of IGA is to substitute the shape functions used in FEM by the shape functions used in the CAD software.

Nevertheless, fracture mechanics problems present difficulty for any numerical method due to the irregularity in the crack domain and the singularity of the stress field in the vicinity of the crack tip. To overcome this difficulty, a new generation of numerical methods has been developed, which includes the so-called Extended Finite Element Method (XFEM).

The same idea was implemented within the IGA (XIGA) and applied successfully for straight and curved cracks in two-dimensional elasticity. However, the published work is limited to simple crack configurations.

The objective of this work is to investigate the applicability and efficiency of the XIGA for cracks of complex geometries given by CAD curves. The work consists in implementing the enrichment method of the XFEM into the in-house IGA code and in verifying the approach on a number of benchmark problems.

Heaviside and Heaviside + crack tip enrichment has been implemented for a benchmark problem of an edge crack and the results are shown to be in excellent agreement with the analytic solution. The comparison of the results is done in L^2, H^1 and Energy norm of the error, which exhibit the sub-optimal convergence rates, when the mesh size tends to zero.

Quadratic and Cubic NURBS were employed to solve the benchmark problem of an edge crack in a circular hole and the results are shown to be in agreement with the analytic solution, despite the complicity of the geometry.

The stress intensity factor (SIF) K_I is computed and in both benchmark problems it is in agreement with their theoretical value.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Damage tolerant design and nondestructive inspections represent a key task for ensuring that structures operate safely for extended periods of service. Damage tolerance is the ability to resist fracture from pre-existent cracks through a given period of time and is an essential attribute of components whose failure could result in a catastrophic loss of life or property [1].

Many industries are in need of simple and accurate tools to study crack propagation and assess the damage tolerance of various structures. Examples range from the aeronautical industry to the automotive industry. The approach to engineering design to account for damage tolerance is based on the assumption that flaws can exist in any structure and such flaws might propagate with usage.

The Finite Element Method (FEM) is highly popular among different engineering disciplines due to its ability to find an approximate solution to boundary value problems (such as loads and forces, vibrations, fluid flow dynamics, heat transfer and other physical phenomenons). It is used widespread in different engineering disciplines and many commercial programs are available, e.g. Abaqus, Adina, Ansys, Comsol, Nastran, etc. Unfortunately, it presents drawbacks when dealing with crack propagation in fracture mechanics: problems such as re-meshing during crack growth makes FEM time/resources expensive.

To overcome these difficulties, different methods have been developed such as Embedded Elements (EFEM) [2], Element free Galerkin methods (EFG) [3], Extended Finite Element Method (XFEM) [4], among others. Between them, the XFEM has earned world-wide acceptance as an effective tool for crack propagation problems.

On the other hand, another difficulty presented by FEM is having to work with complex geometries. The Computer Aided Design (CAD) software, such as AutoCAD, Inventor, Solid-Works, NX among others, are a powerful tool that allows design engineers to use computers in order to create, modify, analyze, optimize and visualize a design. CAD software mostly uses NURBS (Non Uniform Ration B-splines) based curves and geometries. FEM software receives CAD geometry and converts it to a FEM model by discretizing it. This process is known as meshing. The main drawback is that for complex geometries, the mesh does not

match properly with the geometry and there is a loss in accuracy of the solution (especially in stress concentration zones). Figure 1.1 shows a CAD model and different meshes. As seen in the highlighted zone, the complex meshing process must be done in order to approximate the geometry.
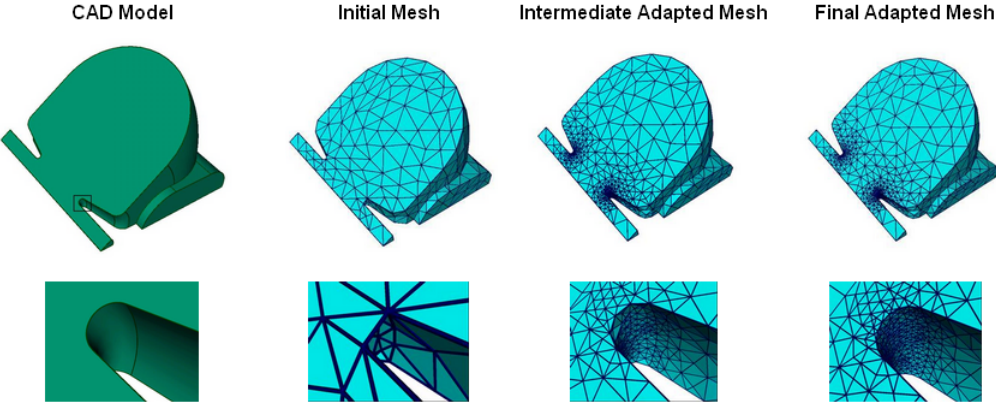
Figure 1.1: Differences between CAD model and mesh.
Source: [5].

To overcome this difficulty, Hughes et al. [6] proposed the Isogeometric Analysis (IGA) as an alternative to merge both FEM and CAD.

IGA is a recently developed computational approach that fills the gap the between CAD and FEM by using the conventional NURBS functions of the CAD design tools into the FEM method. The main idea of IGA is to use the same basis functions (splines) for both: parametrization of the geometry (computational domain) and approximation of the unknown field (see Figure 1.2). Due to a number of advantages offered by the spline basis functions over the standard finite element methods (such as higher continuity and exact representation of the geometry), IGA has transformed in the past years into one of the most efficient tools for numerical analysis and has found many applications in various areas in mechanics.

(a) IGA imput: CAD geometry

(b) IGA output: Distribution of stresses
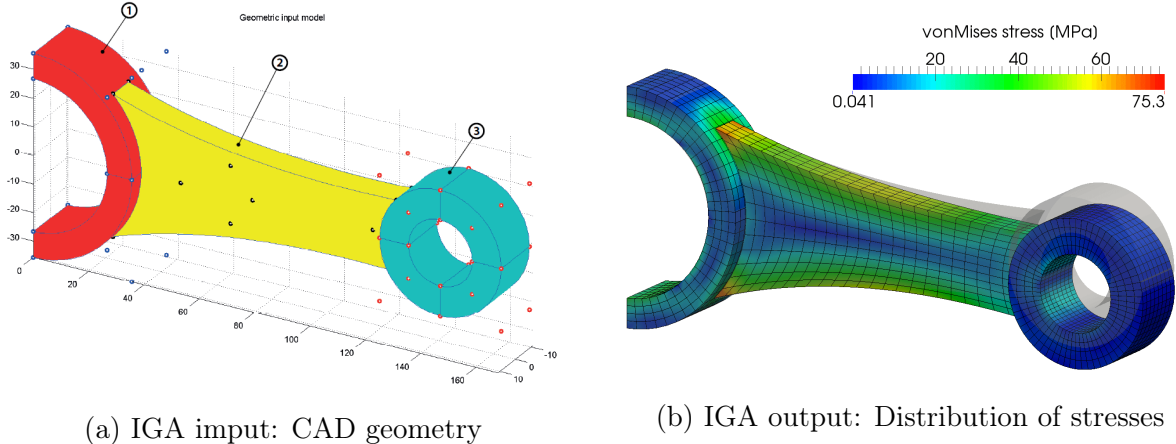
Figure 1.2: Basic idea of IGA.
Source: [7].

Luycker et al. [8] and Ghorashi et al. [9, 10] employed the enrichment criteria of the XFEM method in the isogeometric formulation to solve linear fracture mechanical problems.

This new formulation is known as the Extended Isogeometric Analysis (XIGA) and has been applied successfully for straight [8, 9] and curved cracks [10]. However, the method has only been applied to both simple domain geometries and simple geometries of cracks.

In this work we are going to investigate the accuracy and efficiency of the XIGA for cracks with complex geometries given by CAD curves, with special emphasis on the follow two novelties:

a. The crack is also modelled by a NURBS curve.

b. Cut elements are treated with trimmed NURBS methods [11, 12].

The work will consist in implementing the enrichment method of the XFEM into the in-house IGA code and in verifying the approach on a number of benchmark problems.

1.1 Motivation

Nowadays, engineers mostly design using CAD software and perform analyses by using FEM commercial software. The typical situation in an engineering practice is the one where designs are encapsulated in CAD systems, meshes are generated from CAD data and FEM are performed over the meshes. This allows a totally different geometric description for analysis, which is only approximate. In some instances, mesh generation can be done automatically but in most circumstances it can be done at best semi-automatically. There are still situations in major industries where drawings are made of CAD designs and meshes are built from them. It is estimated that about 80% of overall analysis time is devoted to mesh generation in the automotive, aerospace, and ship building industries [6].

To close the gap between the FEM and CAD, Hughes et al. [6] proposed Isogeometric Analysis (IGA) as a method to use the original CAD geometry and the corresponding NURBS functions to approximate solution in the FEM.

Besides that, fracture mechanical problems present a challenge for the FEM due to irregularity of the crack domain and stress singularity near the crack tip. The challenge becomes even greater when modelling crack propagation because not only the mesh is complex, but the geometry itself changes, leading to a bad convergence of the solution near the crack. Over the years, the principal focus was to update the mesh every time the crack grows, so that the FEM can match the geometry and continue to solve the problem.

The following methods have been proposed in order to accomplish this: Embedded Elements (EFEM) [2], Element free Galerkin methods (EFG) [3], Extended Finite Element Method (XFEM) [4], and others.

Moës et al. [4] proposed the XFEM in order to solve crack growth problems by using the concept of enrichment nodes in the solution. By doing this, they avoided the re-meshing process every time the crack tip would move.

The Extended Isogeometric Analysis (XIGA) [8] is the merge of IGA with the enrichment philosophy of the XFEM. This approach benefits from the advantages of its roots, XFEM and IGA, and is capable of analyzing crack growth problems without any re-meshing. Complex geometries can be modeled with few elements and higher order inter-element continuities are satisfied. XIGA has been successfully applied for the analysis of straight cracks in both parametric and physical spaces [8, 9]. It has also been applied to curved cracks on the parametric space [10].

In view of the above, the main purpose of this project work is to extend the study of XIGA by solving complex crack/geometry problems with XIGA by including the parametrization of the crack as a NURBS curve and using the concept of trimmed NURBS.

1.2 Objectives and Scope

1.2.1 General Objectives

The main objective of this project work is to develop a methodology for the numerical analysis of fracture mechanical problems, based on the XIGA method.

1.2.2 Specific Objectives

The specific objectives of this project work are:

- Formulate of the XFEM in the Framework of IGA.
- Implement Enrichment of the solution into IGA code to make it suitable for fracture.
- Implement the evaluation of J-Integrals for the Stress Intensity Factors.
- Verify the results on a set of benchmark problems.

1.2.3 Scope

The scope of this work are:

- The XIGA code solves two dimensional (2D) problems.
- The XIGA code is tested on specific 2D benchmark problems.

Chapter 2

Methodology

The methodology in this project work is based on the implementation of an Extended Isogeometric Analysis routine for the resolution of 2D crack propagation problems. Therefore the main steps are as follows:

a. Literature Review: The first step consists in performing a literature review of the principal topics that cover this project: Fracture Mechanics, Isogeometric Analysis (IGA), Extended Finite Element Method (XFEM) and Extended Isogeometric (XIGA).

b. IGA code: This step involves coding in C++ language a routine capable of performing IGA over a 2D domain for elasticity problems.

c. Enrichment of the IGA code (XIGA): This next step consists in programming the functions required in order to enrich both in intrinsic and extrinsic ways (analogously to XFEM). This part includes the following functions:

    i. Crack-Element identification: A routine capable of identifying the elements which are cut by the crack, enabling it to generate a list of elements enriched by Heaviside or crack tip criteria.

    ii. Integration: A routine capable of integrating elements which are cut by the crack or contain a crack tip inside.

    iii. Enrichment Assembly: A routine capable of performing a new assembly for the stiffness matrix considering the crack tip and Heaviside enrichment. The functions used for the enrichment in the crack tip will be obtained by the formulation of the stress fields using stress intensity factors for each benchmark problem.

    iv. Post-processing: A routine capable of calculating the final deformation of the problem, the stress field, the J_1 integral, L^2 norm of the error, H^1 norm of the error and $Energy$ norm of the error.

d. Testing of benchmark problems: This fourth step consists of testing the XIGA code with a set of benchmark problems in order to verify the implementation.

e. Result Analysis: After obtaining the results from the XIGA code the last step is to analyze those results and compare them with the analytical and numerical solutions obtained by other methods, which are available in the literature.

The methodology is illustrated in Figure 2.1.

Figure 2.1: Methodology flow-sheet.

2.1 Resources

The resources for this project are related to the software employed for this project. For all the IGA and XIGA routines, the cross platform *Code Blocks* will be used, which is a free C++ IDE available for Windows, OS X and Linux operating systems. For all the post-processing and generation of graphics the software Matlab will be used, which is a numerical computing software available for Windows, OS X and Linux operating systems.

Chapter 3

Literature Review and Theoretical Background

The purpose of this chapter is to give a brief look at the literature of Isogeometric Analysis, Extended Finite Element Method and Extended Isogeometric analysis and later proceed to expose the theoretical background in which this project work is based.

3.1 Literature Review

Hughes et al. [6] proposed the Isogeometric Analysis as an alternative to FEM. It is inspired by NURBS functions, which is the standard technology employed in CAD. The main purpose of IGA is to have a direct link between both the designer's and engineer's work.

The root idea behind the Isogeometric Analysis is that the basis functions used to exactly model the geometry will also serve as the basis functions for the solution space of the numerical method.

This concept is a very well know idea in Finite Element Analysis (FEA). The main difference is that in FEA, the basis chosen to approximate the unknown solution is the same one used to approximate the geometry, on the other hand, the IGA uses a basis capable of exactly representing the known geometry, also to approximate the unknown solution.

A primary goal in IGA is to be geometrically exact no matter how coarse the discretization is. Another goal is to simplify mesh refinement by eliminating the need for communication with the CAD geometry once the initial mesh is constructed (avoiding the pre-process of meshing, which takes more time than actually solving the FEM problem). Yet another goal is to weave more tightly the mesh generation process within CAD.

There has been plenty of research on IGA. Bazilevs et al. [13] and Da Veiga et al. [14] worked on error estimates and stability analysis on IGA. Several works have been done in computer science of fast and efficient implementations of IGA [15, 16, 17]. Applications of

IGA are widely spread in mechanics: from structural vibrations [18], fluid-structures dynamics [19], Reissner-Mindlin shell analysis [20], cohesive zone modeling [21] to gradient damage modeling [22] and also in non-mechanical areas such as shape sensitivity analysis [23] and optimization [24].

Several methods have been proposed using the philosophy of IGA, for example, NURBS Enhanced finite element method (NEFEM) [25], NURBS Enhanced Extended finite element method (NEXFEM) [26], Isogeometric BEM (IGABEM) [27] and Extended Isogeometric Analysis (XIGA) [8, 9]. IGA has also been formulated using other types of basis functions, for example, T-Splines (which are a generalization of NURBS that allows local refinement) [28, 29], Polynomial Hierarchical T-Splines (PHT-Splines) [30, 31] and Rational Hierarchical T-Splines (RHT-Splines) [32].

The XIGA is the merge between IGA and XFEM. It has the advantages of IGA and the enrichment procedures of the XFEM. XIGA has been successfully applied to cases of straight cracks [8, 9], curved cracks [10] and plates with cracks and multiple holes [33]. It has also been formulated for T-splines [34] and even for crack identification methods [35].

The lasted applications of XIGA include fracture mechanics analysis of piezoelectric materials [36], thermal buckling analysis of functionally graded plates with internal defects [37] and buckling and vibration analysis of imperfect graded Reissner-Mindlin plates with internal defects [38] among others.

3.2 Theoretical Background

3.2.1 Fracture Mechanics

Experimental observations have shown that brittle materials tend to fracture when loaded beyond their critical stress level.

Kirsch and Kolosoff (1898) analyzed the problem of an infinite plate with a circular hole under uniform tensile stress and solved it finding out that the stress near the hole was three times the tensile stress (see Figure 3.1).

Both kolosov (1909) and Inglis (1913) solved the problem of an infinite plate with a elliptical hole subject to uniform tensile stress (Figure 3.2). A solution for a sharp cracked problem can then be obtained by degenerating the elliptical hole into a straightly lined crack.

Griffith (1921,1924) related the fact that there are some materials with less strength resistance compared to their theoretical values, to the presence of inner cracks and flaws. Based on the previous works of Kirsh, Kolosoff, Kolosov and Inglis, Griffith found a relation between the semi-length a of a crack and the tensile stress σ_F by means of the total change in energy as the cracking took place.

Irwin (1957) extended Griffith's work by incorporating the crack tip's yield strength,
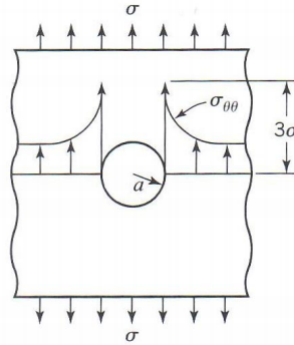
Figure 3.1: Infinite plate plate with a hole under uniform stress.
Source: [39].
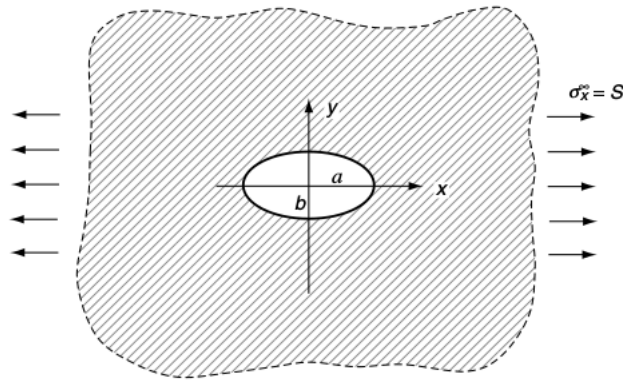
Figure 3.2: Infinite plate plate with an elliptical hole under uniform stress.
Source: [40]

introducing the concept of stress intensity factor K (SIF) as a measure of the strength of the singularity.

Then Rice (1968) introduced the concept of the J integral defined as a independent path contour integral equal to the rate of change of the potential energy for an elastic nonlinear solid during a unit crack extension.

Stern (1976) and Wen (1995) went further by extending the idea to arbitrary mixed mode loading. The importance of those J integrals lies in the fact that they can be related to the Stress Intensity Factors, which are used to formulate crack growth models.

Stress Intensity Factor (SIF) K

The Stress Intensity Factor K was introduced by Irwin as a measure of the strength of the singularity. In a general way, Irwin illustrated that any stress σ_{ij} in a small vicinity of a crack front can be written using the polar coordinates r and θ (see Figure 3.3) in terms of three Stress Intensity Factor K_I, K_{II} and K_{III}:
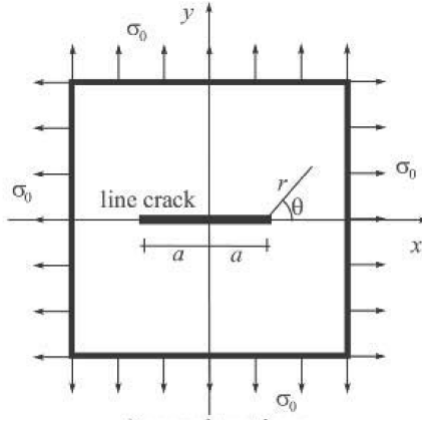
Figure 3.3: Example of crack tip with Cartesian (x, y) and polar coordinates (r, θ).
Source: [41].

$$\sigma_{ij} = \frac{K_I \cdot f_{ij}^I(\theta) + K_{II} \cdot f_{ij}^{II}(\theta) + K_{III} \cdot f_{ij}^{III}(\theta)}{\sqrt{r}} \tag{3.1}$$

where K_I, K_{II} and K_{III} are the SIFs related to three independent crack modes of propagation (Figure 3.4), f_{ij}^I, f_{ij}^{II} and f_{ij}^{III} are functions of θ.
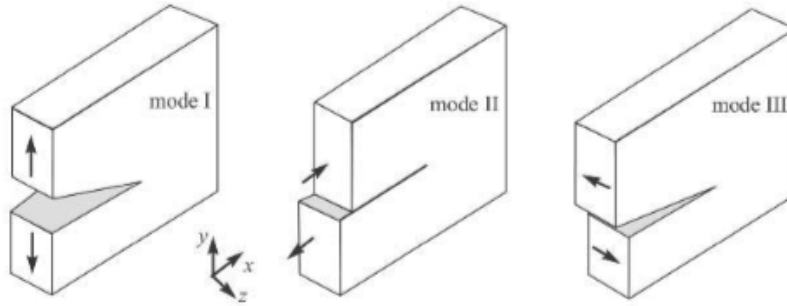
Figure 3.4: Crack modes I, II and III related to K_I, K_{II} and K_{III}, respectively.
Source: [41].

J Integrals

The J_k Integrals are a path independent contour integral equal to the rate of change of the potential energy for an elastic non-linear solid during a unit crack extension. The J_k integral can be taken over any path not including the crack tip, but usually, due to simplicity, a set of circles is used as shown in Figure 3.5.

The J_k-Integral is defined as:
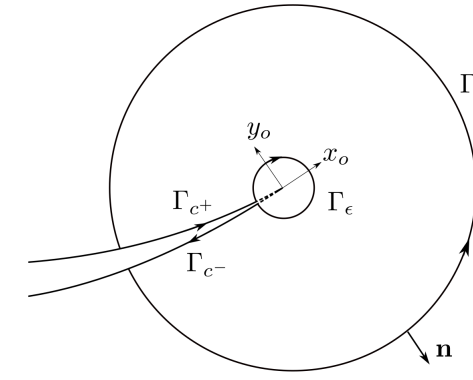
Figure 3.5: Contour for J_k integral.
Source: [42].

$$J_k = \oint_{\Gamma'} \left(W_s \mathrm{d}y - \mathbf{t}\frac{\partial \mathbf{u}}{\partial x} \right) \mathrm{d}\Gamma \tag{3.2}$$

where $\Gamma' = \Gamma_{C^+} + \Gamma_{C^-} + \Gamma_\varepsilon + \Gamma$ is a path that encloses the crack tip, $\mathbf{t} = \sigma \cdot \mathbf{n}$ and W_s is the strain energy density, defined by:

$$W_s = \frac{1}{2}\sigma_{ij}\varepsilon_{ij} \tag{3.3}$$

In a general case the J_k integrals are path depended, and in the particular case of Mode I type problems, the J_1 integral (known as J-integral) is independent to the path [43]. It can also be shown that when J is calculated along a contour around a crack tip, it represents the change in potential energy for a virtual crack extension $\mathrm{d}a$. The total potential energy of a two-dimensional domain including a traction free crack that is surrounded by a contour curve Γ under quasi-static conditions and in the absence of body forces can be defined as (for further details see [41]):

$$\Pi = \int_\Omega W_s \mathrm{d}\Omega - \int_\Gamma t_i u_i \mathrm{d}\Gamma \tag{3.4}$$

For a virtual crack extension $\mathrm{d}a$, the change in potential energy is:

$$\frac{\mathrm{d}\Pi}{\mathrm{d}a} = \int_\Omega \frac{\mathrm{d}W_s}{\mathrm{d}a}\mathrm{d}\Omega - \oint_\Gamma (\frac{\mathrm{d}t_i}{\mathrm{d}a}u_i + \frac{\mathrm{d}u_i}{\mathrm{d}a}t_i)\mathrm{d}\Gamma \tag{3.5}$$

The following can be proved by applying the divergence theorem:

$$J = -\frac{\mathrm{d}\Pi}{\mathrm{d}a} \tag{3.6}$$

11

where $-\frac{d\Pi}{da}$ is the definition of fracture energy release for linear elastic materials: $G = -\frac{d\Pi}{da}$, and the strain energy release rate G is related to each crack mode with the follow equations:

$$J_1 = (K_I^2 + K_{II}^2)\left(\frac{1-\nu^2}{E}\right) \tag{3.7}$$

$$J_2 = -2K_I^2 K_{II}^2 \left(\frac{1-\nu^2}{E}\right) \tag{3.8}$$

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

Li et al. [44] proposed a method for computing the J-Integral using the area inside the contour Γ' by introducing a q function, which is a smooth function that takes the value of 1 in Γ_ε and 0 in Γ. Using the Divergence theorem, it can be proof that the J-Integral is equal to

$$J = \int_A \left[\sigma_{ij}\frac{\partial u_i}{\partial x} - W\delta_{1i}\right]\frac{\partial q_1}{\partial x_j}\mathrm{d}A \tag{3.9}$$

In FEM, usually the inner contour Γ_ε is often taken as the crack tip, and so A naturally corresponds to the area inside Γ. The boundary Γ should also coincide with element boundaries to facilitate numerical calculations.

3.2.2 Problem Formulation for FEM

This section briefly discusses the strong and weak formulation for a 2-D crack propagation problem.

Strong Form

Consider a Body Ω with a boundary Γ as in Figure 3.6. The boundary Γ is composed of $\Gamma_{\bar{u}}$, $\Gamma_{\bar{t}}$ and Γ_C such that $\Gamma = \Gamma_C + \Gamma_u + \Gamma_t$. $\Gamma_{\bar{u}}$ is the boundary with Dirichlet Conditions (Known displacement $\bar{u}$), $\Gamma_{\bar{t}}$ is the boundary with Neumann Conditions (Known traction $\bar{t}$) and Γ_c is the crack surface (which is traction-free), composed by two faces: Γ_{c+} and Γ_{c-}.

The strong form of the elastostatics equation in the absence of body forces is:

$$\triangledown \cdot \boldsymbol{\sigma} = 0 \tag{3.10}$$

with the following boundary conditions:

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{t} \text{ on } \Gamma_{\bar{t}} \tag{3.11}$$
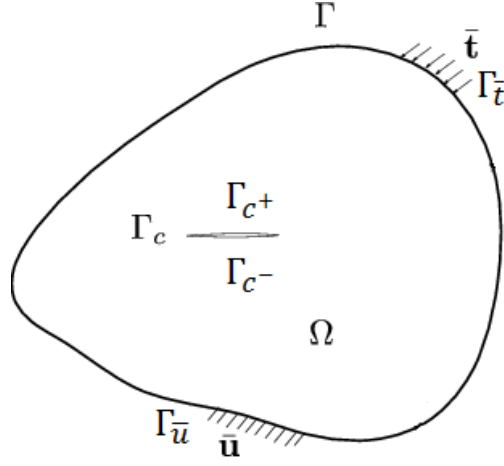
Figure 3.6: Problem formulation.
Source: adapted from [42].

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{c+} \tag{3.12}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{c-} \tag{3.13}$$

$$\boldsymbol{u} = \bar{u} \text{ on } \Gamma_{\bar{u}} \tag{3.14}$$

where $\boldsymbol{n}$ is the unit outward normal, $\boldsymbol{\sigma}$ is the stress tensor and $\boldsymbol{u}$ is the displacement vector.

The relation between the strain tensor $\boldsymbol{\varepsilon}$ and displacement vector is given by:

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T] \tag{3.15}$$

while the relation between the stress tensor and strain tensor is given by Hooke's law:

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \boldsymbol{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{3.16}$$

where $\boldsymbol{C}$ is the Hooke's tensor (4th order) and : is the double dot product. Equation (3.16) can be written in index notation as:

$$\sigma_{ij} = C_{ijkl}\varepsilon_{kl} \tag{3.17}$$

In 2D, the stresses and strains component of $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ can we written as:

13

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} \tag{3.18}$$

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_{xy} \end{Bmatrix} \tag{3.19}$$

For the plane stress case and isotropic material, the Hooke's tensor $\boldsymbol{C}$ can be written as:

$$\boldsymbol{C} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 1-\nu \end{bmatrix} \quad \text{(plane stress)} \tag{3.20}$$

and for plane strain problems with isotropic material, the Hooke's tensor $\boldsymbol{C}$ can be written as:

$$\boldsymbol{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & 1-2\nu \end{bmatrix} \quad \text{(plane strain)} \tag{3.21}$$

Weak Form

The Weak form of the equilibrium equation is given by:

$$\int_{\Omega} (\sigma(\boldsymbol{u}) : \varepsilon(\boldsymbol{w})) \, \mathrm{d}\Omega = \int_{\Gamma} (\bar{t} \cdot \boldsymbol{w}) \, \mathrm{d}\Gamma \tag{3.22}$$

where $\boldsymbol{w}$ belongs to the space of all vector functions whose derivatives are square-integrable which vanish on Ω_u and are discontinuous on Ω_c. $\boldsymbol{u}$ belongs to the space of all vector functions whose derivatives are square-integrable which satisfy the essential boundary condition. Details of this can be found at Appendix A.

Finite Element Method - FEM

The Finite Element Method (FEM) is a widely numerical technique used to obtain an approximated solution of the distribution field variables (in this work: strains, stresses and displacements) which is difficult to obtain analytically. It is done by dividing the domain (discretizing the problem) into a set of elements, where each element is formed by nodes. The continuous field variable is approximated by *shape functions* in each element. The discrete values of the field variable at the node are the unknown. Each element is related to the

others by a governing equation (in this work it is the weak form of the problem: Equation (3.22)). This equation leads to different algebraic relations between the unknown variables. The complete procedure to obtain and solve this method goes as follows:

Domain Discretization   The first step is to divide the solid body Ω into N elements Ω_e, such that each one cannot overlap and the union of the N elements must be Ω. Each element Ω_e is formed by nodes which are connected in a predefined consistent way to create the connectivity of the system.

This process is called meshing and is done by the so called pre-processor in the commercial FEM software. (See Figure 3.7)
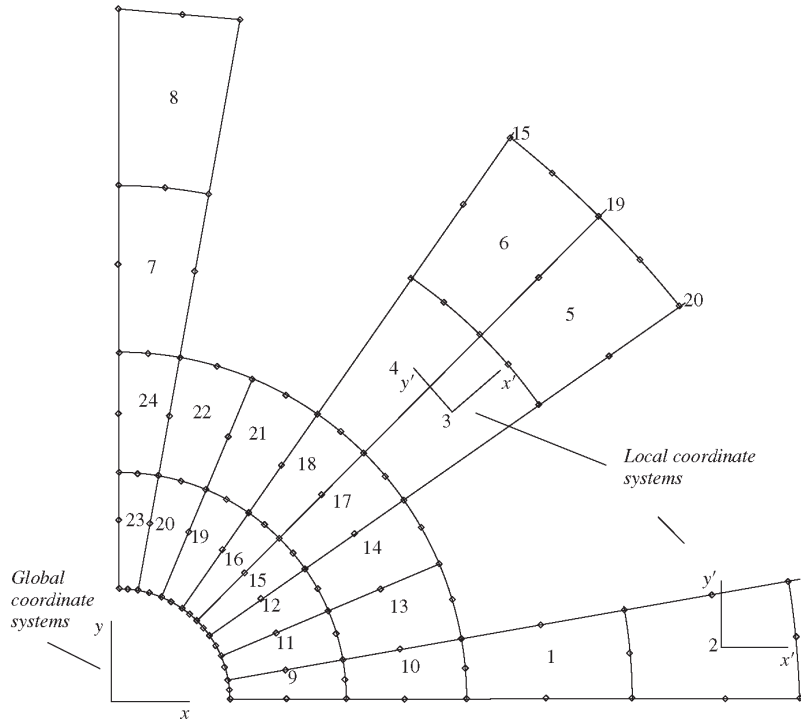
Figure 3.7: Example of a mesh with numbered elements, local and global coordinate system. Source: [45].

Displacement Interpolation: The second step is to interpolate the main function, in this case the displacement $\boldsymbol{u}$ from the weak form problem (Equation (3.22)). This interpolation is done by the *Shape Functions* **N**. Let **u** be the displacement of a point, $\mathbf{u^h}$ the displacement of the nodes (which is the unknown value that FEM wants to find), n_d the number of nodes and n_f the degree of freedom of each node. The displacement $\boldsymbol{u}$ can be approximated by:

$$\boldsymbol{u}(x,y) = \sum_{i=1}^{n_\mathrm{d}} N_i \cdot \boldsymbol{u^h}(x_i, y_i) \tag{3.23}$$

15

In Equation (3.23) N_i is the matrix of shape functions related to the node i, which is arranged as:

$$N_i = \begin{bmatrix} N_{i1} & 0 & 0 & 0 \\ 0 & N_{i2} & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & N_{in_f} \end{bmatrix} \tag{3.24}$$

Formulation of FEM in each element: After Discretizing the domain and obtaining the shape functions, the weak form of the problem (Equation (3.22)) must be moved from the global coordinate system to the local coordinate system. In the FEM context, this is done by selecting the weight function $\boldsymbol{w}$ as the shape functions $\boldsymbol{N}$ to then apply certain properties of $\boldsymbol{N}$ which reduce the weak form to a set of local equations per element [45, 46]. This results in a linear algebraic equation for each element:

$$\boldsymbol{K_e} \cdot \boldsymbol{U_e} = \boldsymbol{f_e} \tag{3.25}$$

where $\boldsymbol{K_e}$ is the stiffness matrix of the element in the local coordinate system, $\boldsymbol{U_e}$ is the displacement of the element in the local system and $\boldsymbol{f_e}$ is the force matrix in the local system.

Assembly of the global equation: Given that there is one system of equations per element, and adjacent elements share nodes, the Equation (3.25) must be assembled to obtain equation:

$$\boldsymbol{K} \cdot \boldsymbol{U} = \boldsymbol{f} \tag{3.26}$$

where $\boldsymbol{K}$ is known as the global stiffness matrix, $\boldsymbol{U}$ is the vector containing unknown values of u^h and $\boldsymbol{f}$ is the force field.

Application of boundary conditions: After assembling the global equation the Dirichlet and Neumann boundary conditions need to be applied. The Neumann conditions are applied in the formulation of the weak form, while the Dirichlet boundary conditions are imposed to the global assembly equation (Equation (3.26)).

Solving the equation: The final step in the FEM procedure is to solve Equation (3.26), which is done by inverting the stiffness matrix $\boldsymbol{K}$:

$$\boldsymbol{U} = \boldsymbol{K^{-1}} \cdot \boldsymbol{f} \tag{3.27}$$

With this, we only have to use Equation (3.23) to obtain the displacement field u.

Shape Functions

All the FEM formulation is based on a coordinate system. The mesh is described in general coordinates but each shape function is described in a local coordinate system (see Figure 3.7). The definition of the shape functions is done in a local coordinate system so it can be used for any element, independent of the element's size.

The shape functions **N** have the following properties:

a. Consistency property: The consistency property states that if the complete order of monomial used to build N_i is k, then the shape function is said to possess a consistency C^k.

b. Reproduction property: The reproduction property says that if there is a function $f(x)$ which can be written in terms of

$$f(\boldsymbol{x}) = \sum_{j}^{k} p_j(\boldsymbol{x}) \cdot \beta_j \tag{3.28}$$

where p_j are monomials and β_j are constants. So there exists a set of f_i such that:

$$f(x) = \sum N_i \cdot f_i \tag{3.29}$$

c. Linear Independence: The shape functions meet the property of linear independence.

d. Delta Function property: The shape functions have the delta function property:

$$N_i(\boldsymbol{x_j}) = \delta_{i,j} = \begin{cases} 1 \text{ , if } i = j \\ 0 \text{ , otherwise} \end{cases} \tag{3.30}$$

e. Partition of unity property: The shape functions form a partition of the unity, i.e. :

$$\sum N_i(\boldsymbol{x}) = 1 \tag{3.31}$$

at any $\boldsymbol{x}$

One of the most used shape functions is the linear shape function. In Figure 3.8, there is an example of a 2D element and its transformation from the global coordinate system x and y to the local coordinates ξ and η.

As an example, the Bi-linear shape functions for a 2D element are (see Figure 3.9):

$$N_1(\xi, \eta) = \frac{1}{2} \cdot (1 - \xi) \cdot \frac{1}{2} \cdot (1 - \eta) \tag{3.32}$$

2D Global coordinate system   2D Local coordinate system
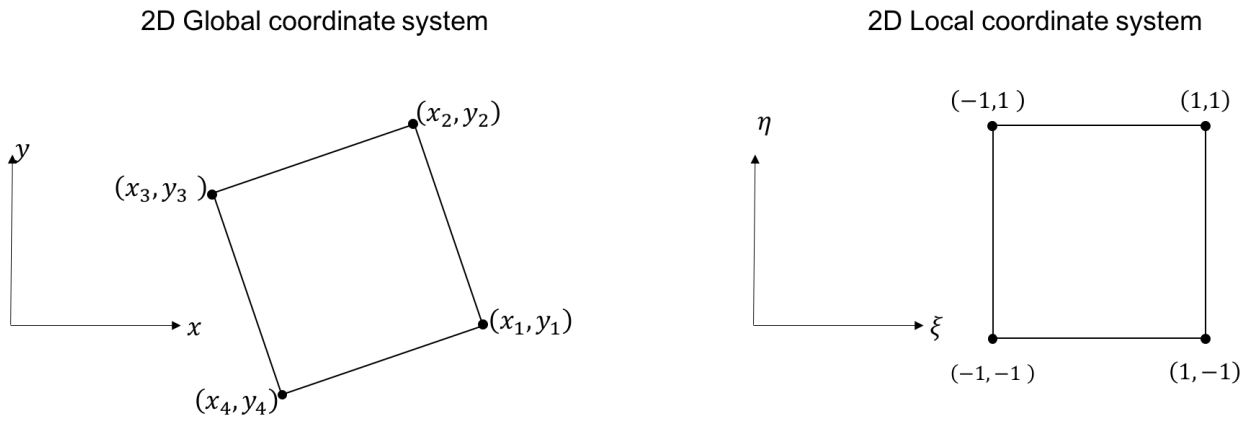
Figure 3.8: Relationship between global and local coordinates in a 2D element.
Source: prepared by the author.

$$N_2(\xi, \eta) = \frac{1}{2} \cdot (1 - \xi) \cdot \frac{1}{2} \cdot (1 + \eta) \tag{3.33}$$

$$N_3(\xi, \eta) = \frac{1}{2} \cdot (1 + \xi) \cdot \frac{1}{2} \cdot (1 - \eta) \tag{3.34}$$

$$N_4(\xi, \eta) = \frac{1}{2} \cdot (1 + \xi) \cdot \frac{1}{2} \cdot (1 + \eta) \tag{3.35}$$
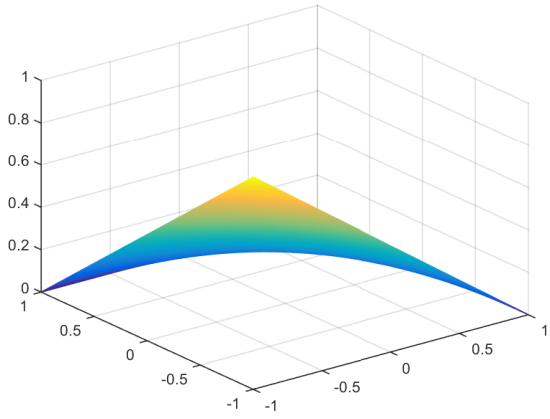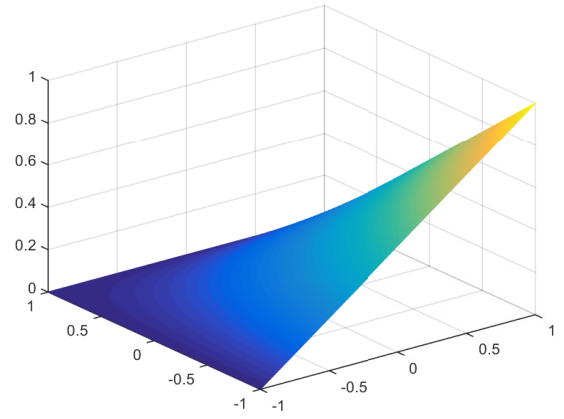
(a) N_1 (b) N_2
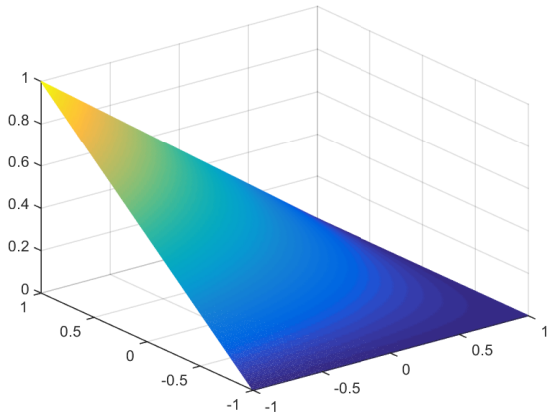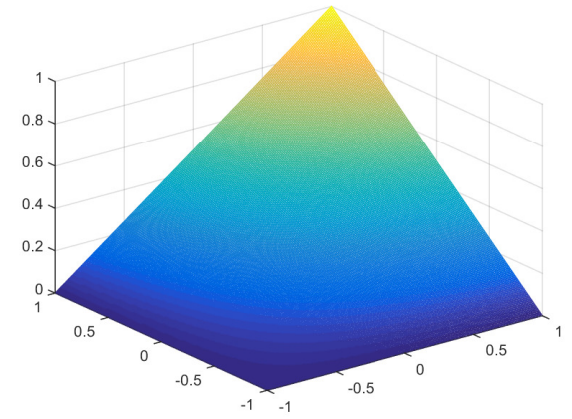
(c) N_3 (d) N_4

Figure 3.9: Bi-linear shape functions N_1, N_2, N_3 and N_4 for a 2D element. Source: Prepared by the author.

3.2.3   Isogeometric Analysis - IGA

This section briefly reviews the basic concepts of IGA.

Non Uniform Rational B-Spline NURBS

Let $\Xi = \xi_0, ..., \xi_{n+p+1}$ be a non-decreasing sequence of real numbers, i.e., $\xi_i < \xi_{i+1}, i = 0, ..., n+p+1$. The ξ_i are called knots and Ξ is called the knot vector. n is the number of basis functions which comprise the NURBS and p is the polynomial order of the NURBS. If knots are equally-spaced in the parametric space, they are said to be *uniform*. If they are unequally spaced, then they are *non-uniform*.

A $p - th$ degree NURBS curve is defined by:

$$C(\xi) = \sum_{i=1}^{n} R_{i,p}(\xi)\mathbf{P_i} \tag{3.36}$$

where $\mathbf{P_i}$ are the control points of the curve and $R_{i,p}$ are the Rational basis functions, defined by:

$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{j=0}^{n} N_{j,p}(\xi)w_j} \tag{3.37}$$

where w_i are the weights associated to each $R_{i,p}(\xi)$ and $N_{i,p}(\xi)$ are the p-th degree B-Spline basis functions, which are defined in a recursive way:

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i < \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+1} - \xi_i}N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}}N_{i+1,p-1}(\xi) \tag{3.38}$$

Figure 3.10 shows an example of a 2D B-spline curve. The Control points are denoted by the red dots and the curve is denoted by the blue line.
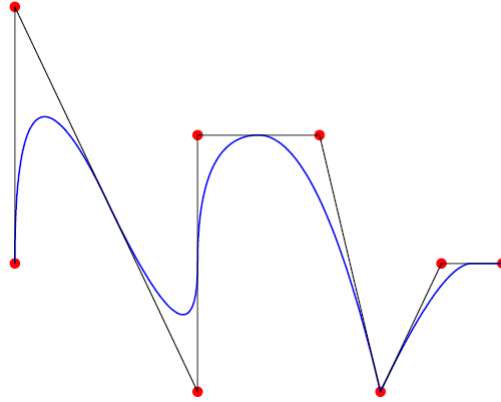
Figure 3.10: B-spline, piece-wise quadratic curve in 2D. Control point locations are denoted by red dots.

Source: [6].

Analogously, a NURBS surface of p-th degree in the ξ direction and q-th degree in the η direction is defined by:

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi, \eta)\mathbf{P_{ij}} \tag{3.39}$$

where $\mathbf{P_{ij}}$ is a set of $n \times m$ bidirectional control net and $R_{i,j}^{p,q}$ is defined by:

$$R_{i,j}^{p,q}(\xi, \eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)}{\sum_{\hat{i}}^{n} \sum_{\hat{j}}^{m} N_{\hat{i},p}(\xi)M_{\hat{j},q}w_{\hat{i},\hat{j}}} \tag{3.40}$$

where $N_{\hat{i},p}(\xi)$ and $M_{\hat{j},q}$ are B-splines functions defined on knots sets Ξ_1 and Ξ_2, respectively.

Figure 3.11 shows an example of 2D basis functions of order $p = q = 1$ (bi-linear), $p = q = 2$ (bi-quadratic) and $p = q = 3$ (bi-cubic). For the bi-linear there are four corner basis function (the same as the standard element method, Figure 3.9), there are four corner basis functions, four edge basis functions and one internal basis function for bi-quadratic and there are four corner basis functions, eight edge basis functions and one internal basis function for bi-cubic.

The B-spline basis functions have the following properties:

- Linear independence.
- Partition of unity.
- Compact support for each $N_{i,p}$ in the interval $[u_i, u_{i+p+1}]$.
- Non-negative basis functions.

Since NURBS are transformations of the B-spline basis functions they inherit their main properties and obtain more. Some of them include:

- Partition of unity.
- Continuity and support of B-spline.
- Affine transformations in a physical space are achieved by applying them to the control points. This means NURBS have the property of affine covariance.
- If the weights of all control points are equal, NURBS become B-splines.
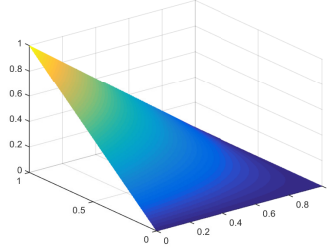
(a) Bi-linear NURBS: Corner basis function
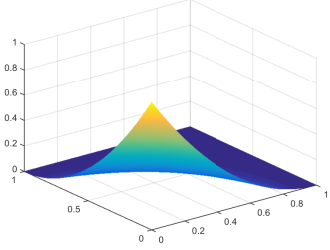
(b) Bi-quadratic NURBS: Corner basis function
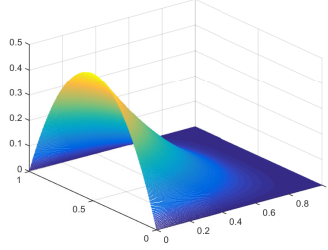
(c) Bi-quadratic NURBS: Edge basis function
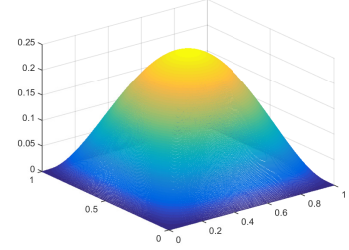
(d) Bi-quadratic NURBS: Internal basis function
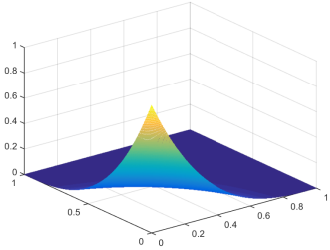
(e) Bi-cubic NURBS: Corner basis function
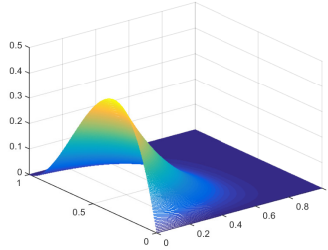
(f) Bi-cubic NURBS: Edge basis function
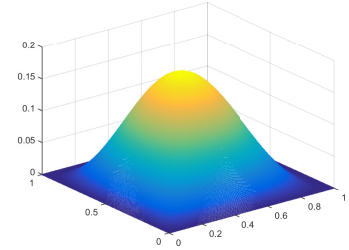
(g) Bi-cubic NURBS: Internal basis function

Figure 3.11: Bi-linear, Bi-quadratic and Bi-cubic basis function for a uniform knot vector $\Xi_1 = \Xi_2 = \{0, 1, 2, 3, 4, ...\}$. Source: prepared by the author.

Solution Discretization

In the NURBS based Isogeometric concept, the physical field is approximated by the same NURBS basis functions as those used to describe the geometry. The physical domain is denoted by Ω and the parametric domain by $\hat{\Omega}$ (which is the domain that includes the knot vector), the mapping from the parametric domain to the physical domain is then given by:

$$\boldsymbol{X} = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\boldsymbol{\xi}) \cdot \boldsymbol{P_{ij}} \tag{3.41}$$

where $\boldsymbol{\xi} = (\xi, \eta)$ are the parametric coordinates from the NURBS system, $\boldsymbol{X} = (x, y)$ are the physical coordinates, $R_{i,j}^{p,q}$ are the basis of NURBS functions and $\mathbf{P_{ij}}$ are the control points.

22

The isoparametric formulation for the displacement is:

$$\boldsymbol{u} = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\boldsymbol{\xi}) \cdot \boldsymbol{u_{ij}} \tag{3.42}$$

where $\boldsymbol{u_{ij}}$ are the control variables. In this formulation an element is defined in the parameter space as an interval between two consecutive non-repeated knots $[\xi_a, \xi_b]$. By applying this formulation to the weak form equation (Equation (3.22)) the same equation as in Equation (3.26) is obtained with the control variables as the unknowns. The details can be found in [47]

Isogeometric Analysis - IGA

Since IGA uses NURBS to match the geometry exactly, it slightly changes the way the FEM algorithm works. NURBS are defined on "patches" rather than elements. Patches play the role of sub-domains within element types and material model assumed to be uniform [6].

A summary of the similarities and differences between FEA and IGA is presented in Table 3.1. In [6] and [48] expand further in detail.

Table 3.1: Comparison between FEA and IGA based on NURBS

Comparison of Finite Element Analysis and Isogeometric Analysis based on NURBS	
Differences	
Finite Element Analysis	Isogeometric Analysis
Nodal points	Control Points
Nodal Variables	Control Variables
Mesh	Knots
Basis interpolates nodal points and variables	Basis does not interpolate control points and variables
Approximate geometry	Exact geometry
Gibbs phenomena	Variation diminishing
Sub-domains	Patches
Polynomial basis	NURBS Basis
Similarities	
Compact support	
Partition of unity	
Isoparametric concept	
Affine covariance	
Patch test satisfied	

Source: Adapted from [6].

According to Hughes, the framework of IGA is:

a. A mesh for NURBS patch is defined by the product of knot vector. For a 2D problem it distinguishes between the physical space given by the coordinates (x, y) and the parametric space given by a NURBS plane (ξ, η).

b. Knot spans subdivide the domain into "elements".

c. The support of each basis function consists of a small number of "elements".

d. The control points associated with the basis functions define the geometry.

e. The isoparametric concept is applied to the unknown field (e.g., displacement). The field is represented in terms of the same basis functions as the geometry. The coefficients of the basis functions are the degrees-of-freedom.

f. Arrays constructed from isoparametric NURBS patches can be assembled into global arrays in the same way as finite elements.

g. Boundary conditions are applied in a similar way as in the FEM. Neumann boundary conditions are satisfied naturally, while Dirichlet boundary conditions are applied to the control variables (using the affine covariance property of the NURBS).

h. Solve the problem the same way as in the FEM (Equation (3.27)) to find the coefficients of the basis functions of the field ($\mathbf{u_{ij}}$) which will be use to approximate the solution in every point of the physical space by using the transformation between the physical and parametric space.

3.2.4   Extended Finite Element Method

In 1996 Melek and Babuska [49] introduced the concept of Partition of the Unity Finite Element Method (PUFEM). The main idea is to define a set of m functions $f_k(x)$ within a domain Ω_{pu} such that:

$$\sum_{k=1}^{m} f_k(x) = 1 \tag{3.43}$$

with the following property for any $g(x)$ function:

$$\sum_{k=1}^{m} (f_k(x)g(x)) = g(x) \tag{3.44}$$

In the FEM the shape functions N_j are a set of Partition of Unity (PU) functions.

The concept of partition of unity provides a mathematical framework for the development of an enriched solution, which is used by the XFEM method. Further details can be found in Mohammadi [41].

Enrichment

Theoretically speaking, the concept of enrichment consist in to increase the order of completeness that can be achieved. It is done with the goal to increase the accuracy of the solution (especially in cases like crack tip solution).

The choice of the enriched function depends on the *a priory* knowledge about the solution of the problem. For instance, in a crack analysis, enrichment functions represent the asymptotic behavior of the solution near the crack tip [41].

There are two types of enrichment procedures: Intrinsic enrichment and Extrinsic enrichment. The first one consists in adding new basis functions to the shape functions $\boldsymbol{N}$ and proceed with the normal FEM procedure. On the other hand, the extrinsic enrichment consists in adding new degrees of freedom (DOF) to the system and using a new basis of linear independent functions for those enriched degrees, i.e. the solution is defined in the form:

$$\mathbf{u}(x,y) = \sum N_i(x,y) \cdot \mathbf{u^h} + \sum N_k(x,y) \sum \mathbf{c_l} \cdot f_l(x,y) \tag{3.45}$$

where c_l are the new degrees related to the f_l enrichment functions. This type of enrichment involves a new assembly procedure in order to include the new degrees of freedom into the Equation (3.26). This approach is known as Partition of Unity FEM (PUFEM).

Methods such as the PUFEM and the Generalized Finite Element Method(GFEM) use both intrinsic and extrinsic enrichment. The PUFEM uses the classical finite element shape functions $N_j(x)$ in both types of enrichment. In the GFEM, different shape functions are used for the classical and enriched parts of the approximation.

The Extended Finite Element Method (XFEM), proposed by Moës et al. [4], uses the same concept of intrinsic and extrinsic enrichment but instead of applying it in all the domain, the enrichment is applied only at a local level. For example, in crack problems only a small region around the crack tip is enriched.

In the framework of crack tip problems, Moës et al. [4] implemented a discontinuous extrinsic enrichment by using a Heaviside function enrichment for the nodes related to the crack face and an asymptotic enrichment for the nodes related to the crack tip. Figure 3.12 shows a crack modeling mesh with blue circles representing the Heaviside enrichment and red squares representing the crack tip enrichment. The following equation shows both types of enrichment:

$$\mathbf{u}(x,y) = \sum N_i(x,y) \cdot \mathbf{u^h} + \sum N_j(x,y) \cdot \mathbf{d_j} \cdot H(x,y) + \sum N_k(x,y) \sum \mathbf{c_l} \cdot f_l(x,y) \tag{3.46}$$

where $\mathbf{d_j}$ are the new degrees of freedom related to the crack face and $\mathbf{c_l}$ are the new degrees of freedom related to the crack tip. The Heaviside function is defined in the local crack

coordinate system as follows:

$$H(x, y) = \begin{cases} 1 \text{ for nodes above the crack} \\ -1 \text{ for nodes bellow the crack} \end{cases} \tag{3.47}$$
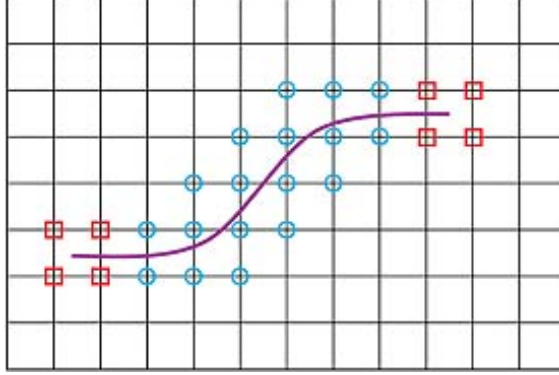
Figure 3.12: Enrichment: Heaviside enrichment is represented by blue circles and crack tip enrichment is represented by red squares.

Source: [50].

The basis functions for the crack tip enrichment are defined as:

$$\{f_l(r, \theta)\} = \{\sqrt{r}sin(\theta/2), \sqrt{r}cos(\theta/2), \sqrt{r}sin(\theta/2)sin(\theta), \sqrt{r}cos(\theta/2)sin(\theta)\} \tag{3.48}$$

where (r, θ) is the local polar coordinate system at the crack tip (see Figure 3.13).
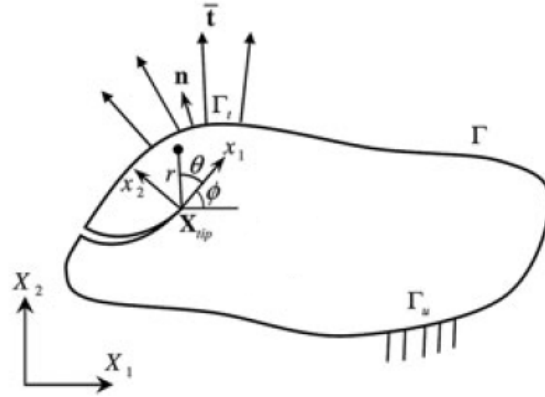
Figure 3.13: Local and global coordinates at the crack tip.

Source: [9].

The transformation between the local polar coordinate system to the local Cartesian coordinate system at the crack tip is as follows:

$$\begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = arctan(\frac{y}{x}) \end{cases} \tag{3.49}$$

26

The following is the transformation between the local Cartesian coordinate system at the crack tip and the physical space:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} cos(\phi) & sin(\phi) \\ -sin(\phi) & cos(\phi) \end{bmatrix} \begin{Bmatrix} X_1 - X_{ct} \\ Y_1 - Y_{ct} \end{Bmatrix} \tag{3.50}$$

where X_1, Y_1 are the physical coordinates (defined by Equation (3.41)), X_{ct}, Y_{ct} are the crack tip coordinates at the physical space and ϕ is the angle between the crack tip with respect to the horizontal line.

3.2.5 Extended Isogeometric Analysis - XIGA

By taking into account the pros of both XFEM and IGA, the Extended Isogeometric Analysis (XIGA) is proposed. This combined approach allows for the entire crack to be represented independently of the mesh. The isogeometric approximation is locally enriched to simulate discontinuities and singular fields. According to the location of any crack, a few degrees of freedom (DOF) are added to the selected control points of the original IGA model near the crack and contribute to the overall approximation through the use of enrichment functions.

The generalization of Equation (3.23) with the NURBS functions of the IGA and the enrichment of the XFEM is:

$$\mathbf{u^h}(\boldsymbol{\xi}) = \sum_{i=1}^{n_e} R_i(\boldsymbol{\xi}) \cdot \mathbf{u_i} + \sum_{j=1}^{n_{cf}} R_j(\boldsymbol{\xi}) \cdot H(\boldsymbol{\xi}) \cdot \mathbf{d_j} + \sum_{k=1}^{n_{ct}} R_k(\boldsymbol{\xi}) \cdot (\sum_{l=1}^{n_{ef}} f_l(\boldsymbol{\xi}) \cdot \mathbf{c_l}) \tag{3.51}$$

where $\boldsymbol{\xi} = (\xi_1, \xi_2)$ represents the parametric coordinates from the NURBS system, R_i is the basis of NURBS functions, n_e is the number of control points in the parametric system, n_{cf} is the number of enriched control points related to the crack face, n_{ct} is the number of enriched control points related to the crack tip, n_{ef} is the number of enrichment basis functions, $H(\xi)$ is the Heaviside function, f_k represents the crack tip enrichment functions, $\mathbf{d_j}$ and $\mathbf{C_k}$ are the additional DOF related to the crack face and crack tip, respectively.

The main steps for an XIGA method are as follows:

Crack Modeling

Given the framework of Fracture Mechanics and NURBS, the cracks are modeled as NURBS curves (Equation (3.36)) with no limitations in simple geometries (Figure 3.14 shows an example of a crack modeled as a NURBS curve). As the enrichment for both crack faces and crack tips takes place, this will involve the creation of routines, allowing the identifications of all parameters necessary for identifying a NURBS patch whose support contains a crack tip or crack face.

27
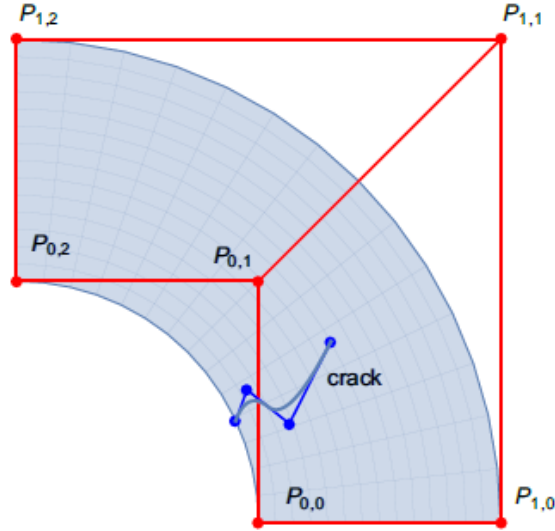
Figure 3.14: A crack as a NURBS curve.
Source: prepared by the author.

Selection of Enriched Control Points

For the selection of the enriched control points the same procedure used in [8] will take place:

In IGA it is know that there are the same number of basis functions as control points, therefore each basis function can be uniquely assigned to its corresponding control point. Further, each basis function has its own support and becomes zero outside it. Taking into account that, the follow criteria is applied:

a. Heaviside enrichment: The control points associated to the basis functions whose support contains a crack face will be enriched by the Heaviside functions.

b. Crack tip enrichment: The control points associated to the basis functions whose support contains a crack tip will be enriched by the crack tip functions.

It is necessary to clarify that the control points selected for both Heaviside and crack tip enrichment are only considered as crack tip enriched control points. Figure 3.15 exemplifies this. There are three highlighted control points on the Ω domain : P_j (red circle), P_k (blue circle) and P_l (green circle) with their respective basis functions support (dashed circles). There is also the crack with their respective crack tip (x_{ct}). Since the green dashed circle is crossed by the crack, the control point P_l is enriched by the Heaviside criteria. The blue dashed circle is crossed by the crack and also has the crack tip, so the control point P_k is enriched by the crack tip criteria. The red dashed circle is not crossed by the crack or contains a crack tip, hence the control point P_j is not enriched.
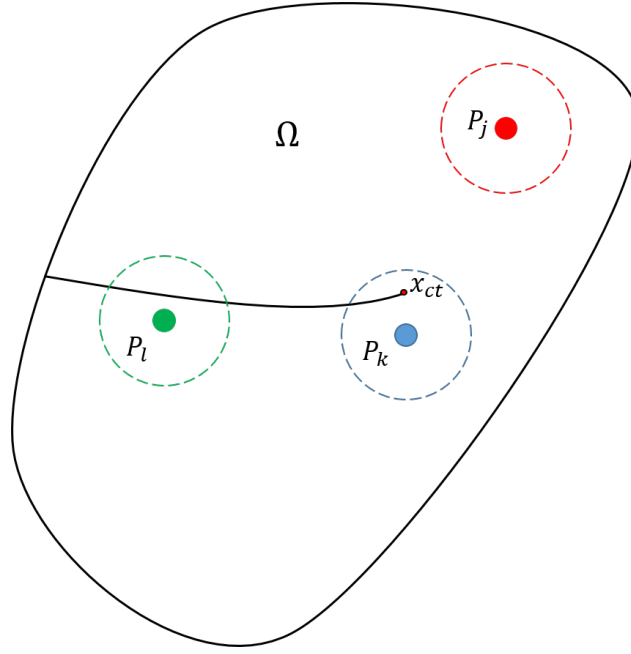
Figure 3.15: Selection of enriched Control points. P_j, P_k and P_l are three control points. The dashed circles are their basis support. The control point P_k is enriched by the crack tip criteria, the control point P_l is enriched by the Heaviside criteria and the control point P_j is not enriched. Source: prepared by the author.

Numerical Integration

Given that the weak form problem in the IGA formulation is being integrated, the Gauss quadrature is used for that procedure. In the case that the element contains a crack, it will be separated into cases of crack face integration and crack tip integration. In both cases the Gauss quadrature procedure is used.

Extended Isogeometric Formulation

New assembled matrix for both the global stiffness matrix $\boldsymbol{K}$ and global force vector $\boldsymbol{F}$ must be performed to include the new DOF incorporated by the enrichment. This is the same assembly process taken place in the XFEM method and further details can be found in [41].

The governing equation for the extended isogeometric analysis can be written as:

$$\mathbf{K_{enr}U_{enr}} = \mathbf{F_{enr}} \tag{3.52}$$

where $\mathbf{K_{enr}}$ is the enriched stiffness matrix, $\mathbf{F_{enr}}$ is the Force Vector and $\mathbf{U_{enr}}$ is the enriched displacement vector:

$$\mathbf{U_{enr}} = \left\{ \mathbf{U} \quad \mathbf{d} \quad \mathbf{c_1} \quad \mathbf{c_2} \quad \mathbf{c_3} \quad \mathbf{c_4} \right\}^T \tag{3.53}$$

where $\mathbf{U}$ is the vector of DOF related to the normal IGA, $\mathbf{d}$ is the vector of DOF related to Heaviside enrichment and $\mathbf{c_1}$, $\mathbf{c_2}$, $\mathbf{c_3}$ and $\mathbf{c_4}$ are the DOF related to the crack tip enrichment functions.

$\mathbf{K_{enr}}$ and $\mathbf{F_{enr}}$ are assembled from the element stiffness matrix as:

$$\mathbf{K_{enr}} = \begin{bmatrix} K^{uu} & K^{ud} & K^{uc_1} & K^{uc_2} & K^{uc_3} & K^{uc_4} \\ K^{du} & K^{dd} & K^{dc_1} & K^{dc_2} & K^{dc_3} & K^{dc_4} \\ K^{c_1 u} & K^{c_1 d} & K^{c_1 c_1} & K^{c_1 c_2} & K^{c_1 c_3} & K^{c_1 c_4} \\ K^{c_2 u} & K^{c_2 d} & K^{c_2 c_1} & K^{c_2 c_2} & K^{c_2 c_3} & K^{c_2 c_4} \\ K^{c_3 u} & K^{c_3 d} & K^{c_3 c_1} & K^{c_3 c_2} & K^{c_3 c_3} & K^{c_3 c_4} \\ K^{c_4 u} & K^{c_4 d} & K^{c_4 c_1} & K^{c_4 c_2} & K^{c_4 c_3} & K^{c_4 c_4} \end{bmatrix} \tag{3.54}$$

$$\mathbf{F_{enr}} = \left\{ \mathbf{F^u} \quad \mathbf{F^d} \quad \mathbf{F^{c_1}} \quad \mathbf{F^{c_2}} \quad \mathbf{F^{c_3}} \quad \mathbf{F^{c_4}} \right\}^T \tag{3.55}$$

where the size of each quadrant of $\mathbf{K_{enr}}$ and $\mathbf{F_{enr}}$ is given by the number of control points n_e, n_{c_f}, n_{ct} and the number of enrichment basis functions n_{ef}. Each component of the $\mathbf{K_{enr}}$ and $\mathbf{F_{enr}}$ can we written as:

$$\mathbf{K_{ij}^{rs}} = \int_{\Omega_e} (\mathbf{B_i^r})^T \mathbf{C} \mathbf{B_j^s} d\Omega \quad (r, s = u, d, c_1, c_2, c_3, c_4) \tag{3.56}$$

$$\mathbf{F_i^u} = \int_{\Gamma_{\bar{t}}} \mathbf{R_i^T} \bar{t} d\Gamma \tag{3.57}$$

$$\mathbf{F_i^d} = \int_{\Gamma_{\bar{t}}} \mathbf{R_i^T} H \bar{t} d\Gamma \tag{3.58}$$

$$\mathbf{F_i^{c_\alpha}} = \int_{\Gamma_{\bar{t}}} \mathbf{R_i^T} f_{c_\alpha} \bar{t} d\Gamma \quad (\alpha = 1, 2, 3, 4) \tag{3.59}$$

$$\mathbf{B_i^u} = \begin{bmatrix} \frac{\partial R_i}{\partial x} & 0 \\ 0 & \frac{\partial R_i}{\partial y} \\ \frac{\partial R_i}{\partial y} & \frac{\partial R_i}{\partial x} \end{bmatrix} \tag{3.60}$$

$$\mathbf{B_i^d} = \begin{bmatrix} \frac{\partial R_i}{\partial x} H & 0 \\ 0 & \frac{\partial R_i}{\partial y} H \\ \frac{\partial R_i}{\partial y} H & \frac{\partial R_i}{\partial x} H \end{bmatrix} \tag{3.61}$$

$$\mathbf{B_i^{c_\alpha}} = \begin{bmatrix} \frac{\partial R_i f_{c_\alpha}}{\partial x} & 0 \\ 0 & \frac{\partial R_i f_{c_\alpha}}{\partial y} \\ \frac{\partial R_i f_{c_\alpha}}{\partial y} & \frac{\partial R_i f_{c_\alpha}}{\partial x} \end{bmatrix} \quad (\alpha = 1, 2, 3, 4) \tag{3.62}$$

where H is the Heaviside function (Equation (3.47)) and f_{c_α} ($\alpha =$1,2,3,4) represents the crack tip enrichment function (Equation (3.48)).

The derivative of the basis functions with respect to the physical coordinates can be calculated as

$$\begin{bmatrix} \frac{\partial R_i}{\partial X_1} \\ \frac{\partial R_i}{\partial X_2} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \frac{\partial R_i}{\partial \xi_1} \\ \frac{\partial R_i}{\partial \xi_2} \end{bmatrix} \tag{3.63}$$

where the Jacobian matrix $\mathbf{J}$ for the transformation between physical and parametric spaces is defined as

$$\mathbf{J} = \begin{bmatrix} \frac{\partial X_1}{\partial \xi_1} & \frac{\partial X_2}{\partial \xi_1} \\ \frac{\partial X_1}{\partial \xi_2} & \frac{\partial X_2}{\partial \xi_2} \end{bmatrix} \tag{3.64}$$

and the NURBS derivatives are computed as in [6].

3.3 Displacements, Stresses and Strains

By solving the Equation (3.27) with the stiffness matrix and force vector obtained with the results of the previous section, the next step is to obtain the approximated displacements $\mathbf{u^h}$, stresses and strains. The displacement field $\mathbf{u^h}$ is obtained by computing the Equation (3.51) using the controls points obtained by the resolution of Equation (3.27). The strains are retrieved from $\mathbf{u^h}$ using the follow equation:

$$\boldsymbol{\varepsilon}(\boldsymbol{\xi}) = \boldsymbol{L}\boldsymbol{u}^h(\boldsymbol{\xi}) \tag{3.65}$$

where $\mathbf{L}$ is the differential operator defined as

$$\boldsymbol{L} = \begin{bmatrix} \frac{\partial}{\partial X_1} & 0 \\ 0 & \frac{\partial}{\partial X_2} \\ \frac{\partial}{\partial X_2} & \frac{\partial}{\partial X_1} \end{bmatrix} \tag{3.66}$$

The stresses are computed using the Hooke's law (Equation (3.16)). Finally, the von Mises stress is computed as:

$$\sigma_{vm} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 - \sigma_{xx}\sigma_{yy} + 3\sigma_{xy}^2} \tag{3.67}$$

3.4   Error Estimates

Three different types of measurement for relative error are calculated to assess the accuracy of the method: L^2 norm of the error, H^1 norm of the error and *Energy* norm of the error. The equations for each measurement are as follows:

$$\frac{\|e\|_{L^2}}{\|u\|_{L^2}} = \frac{\sqrt{\int_\Omega (\mathbf{u} - \mathbf{u^h})^T (\mathbf{u} - \mathbf{u^h}) \mathrm{d}\Omega}}{\sqrt{\int_\Omega \mathbf{u}^T \mathbf{u} \mathrm{d}\Omega}} \tag{3.68}$$

$$\frac{\|e\|_{H^1}}{\|u\|_{H^1}} = \frac{\sqrt{\int_\Omega \left[(\mathbf{u} - \mathbf{u^h})^T (\mathbf{u} - \mathbf{u^h}) + (\frac{\partial \mathbf{u}}{\partial x} - \frac{\partial \mathbf{u^h}}{\partial x})^T (\frac{\partial \mathbf{u}}{\partial x} - \frac{\partial \mathbf{u^h}}{\partial x}) + (\frac{\partial \mathbf{u}}{\partial y} - \frac{\partial \mathbf{u^h}}{\partial y})^T (\frac{\partial \mathbf{u}}{\partial y} - \frac{\partial \mathbf{u^h}}{\partial y}) \right] \mathrm{d}\Omega}}{\sqrt{\int_\Omega \left[\mathbf{u}^T \mathbf{u} + \frac{\partial \mathbf{u}}{\partial x}^T \frac{\partial \mathbf{u}}{\partial x} + \frac{\partial \mathbf{u}}{\partial y}^T \frac{\partial \mathbf{u}}{\partial y} \right] \mathrm{d}\Omega}} \tag{3.69}$$

$$\frac{\|e\|_E}{\|u\|_E} = \frac{\sqrt{\int_\Omega (\varepsilon - \varepsilon^{\mathbf{h}})^T (\sigma - \sigma^{\mathbf{h}}) \mathrm{d}\Omega}}{\sqrt{\int_\Omega \varepsilon^T \sigma \mathrm{d}\Omega}} \tag{3.70}$$

Chapter 4

Implementation

This chapter discusses the main topics of the implementation of enrichment procedures and post-process procedures for the Extended Isogeometric formulation.

4.1 Enrichment of the IGA Code (XIGA)

For the implementation of the enrichment method in Isogeometric Analysis, the follow functions are created:

4.1.1 Crack-Element Identification

A Crack-Element identification routine is implemented in order to select which control points are enriched by Heaviside or Crack tip criteria. Given that the crack is described as a NURBS curve (Equation (3.36)) and the body is described as a NURBS surface (Equation (3.41)) where each one has its own parametric space, the identification of the enriched Control points is done in the physical space by determining whether or not each element is cut by the crack. This process is explained in algorithm 1. This algorithm also saves the principal data that will be used during the integration and assembly process, such as the element's identification number, coordinates in the parametric space of the intersection between the crack and element, and NURBS functions which are cut by the crack. In this procedure the function *Crack cuts element* is used, which is explained in Algorithm 2.

In Algorithm 2, the function *crack intersects edge* determines if the crack collides with an edge of an element by doing a Newton-Raphson algorithm.

Algorithm 1: Crack-element identification algorithm for finding which elements are cut by the crack and store the principal information related to it.

---

1 Crack-Element identification ;

   Input   : Physical domain **X** and crack **C**

   Output: List of elements cuts by crack and principal parameters

2 **for** *each element* $\mathbf{X}_i$ *on* **X** **do**

3   **if** *Crack cuts elements* **then**

4   **if** *Crack Tip is inside the element* **then**

5   Save index number for crack tip enrichment;

6   Save parametric coordinates of the intersection between crack and element;

7   Save parametric coordinates of the crack tip inside the element;

8   Save shape functions support of the element;

9   **else**

10   Save index number for Heaviside enrichment;

11   Save parametric coordinates of the two intersections between the crack and the element;

12   Save shape functions support of the element;

13   **end**

14   **else**

15   **end**

16 **end**

17 **for** *each shape function* R_j *on Heaviside enrichment* **do**

18   **if** R_j *is on crack tip shape functions support list* **then**

19   Delete R_j from Heaviside Enrichment

20   **else**

21   **end**

22 **end**

23 return Index numbers for Heaviside and crack tip enrichment. Shape function list for Heaviside and crack tip enrichment.

Algorithm 2: Crack cuts element algorithm to determine whether an element is cut or not by crack.

---

1 Crack cuts element ;

   Input   : Physical domain $\mathbf{X}_i$ of an element and crack **C**

   Output: True if Crack cuts the element, otherwise false

2 **for** *each edge of the element* $\mathbf{X}_i$ **do**

3   **if** *crack intersects edge* **then**

4   return True;

5   **else**

6   **end**

7 **end**

8 return False;

4.1.2 Crack Mapping Process

Given that the crack $C(\xi_c)$ is defined as a NURBS curve and the body $X(\xi, \eta)$ is defined as a NURBS surface, each one with its own parametric space, it is necessary to choose a unique parametric space to work with. In this case, it is optimal to work in the parametric space of the body. In order to do that the crack is mapped from the physical space to the body parametric space (see Figure 4.1).
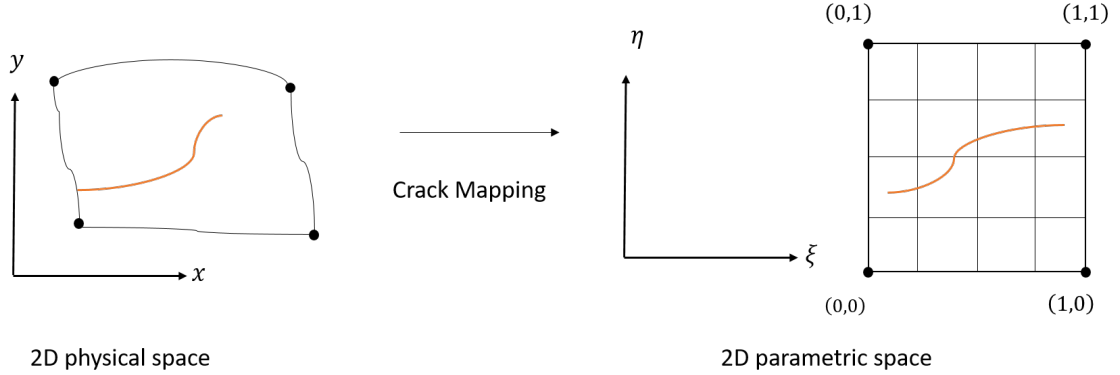
Figure 4.1: Scheme of crack mapping from the physical space to the field parametric space. Source: prepared by the author.

This is done by an inverse-map algorithm, which finds the point (ξ_p, η_p) in the parametric space such that $X(\xi_p, \eta_p) = C(\xi_c)$. Algorithm 3 explains the crack mapping process. This is done by splitting the crack's parametric space into n points ξ_{c_i} and then finding their value $C(\xi_{c_i}) = (x_i, y_i)$ on the physical space. In this Algorithm the *inverse map* function computes the value ξ, η such that $(x, y) = X(\xi, \eta)$. This function is implemented based on the algorithm proposed by Piegl in [51].

Algorithm 3: Crack mapping algorithm to determine the field's parametric coordinates of the crack.

---
1 Crack mapping ;
    Input : Physical domain $\mathbf{X}$ and crack $\mathbf{C}$
    Output: vectors ξ_p and η_p)
2 **for** $i = 0$; $i < n$; $i++$ **do**
3   compute ξ_{c_i} ;
4   compute $\mathbf{C}(\xi_{c_i})$;
5 $[\xi_{p_i}, \eta_{p_i}] = inverse\_map(\mathbf{C}(\xi_{c_i}))$;
6 **end**
7 return vectors ξ_p and η_p;

4.1.3 Crack-Element Integration

The procedure for numerical integration is performed by the Gaussian quadrature process. This is done in the parametric space of the field approximation. In the case of elements which are cut by the crack or contain a crack tip, the integral is split into several parts in order to integrate avoiding discontinuity. This process can be divided into two cases: elements cut by crack and elements containing the crack tip. Figure 4.2 outlines both cases:
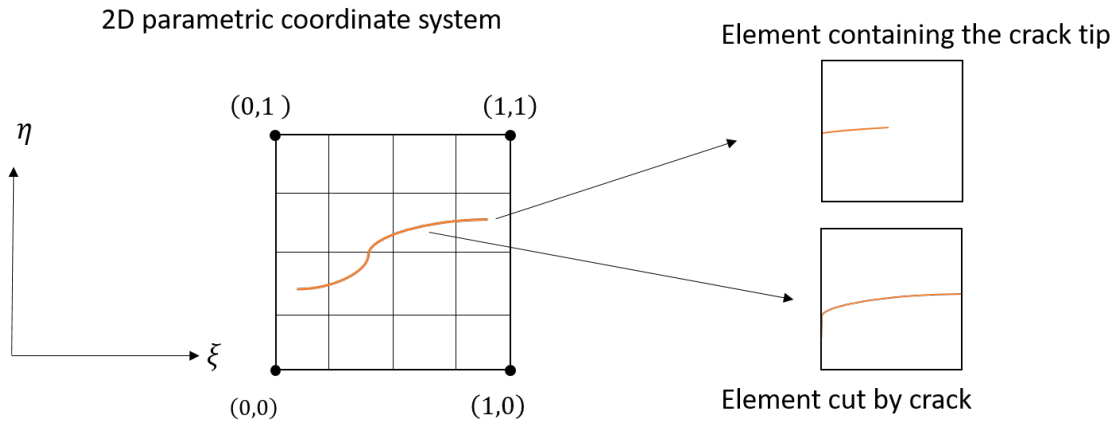
Figure 4.2: Different types of Crack-element Integration represented on the parametric space (crack represented by orange line). Source: prepared by the author.

Elements cut by crack

Given that elements are squares in the parametric space, the crack can only cross the element in the following cases (Figure 4.3):
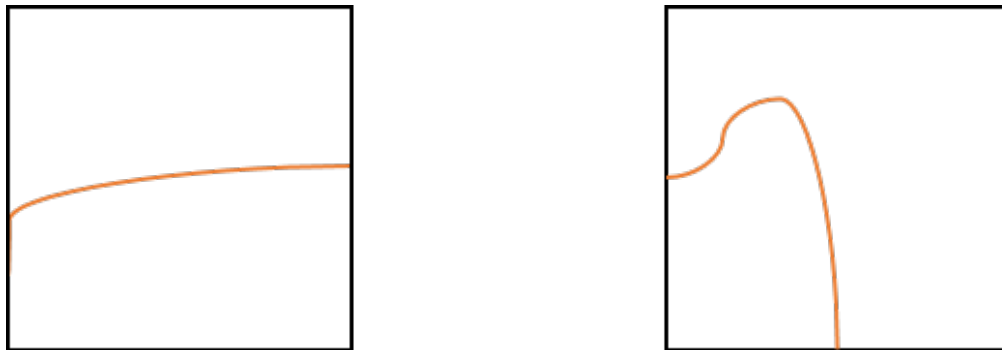
Figure 4.3: Two different examples of crack cutting a square element.
Source: prepared by the author.

Considering that, the algorithm required to accomplish the numerical integration consists of reparametrizing the split elements into square ones. It is done by the NURBS trimming algorithm proposed by Beers [11, 12]. The explanation of the mapping technique for elements cut by the crack is illustrated in Figure 4.4.
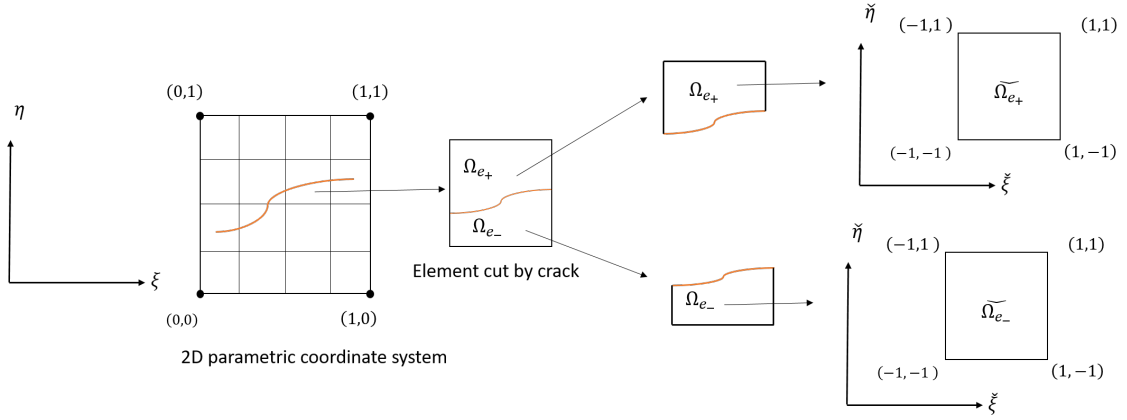
Figure 4.4: Mapping technique for elements cut by the crack.
Source: prepared by the author.

It starts with mapping the two trimming curves defining the upper and lower boundaries of the trimmed surface (Figure 4.5):
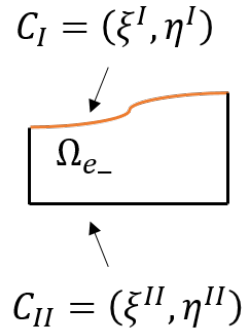
Figure 4.5: Detail on a trimmed element. C_I and C_{II} are the two trimming curves.
Source: prepared by the author.

$$C_I = (\xi^I, \eta^I) \tag{4.1}$$

$$C_{II} = (\xi^{II}, \eta^{II}) \tag{4.2}$$

where

$$\xi^I(\hat{\xi}) = \sum_{n=1}^{N_I} R_n^I(\hat{\xi}) \cdot \xi_n^I \tag{4.3}$$

$$\eta^I(\hat{\xi}) = \sum_{n=1}^{N_I} R_n^I(\hat{\xi}) \cdot \eta_n^I \tag{4.4}$$

$$\xi^{II}(\hat{\xi}) = \sum_{n=1}^{N_{II}} R_n^{II}(\hat{\xi}) \cdot \xi_n^{II} \tag{4.5}$$

37

$$\eta^{II}(\hat{\xi}) = \sum_{n=1}^{N_{II}} R_n^{II}(\hat{\xi}) \cdot \eta_n^{II} \tag{4.6}$$

where $R_n^I(\hat{\xi})$, $R_n^I(\hat{\eta})$ are one-dimensional NURBS basis functions defining the trimming curves, N^I, N^{II} are the number of control points and ξ^I,η^I and ξ^{II},η^{II} are the (ξ, η) coordinates of the control points. The superscript I and II refers to the bottom and top curve respectively.

Linear interpolation is used between the curves to map the trimmed area:

$$\xi(\hat{\xi}, \hat{\eta}) = N_1(\hat{\eta}) \cdot \xi^I(\hat{\xi}) + N_2(\hat{\eta}) \cdot \xi^{II}(\hat{\xi}) \tag{4.7}$$

$$\eta(\hat{\xi}, \hat{\eta}) = N_1(\hat{\eta}) \cdot \eta^I(\hat{\xi}) + N_2(\hat{\eta}) \cdot \eta^{II}(\hat{\xi}) \tag{4.8}$$

Finally, the map from the $(\hat{\xi}, \hat{\eta})$ system to the global (x, y) system is:

$$\mathbf{X} = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\hat{\xi}, \hat{\eta}) \mathbf{P_{ij}} \tag{4.9}$$

In our case, one of the trimming NURBS is the boundary of the element and the other one is the part of the crack that lies on the element. The algorithm implemented for the crack detection allow us to know the parameters of the crack that are involve in this process. Given that the crack is know in the physical space and has its own parametrization as a NURBS curve, it is necessary to perform an inverse mapping process from the physical space to the parametric space (which is explained in section 4.1.2). Also, given that Gauss quadrature is performed directly inside the trimmed element, the whole process is resumed in Algorithm 4.

Elements with the crack tip

In this case, the elements can be split into a sub-element with the crack and a sub-element without the crack, applying the elements cut by crack procedure to the sub-elements which include the crack. The other sub-element is integrated by the normal Gaussian quadrature procedure. The explanation of this mapping technique for elements with the crack tip is illustrated in Figure 4.6.

Algorithm 4: Gauss quadrature performed over a trimmed element.
---
1 Trimmed Element Gauss Quadrature ;

   Input : function f, Element cut by the crack, crack beginning and end in the
           element, Gauss points and weights.

   Output: Numerical integration S over a trimmed element

2 for *each Gauss point in ξ direction* **do**

3   **for** *each Gauss point in η direction* **do**

4   Calculate ξ_{gp} and u_{gp};

5   Calculate η_{gp} by performing a linear interpolation between the element
           boundary and the *inverse_map*$(\boldsymbol{C}(\xi_{u_{gp}}))$;

6   Perform the integration $S+=w_{gp} \times f(\xi_{gp}, \eta_{u_{gp}})$

7   **end**

8 end

9 return S;
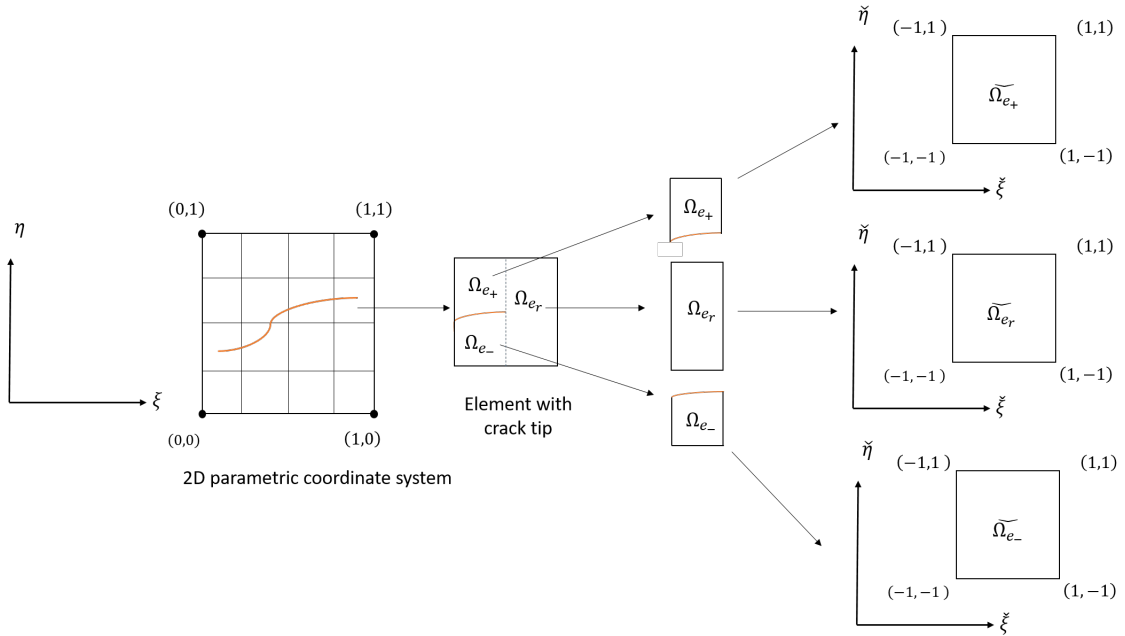
Figure 4.6: Mapping technique for elements with the crack tip.
Source: prepared by the author.

4.1.4 Heaviside Enrichment

The Heaviside enrichment is implemented using the formulation for XIGA. This process
is done by implementing the assembly of the stiffness matrix and force vector. Algorithm 5
explains the assembly process of the Heaviside enriched stiffness matrix. Algorithm 6 explains
the assembly process of the Heaviside enriched force vector:

Algorithm 5: Stiffness matrix algorithm assembly with Heaviside enrichment.

---
1 Stiffness Heaviside Enrichment Matrix ;
    Input   : List of elements cut by the crack and their principal parameters, Physical
                    domain $\mathbf{X}$, crack $\mathbf{C}$ and Stiffness Matrix $\mathbf{K}$
    Output: Stiffness Heaviside Enrichment Matrix $\mathbf{K}$
2 **for** *each control point* $\mathbf{U}_i$ **do**
3   **for** *each control point* $\mathbf{U_j}$ **do**
4   Calculate $\mathbf{K_{ij}^{uu}}$ from Equation (3.56);
5   Assembly $\mathbf{K_{ij}^{uu}}$ to The global stiffness matrix $\mathbf{K}$;
6   **if** *we are in a Heaviside Enrichment control point* **then**
7   Calculate $\mathbf{K_{ij}^{ud}}$,$\mathbf{K_{ij}^{du}}$ and $\mathbf{K_{ij}^{dd}}$ from Equation (3.56);
8   Assembly to The global stiffness matrix $\mathbf{K}$
9   **else**
10   **end**
11   **end**
12 **end**
13 return $\mathbf{K}$;

Algorithm 6: Force vector algorithm assembly with Heaviside enrichment.

---
1 Force Heaviside Enrichment Vector ;
    Input   : List of elements cut by the crack and their principal parameters, Physical
                    domain $\mathbf{X}$, crack $\mathbf{C}$ and Force Vector F
    Output: Force Heaviside Enrichment Vector $\mathbf{K}$
2 **for** *each control point* $\mathbf{U}_i$ **do**
3   Calculate $\mathbf{F_i^u}$ from Equation (3.57);
4   Assembly $\mathbf{F_i^u}$ to The global Force vector $\mathbf{F}$;
5   **if** *we are in a Heaviside Enrichment control point* **then**
6   Calculate $\mathbf{F_i^d}$ from Equation (3.58);
7   Assembly to The global Force vector $\mathbf{F}$
8   **else**
9   **end**
10 **end**
11 return $\mathbf{F}$;

4.1.5   Heaviside and Crack Tip Enrichment

Both Heaviside and Crack Tip enrichment are implemented using the formulation for XIGA. This process is done by implementing a new assembly of the stiffness matrix and force vector. Algorithm 7 explains the assembly process for the stiffness matrix. Algorithm 8 explains the assembly process for the enriched force vector.

Algorithm 7: Stiffness matrix algorithm assembly with Heaviside and Crack Tip enrichment.

---

1 Stiffness Enrichment Matrix ;

    Input : List of elements cut by the crack and their principal parameters, Physical domain $\mathbf{X}$, crack $\mathbf{C}$ and Stiffness Matrix $\mathbf{K}$

    Output: Stiffness Enrichment Matrix $\mathbf{K}$

2 **for** *each control point* $\mathbf{U}_i$ **do**

3   **for** *each control point* $\mathbf{U_j}$ **do**

4   Calculate $\mathbf{K_{ij}^{uu}}$ from Equation (3.56);

5   Assembly $\mathbf{K_{ij}^{uu}}$ to The global stiffness matrix $\mathbf{K}$;

6   **if** *we are in a Heaviside Enrichment control point* **then**

7   Calculate $\mathbf{K_{ij}^{ud}}$, $\mathbf{K_{ij}^{du}}$, $\mathbf{K_{ij}^{dd}}$, $\mathbf{K_{ij}^{c_m d}}$ and $\mathbf{K_{ij}^{dc_m}}$ (m from 1 to 4) from Equation (3.56);

8   Assembly to The global stiffness matrix $\mathbf{K}$.

9   **else if** *we are in Crack Tip Enrichment control point* **then**

10   Calculate $\mathbf{K_{ij}^{c_m u}}$, $\mathbf{K_{ij}^{uc_m}}$, $\mathbf{K_{ij}^{c_m d}}$, $\mathbf{K_{ij}^{dc_m}}$, $\mathbf{K_{ij}^{c_l c_m}}$ and $\mathbf{K_{ij}^{c_m c_l}}$ (m and l from 1 to 4) from Equation (3.56);

11   Assembly to The global stiffness matrix $\mathbf{K}$.

12   **end**

13 **end**

14 return $\mathbf{K}$;

Algorithm 8: Force vector algorithm assembly with Heaviside and Crack Tip enrichment.

---

1 Force Enrichment Vector ;

    Input : List of elements cut by the crack and their principal parameters, Physical domain $\mathbf{X}$, crack $\mathbf{C}$ and Force Vector $\mathbf{F}$

    Output: Force Heaviside Enrichment Vector $\mathbf{K}$

2 **for** *each control point* $\mathbf{U}_i$ **do**

3   Calculate $\mathbf{F_i^u}$ from Equation (3.57);

4   Assembly $\mathbf{F_i^u}$ to The global Force vector $\mathbf{F}$;

5   **if** *we are in a Heaviside Enrichment control point* **then**

6   Calculate $\mathbf{F_i^d}$ from Equation (3.58);

7   Assembly to The global Force vector $\mathbf{F}$

8   **else if** *we are in a Crack Tip Enrichment control point* **then**

9   Calculate $\mathbf{F_i^{c_m}}$ (m from 1 to 4) from Equation (3.59);

10   Assembly to The global Force vector $\mathbf{F}$

11 **end**

12 return $\mathbf{F}$;

4.2 Post-Processing

4.2.1 J-Integral

For the numerical computation of the J-Integral, the domain method proposed by Li et al. [44] is implemented. In the isogeometric approach, a smooth q-function can be defined by the NURBS basis function as

$$q(\hat{\xi}, \hat{\eta}) = \sum_{i}^{n} \sum_{j}^{m} R_{i,j}^{p,q}(\hat{\xi}, \hat{\eta})\boldsymbol{Q_{ij}} \tag{4.10}$$

where $\boldsymbol{Q_{ij}}$ are the corresponding 3D control points. Figure 4.7 shows an example of a q function. It can be seen that $q(0.5, 0.5) = 1$ and $q = 0$ in all the boundaries. Table 4.1 shows the control points and weights employed to generate the q function.
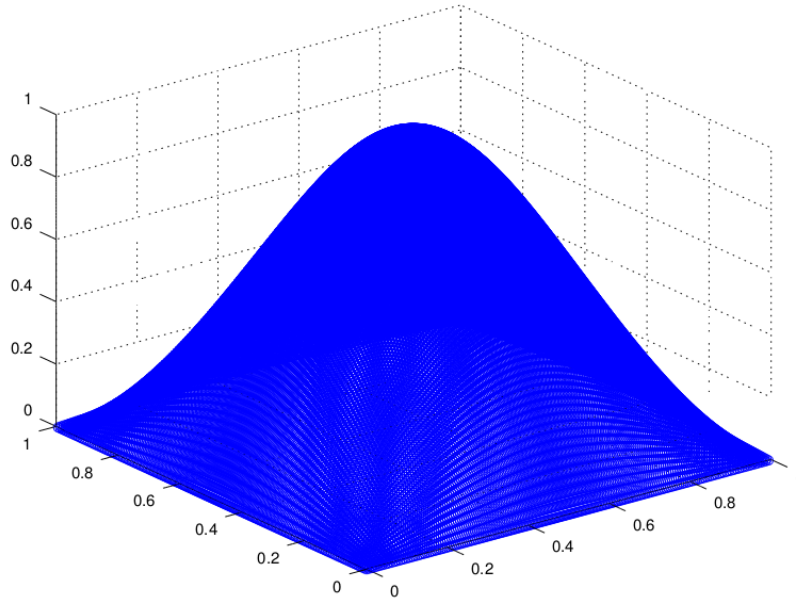
Figure 4.7: q-function employed for the computation of J-integral. Source: prepared by the author.

The derivatives of q with respect to the physical coordinates $\boldsymbol{X}$ can be computed using the chain rule:

$$\frac{\partial q}{\partial x} = \left(\frac{\partial q}{\partial \hat{\xi}}\frac{\partial \hat{\xi}}{\partial \xi} + \frac{\partial q}{\partial \hat{\eta}}\frac{\partial \hat{\eta}}{\partial \xi}\right) \frac{\partial \xi}{\partial x} + \left(\frac{\partial q}{\partial \hat{\xi}}\frac{\partial \hat{\xi}}{\partial \eta} + \frac{\partial q}{\partial \hat{\eta}}\frac{\partial \hat{\eta}}{\partial \eta}\right) \frac{\partial \eta}{\partial x} \tag{4.11}$$

$$\frac{\partial q}{\partial y} = \left(\frac{\partial q}{\partial \hat{\xi}}\frac{\partial \hat{\xi}}{\partial \xi} + \frac{\partial q}{\partial \hat{\eta}}\frac{\partial \hat{\eta}}{\partial \xi}\right)\frac{\partial \xi}{\partial y} + \left(\frac{\partial q}{\partial \hat{\xi}}\frac{\partial \hat{\xi}}{\partial \eta} + \frac{\partial q}{\partial \hat{\eta}}\frac{\partial \hat{\eta}}{\partial \eta}\right)\frac{\partial \eta}{\partial y} \tag{4.12}$$

where $\frac{\partial \hat{\xi}}{\partial \xi}$, $\frac{\partial \hat{\eta}}{\partial \xi}$, $\frac{\partial \hat{\xi}}{\partial \eta}$, $\frac{\partial \hat{\eta}}{\partial \eta}$ are the derivatives between the parametric space and the q-space. $\frac{\partial \xi}{\partial x}$, $\frac{\partial \eta}{\partial x}$, $\frac{\partial \xi}{\partial y}$, $\frac{\partial \eta}{\partial y}$ are the derivatives between the physical and the parametric space (Equation (3.63)).

Table 4.1: Control Points and weights for q function

Control Point $\mathbf{X_{ij}}$	weight w_{ij}	control Point $\mathbf{Q_{ij}}$	weight w_{ij}
(0,0,0)	1.0	(0,0.66,0)	1.0
(0.33,0,0)	1.0	(0.33,0.66,1)	1.5
(0.66,0,0)	1.0	(0.66,0.66,1)	1.5
(1,0,0)	1.0	(1,0.66,0)	1.0
(0,0.33,0)	1.0	(0,1,0)	1.0
(0.33,0.33,1)	1.5	(0.33,1,0)	1.0
(0.66,0.33,1)	1.5	(0.66,1,0)	1.0
(1,0.33,0)	1.0	(1,1,0)	1.0

Source: prepared by the author.

The expression for the J-integral is computed using the domain integral method (Equation (3.9)). The domain for the integral is taken as the complete physical space and is computed by the Gaussian quadrature:

$$J = \sum_{\substack{\text{elements} \\ \text{in } \Omega}} \sum_{p=1}^{N_{gp}} \left[\left[\sigma_{ij}\frac{\partial u_i}{\partial x} - W\delta_{1i}\right]\frac{\partial q_1}{\partial x_j}\det\left(\frac{\partial x_k}{\hat{\xi}}\right)\right]_p w_p \tag{4.13}$$

where the expressions inside $[\ ]_p$ are evaluated at the Gauss points, and w_p are the respective Gauss weights.

4.2.2 Error Estimates

L^2, H^1 and Energy norms of the error are computed using the post-process results obtained by Equations (3.51),(3.65) and (3.66). The integrals are solved by integrating over each element using the Gaussian quadrature:

$$\frac{\|e\|_{L^2}}{\|u\|_{L^2}} = \frac{\sqrt{\sum\limits_{\substack{\text{elements} \\ \text{in } \Omega}} \sum\limits_{p=1}^{N_{gp}} [(\mathbf{u} - \mathbf{u^h})^T(\mathbf{u} - \mathbf{u^h})\det(J)]_p\, w_p}}{\sqrt{\sum\limits_{\substack{\text{elements} \\ \text{in } \Omega}} \sum\limits_{p=1}^{N_{gp}} [(\mathbf{u}^T\mathbf{u})\det(J)]_p\, w_p}} \tag{4.14}$$

$$\frac{\|e\|_{H^1}}{\|u\|_{H^1}} = \frac{\sqrt{\sum\limits_{\substack{elem. \\ in \ \Omega}} \sum\limits_{p=1}^{N_{gp}} \left[\left((\mathbf{u}-\mathbf{u^h})^T(\mathbf{u}-\mathbf{u^h}) + (\frac{\partial \mathbf{u}}{\partial x} - \frac{\partial \mathbf{u^h}}{\partial x})^T(\frac{\partial \mathbf{u}}{\partial x} - \frac{\partial \mathbf{u^h}}{\partial x}) + (\frac{\partial \mathbf{u}}{\partial y} - \frac{\partial \mathbf{u^h}}{\partial y})^T(\frac{\partial \mathbf{u}}{\partial y} - \frac{\partial \mathbf{u^h}}{\partial y}) \right) det(J) \right]_p w_p}}{\sqrt{\sum\limits_{\substack{elem. \\ in \ \Omega}} \sum\limits_{p=1}^{N_{gp}} \left[\left(\mathbf{u}^T\mathbf{u} + \frac{\partial \mathbf{u}}{\partial x}^T \frac{\partial \mathbf{u}}{\partial x} + \frac{\partial \mathbf{u}}{\partial y}^T \frac{\partial \mathbf{u}}{\partial y} \right) det(J) \right]_p w_p}} \qquad (4.15)$$

$$\frac{\|e\|_{E}}{\|u\|_{E}} = \frac{\sqrt{\sum\limits_{\substack{elem. \\ in \ \Omega}} \sum\limits_{p=1}^{N_{gp}} \left[(\varepsilon - \varepsilon^{\mathbf{h}})^T(\sigma - \sigma^{\mathbf{h}}) det(J) \right]_p w_p}}{\sqrt{\sum\limits_{\substack{elem. \\ in \ \Omega}} \sum\limits_{p=1}^{N_{gp}} \left[(\varepsilon^T \sigma) \ det(J) \right]_p w_p}} \qquad (4.16)$$

where the expressions inside $[\ ]_p$ are evaluated at the Gauss points, w_p are the respective Gauss weights and $det(J)$ is the determinant of the Jacobian transformation for the integral.

Chapter 5

Results

The XIGA routine is tested with two numerical simulations. The first numerical simulation consists in the problem of an infinite plate with a crack under tensile loading. It is tested with a partial enriched IGA routine (only Heaviside enrichment) and a XIGA (Heaviside + crack tip enrichment) routine. L^2, H^1 and Energy norms of the error are computed to test the accuracy of the method. K_I is computed by the J-integral to validate the results and u_x, u_y, ε_x, ε_y, ε_{xy} and σ_{vm} contour plots are generated for both the numerical and the analytical solution.

The second numerical simulation consists in the problem of a plate with a circular hole containing a straight crack under tensile loading. It is tested with a XIGA routine using quadratic and cubic order NURBS. L^2, H^1 and Energy norms of the error are also computed to test the accuracy of the method. K_I is computed by the J-integral to validate the results and u_x, u_y, ε_x, ε_y, ε_{xy} and σ_{vm} contour plots are generated for both the numerical and the analytical solution.

5.1 Infinite Plate with a Crack Under Uniform Tensile Loading

An infinite plate containing a regular straight crack of length $2a$ under uniform tensile loading σ_o is considered. The plate is in a plane strain state. The closed area ABCD which includes the c_l part of the crack, as depicted in Figure 5.1, is modeled. The analytical solution for the displacement and stress fields in terms of local polar coordinates from the crack tip are:

$$u_x(r,\theta) = \frac{2(1+\nu)}{\sqrt{2\pi}}\frac{K_I}{E}\sqrt{r}cos\frac{\theta}{2}(2-2\nu-cos^2\frac{\theta}{2}) \qquad (5.1)$$

$$u_y(r,\theta) = \frac{2(1+\nu)}{\sqrt{2\pi}}\frac{K_I}{E}\sqrt{r}sin\frac{\theta}{2}(2-2\nu-cos^2\frac{\theta}{2}) \qquad (5.2)$$

$$\sigma_{xx}(r,\theta) = \frac{K_I}{\sqrt{2\pi r}}cos\frac{\theta}{2}(1 - sin\frac{\theta}{2}sin\frac{3\theta}{2}) \tag{5.3}$$

$$\sigma_{yy}(r,\theta) = \frac{K_I}{\sqrt{2\pi r}}cos\frac{\theta}{2}(1 + sin\frac{\theta}{2}sin\frac{3\theta}{2}) \tag{5.4}$$

$$\sigma_{xy}(r,\theta) = \frac{K_I}{\sqrt{2\pi r}}sin\frac{\theta}{2}cos\frac{\theta}{2}cos\frac{3\theta}{2} \tag{5.5}$$
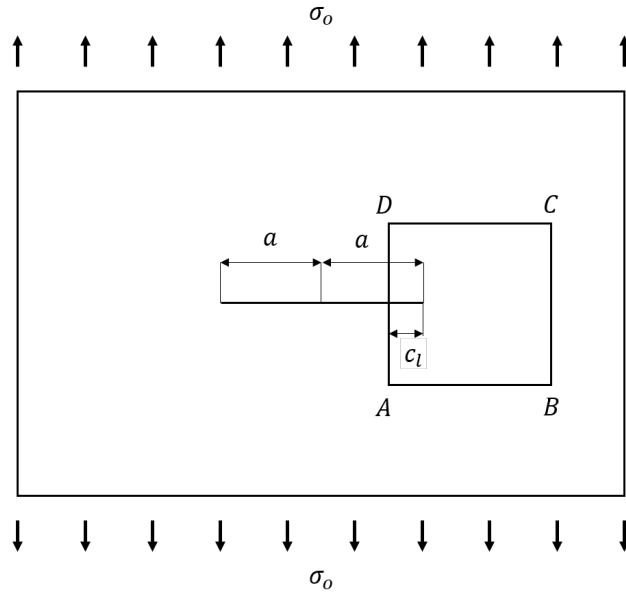

Figure 5.1: Geometry and loading in a center crack plate under remote tension.
Source: adapted from [9].

The boundary conditions for the top, right and bottom edges are essential boundaries, while the left edge is considered as a natural boundary. The physical parameters are presented in Table 5.1. The control points and weights used for the parametrization of the geometry and the crack are presented in Table 5.2 and Table 5.3 respectively. The Knot vectors used for the parametrization of the geometry and the crack are presented in Table 5.4 and Table 5.5 respectively. This problem is analyzed by Heaviside enriched IGA and Heaviside + Crack tip enriched IGA to see the influence of the enrichment at the crack tip. Linear NURBS are used for all the simulations:

Four different types of discretization has been considered to measure the accuracy and convergence rate of this numerical simulation. This is accomplish by a h-refinement method, which is a standard NURBS refinement procedure. The details can be found at [48]. Table 5.6 resumes the principal information for the discretization. Figure 5.2 shows the four meshes. 2×2 Gaussian quadrature are used for the integration in each element.

Table 5.1: Physical Parameters for Infinite Plate with a Crack Under Uniform Tensile Loading

Physical Parameters	
Young's Modulus E	10
Poisson Ratio's ν	0.3
Mode 1 Stress Intensity Factor K_I	1
geometry size $AB = BC = CD = DA$	2
Crack length C_l	0.5

Source: prepared by the author.

Table 5.2: Control Points and weights for Physical Domain - Numerical Simulation 1

Control Point $\mathbf{X_{ij}}$	weight w_{ij}
(-1,-1)	1.0
(-1,1)	1.0
(1,-1)	1.0
(1,1)	1.0

Source: prepared by the author.

Table 5.3: Control Points and weights for crack - Numerical Simulation 1

Control Point $\mathbf{X_{ij}}$	weight w_{ij}
(-1,0)	1.0
(0,0)	1.0

Source: prepared by the author.

Table 5.4: Knot vector for Physical Domain - Numerical Simulation 1

Direction	Knot vector
ξ	$\Xi_1 = \{0, 0, 1, 1\}$
η	$\Xi_2 = \{0, 0, 1, 1\}$

Source: prepared by the author.

Table 5.5: Knot vector for crack - Numerical Simulation 1

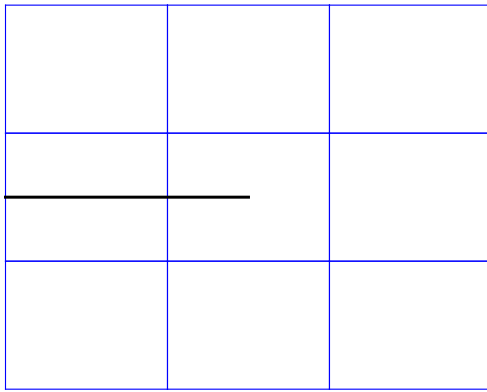| Direction	Knot vector
$\xi_{\mathbf{c}}$	$\Xi_{1_c} = \{0, 0, 1, 1\}$

Source: prepared by the author.

Table 5.6: Maximum edge size and number of elements for each mesh - Numerical Simulation 1

Mesh	Maximum edge size h	Number of Elements
1	0.333	3×3
2	0.111	9×9
3	0.037	27×27
4	0.012	81×81

Source: prepared by the author.

(a) Mesh 1
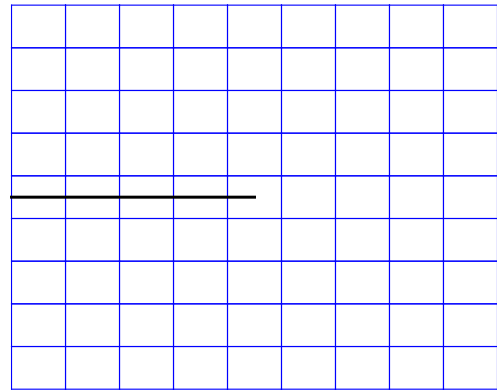
(b) Mesh 2

(c) Mesh 3

(d) Mesh 4

Figure 5.2: Infinite plate with a crack under uniform tensile loading: Meshes produced by uniform h-refinement. The black line represents the crack. Source: prepared by the author.

Figure 5.3 shows the selection of enriched Control points by the Heaviside and Crack tip criteria for two different meshes.
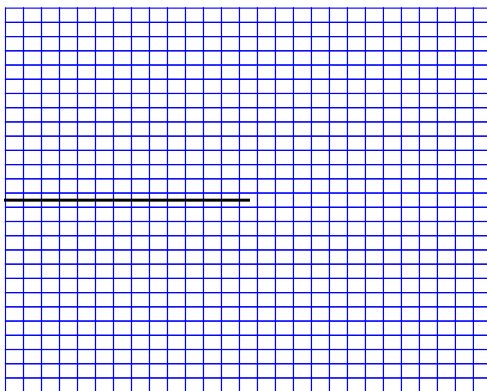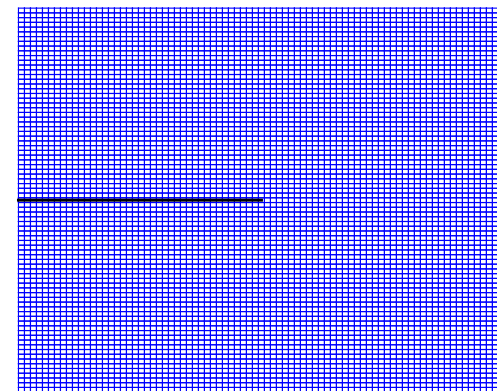
Figures 5.4, 5.5 and 5.6 show the log-log L^2, H^1 and Energy norms of the error respectively, obtained for Heaviside enrichment and Heaviside + Crack tip enrichment.

Figure 5.7 shows the Stress Intensity Factor K_1 obtained for Heaviside enrichment and Heaviside + Crack tip enrichment computed by J-integral using the domain method.

Figures 5.8 and 5.9 show the contour plot for the u_x displacement obtained with XIGA and the exact solution respectively. Figures 5.10 and 5.11 show the contour plot for the u_y displacement obtained with XIGA and the exact solution, respectively.

Figures 5.12 and 5.13 show the contour plot for the ε_x strain obtained with XIGA and the exact solution respectively. Figures 5.14 and 5.15 show the contour plot for the ε_y strain obtained with XIGA and the exact solution respectively. Figures 5.16 and 5.17 show the contour plot for the ε_{xy} strain obtained with XIGA and the exact solution, respectively.

Figure 5.18 shows the contour plot for the equivalent von Mises stress obtained with XIGA and Figure 5.19 shows the contour plot for the equivalent von Mises stress obtained with the exact solution.

In all the contour plots a mesh of 81×81 elements is used.

(a) Mesh 2   (b) Mesh 3
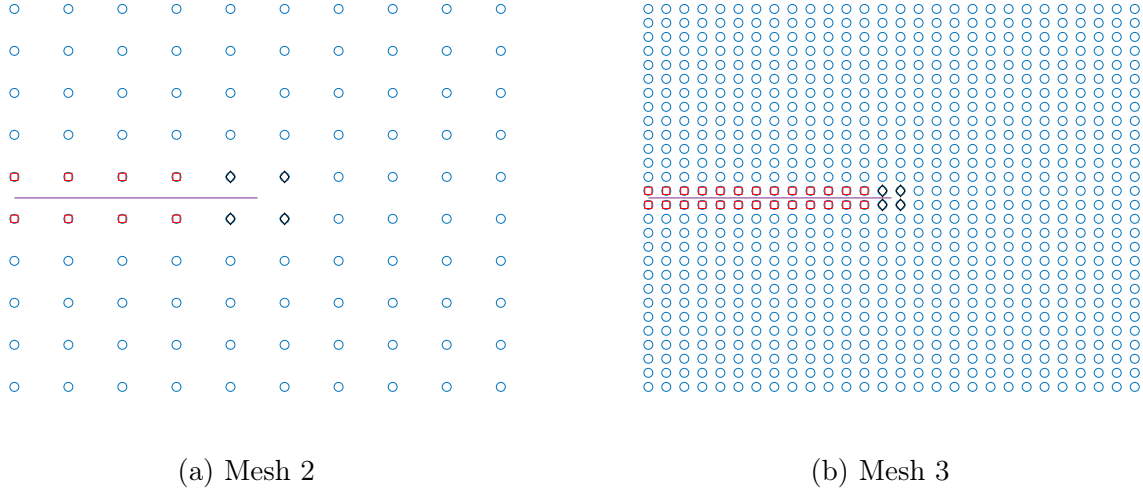
Figure 5.3: Control Point selection for enrichment for mesh 2 (left) and mesh 3 (right) : Control Points marked by red squares are enriched by Heaviside function and the black ones are enriched by the Crack tip functions. The black line represents the crack. Source: prepared by the author.

Figure 5.4: Heaviside and Heaviside + Crack Tip enriched IGA. Convergence in the L^2 norm upon decreasing the element size. Source: prepared by the author.
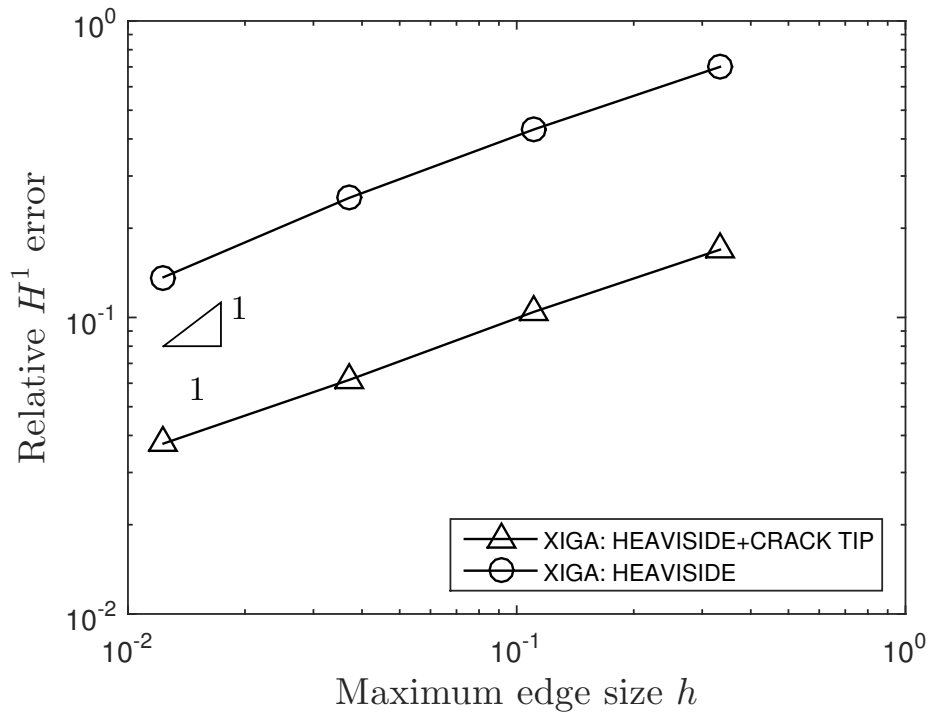
Figure 5.5: Heaviside and Heaviside + Crack Tip enriched IGA. Convergence in the H^1 norm upon decreasing the element size. Source: prepared by the author.
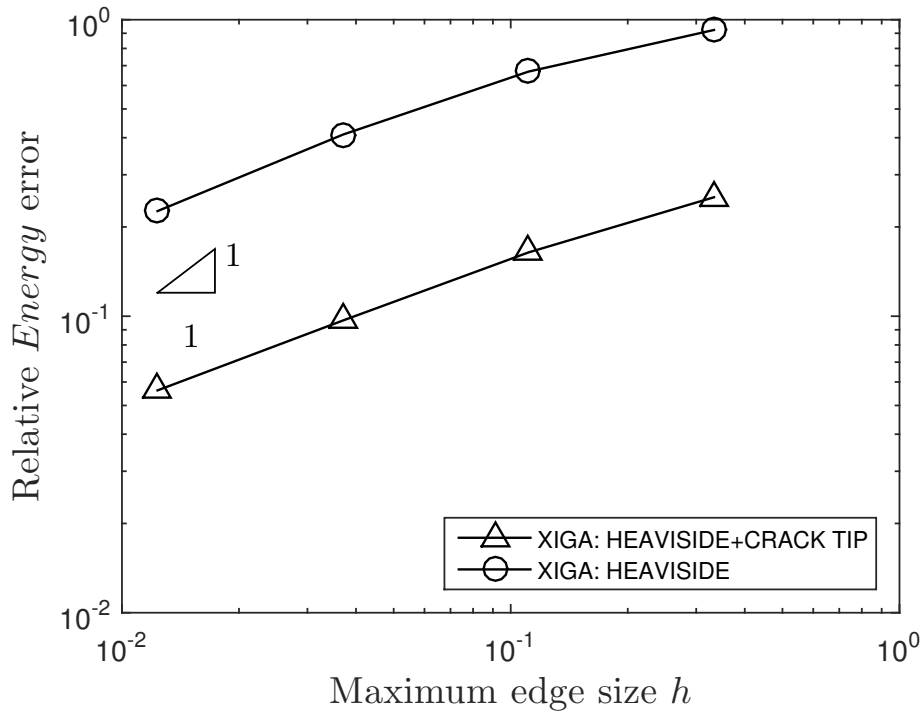
Figure 5.6: Heaviside and Heaviside + Crack Tip enriched IGA. Convergence in the Energy norm upon decreasing the element size. Source: prepared by the author.
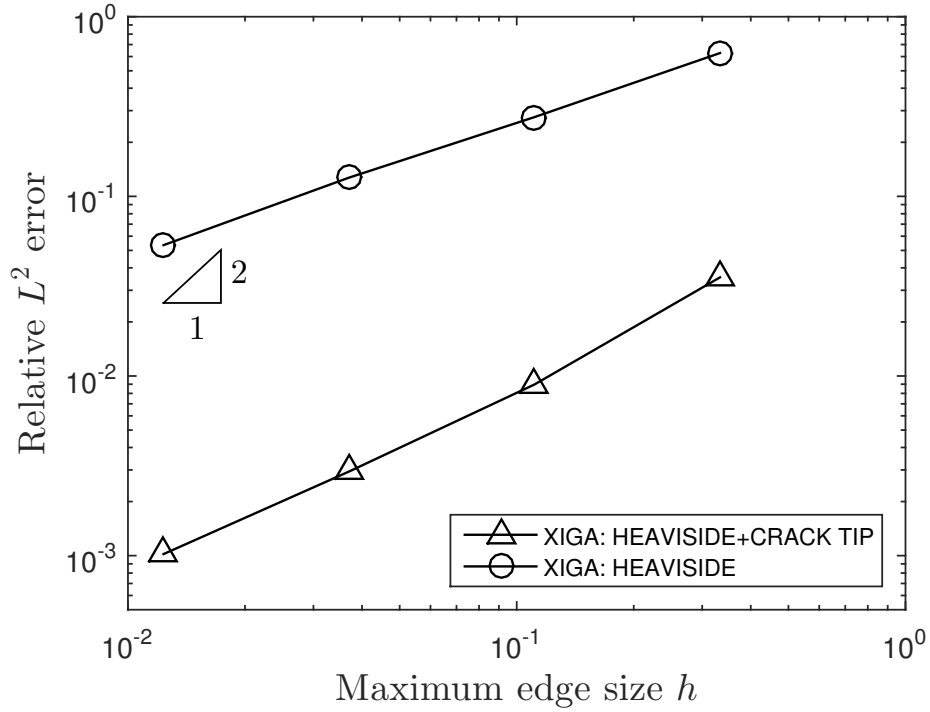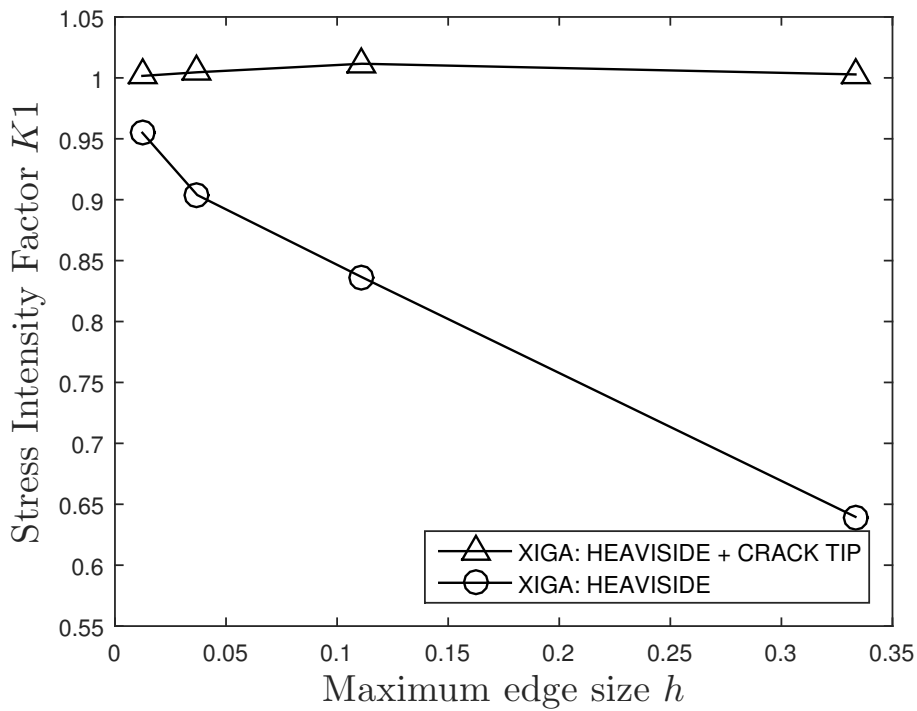

Figure 5.7: Heaviside and Crack Tip enriched IGA. Convergence of the Stress Intensity Factor K_1 upon decreasing the element size. Source: prepared by the author.
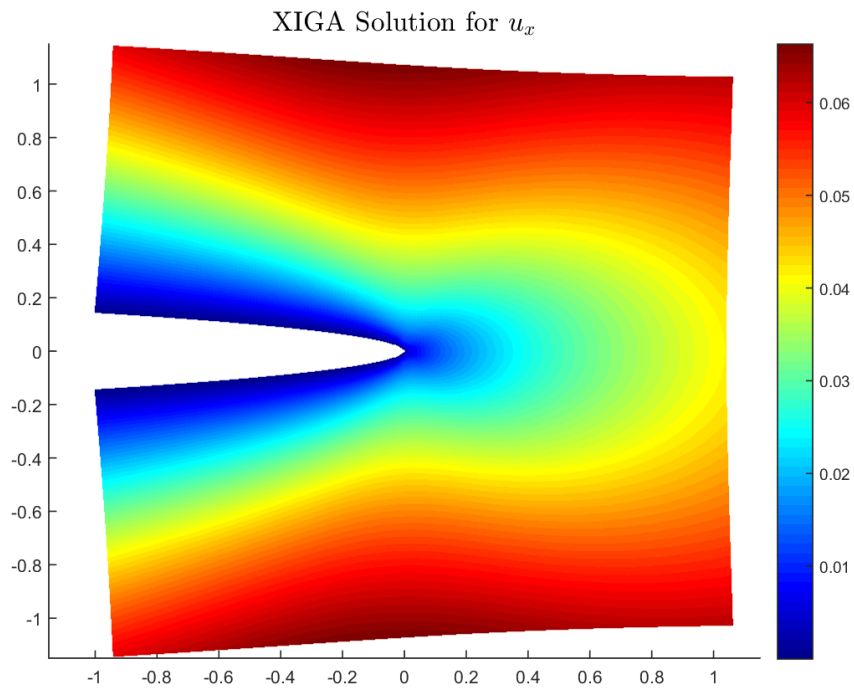
Figure 5.8: u_x displacement contour plot generated by XIGA. Source: prepared by the author.
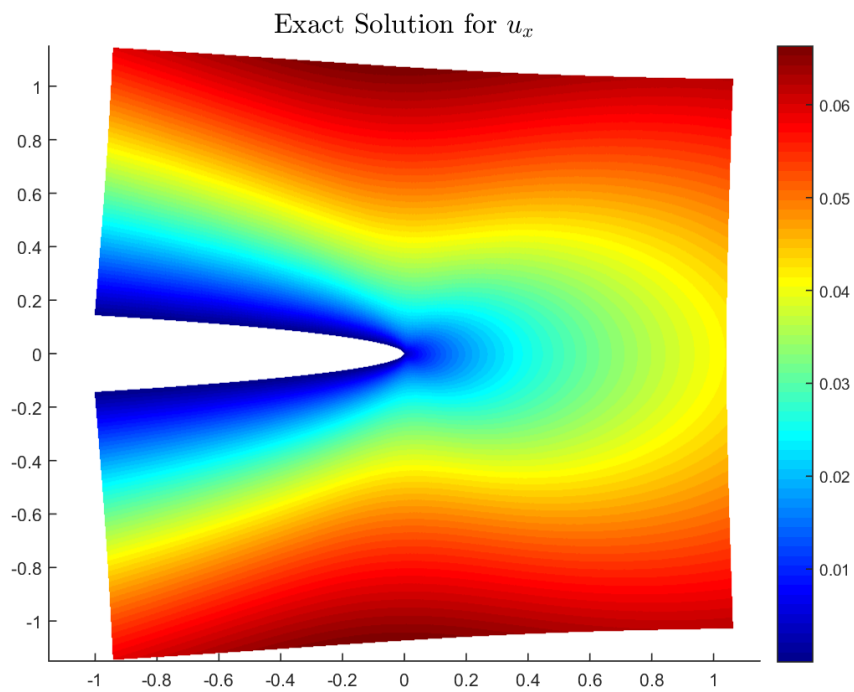
Figure 5.9: u_x displacement contour plot generated by exact solution. Source: prepared by the author.
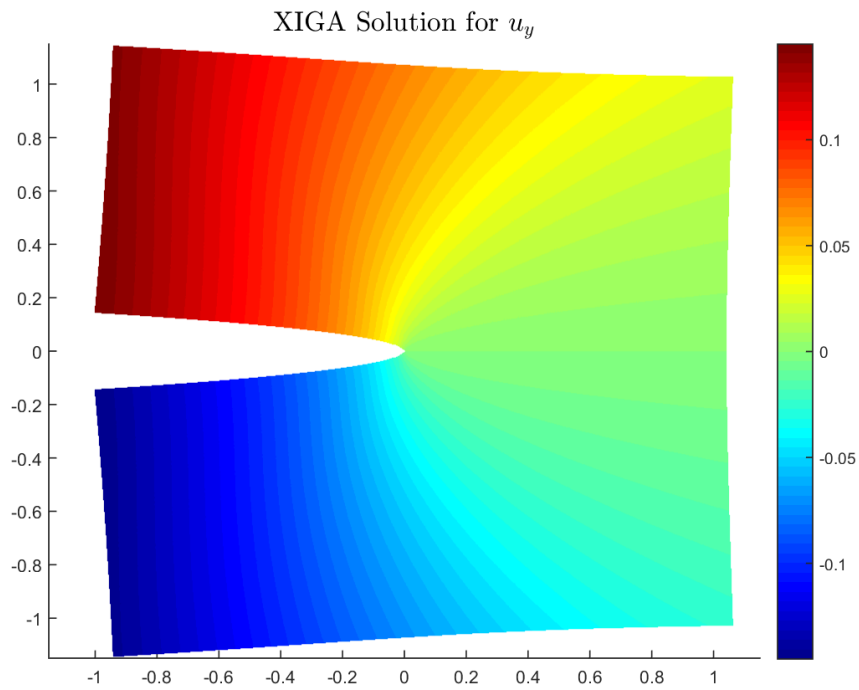
Figure 5.10: u_y displacement contour plot generated by XIGA. Source: prepared by the author.
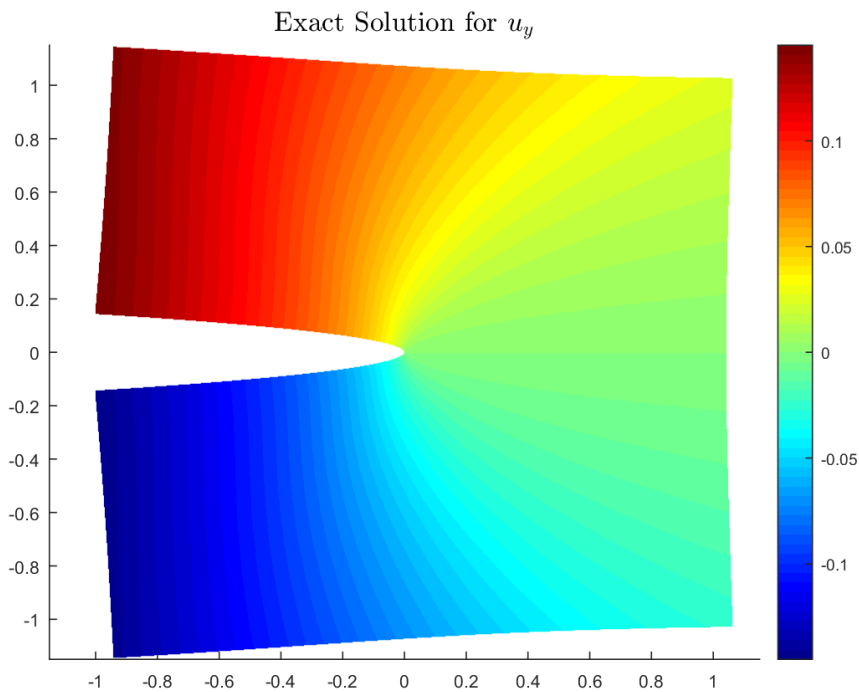
Figure 5.11: u_y displacement contour plot generated by exact solution. Source: prepared by the author.
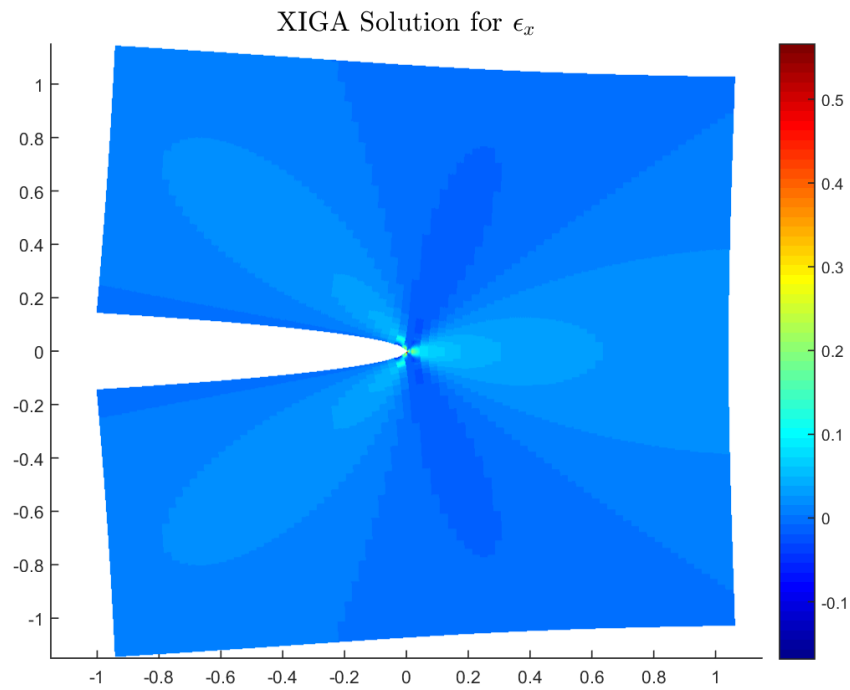
Figure 5.12: ε_x strain contour plot generated by XIGA. Source: prepared by the author.
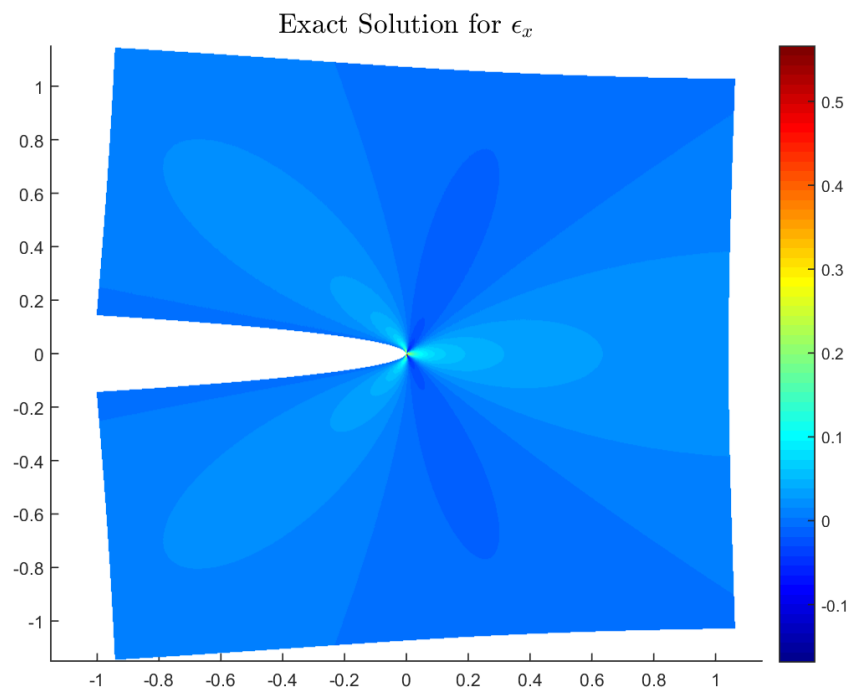
Figure 5.13: ε_x strain contour plot generated by exact solution. Source: prepared by the author.
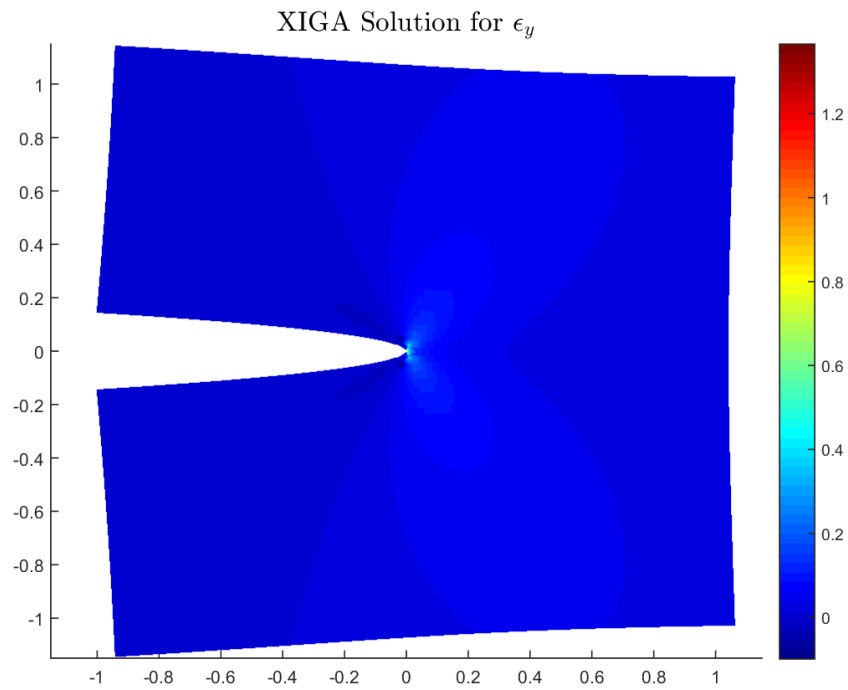
Figure 5.14: ε_y strain contour plot generated by XIGA. Source: prepared by the author.
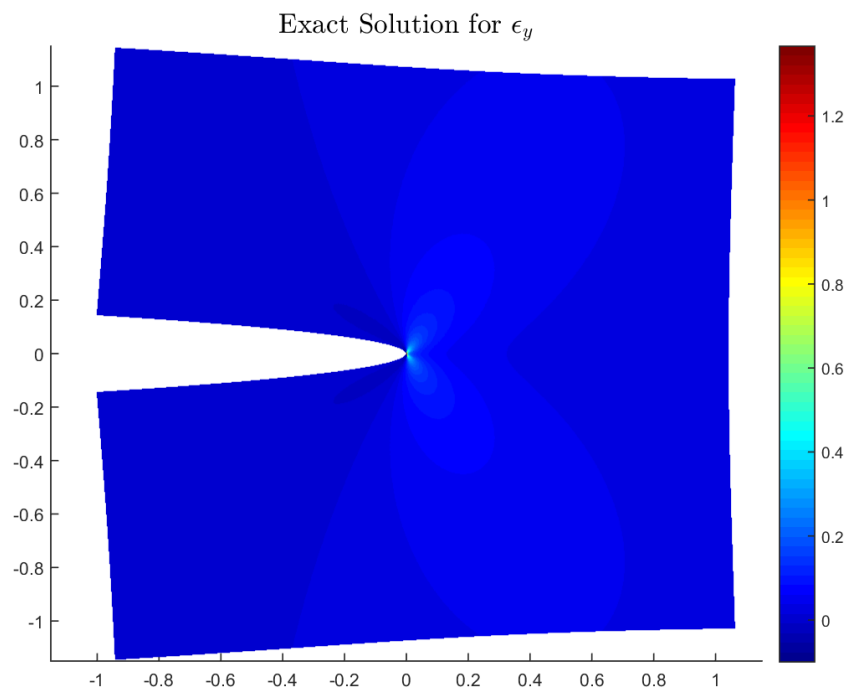
Figure 5.15: ε_y strain contour plot generated by exact solution. Source: prepared by the author.
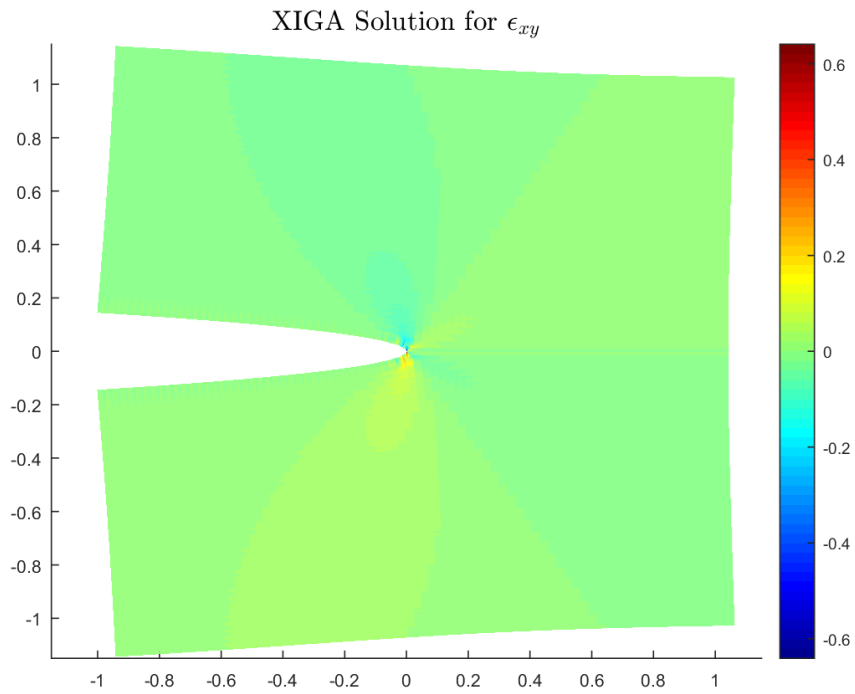
Figure 5.16: ε_{xy} strain contour plot generated by XIGA. Source: prepared by the author.
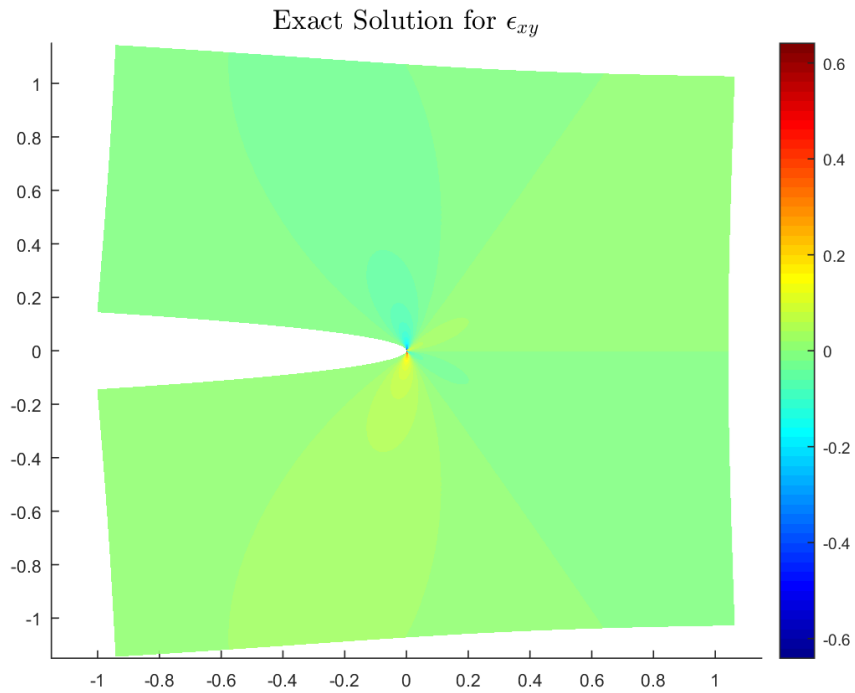
Figure 5.17: ε_{xy} strain contour plot generated by exact solution. Source: prepared by the author.
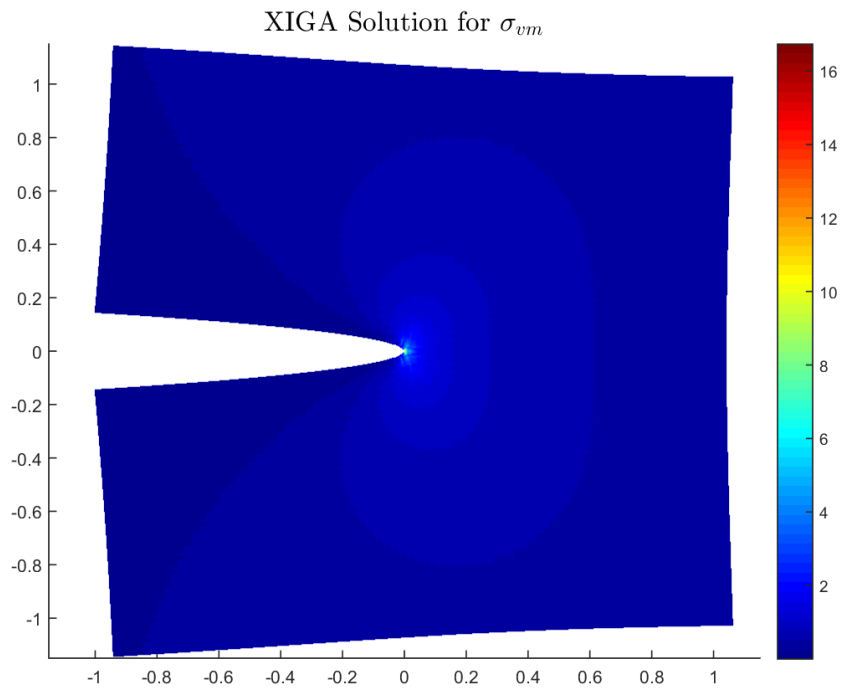
Figure 5.18: Von Mises equivalent stress contour plot generated by XIGA. Source: prepared by the author.
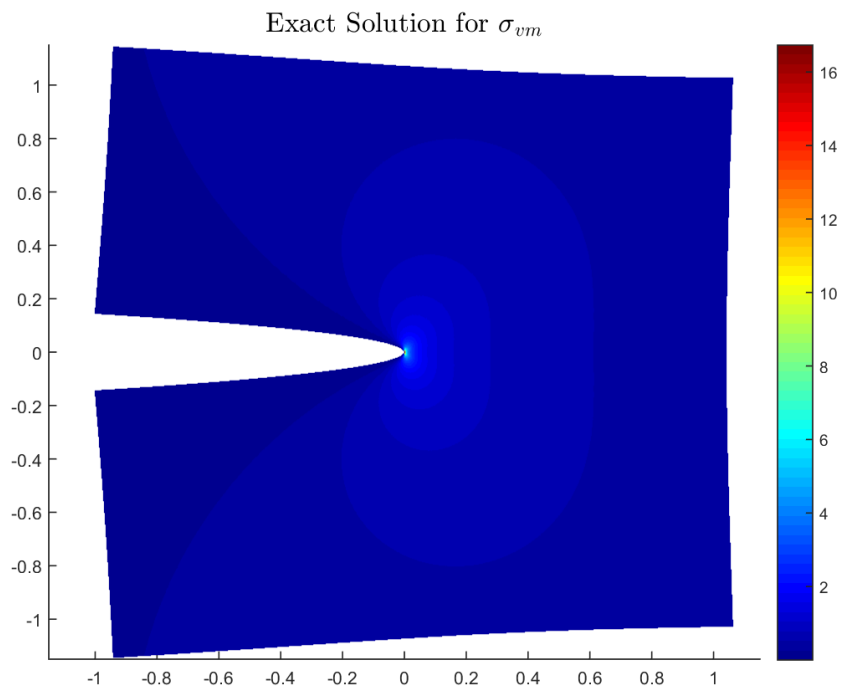
Figure 5.19: Von Mises equivalent stress contour plot generated by exact solution. Source: prepared by the author.

5.2 Plate with a Circular Hole Containing a Straight Crack

In the next example we consider the same analytical solution as in the previous numerical example (Equations (5.1, 5.2, 5.3, 5.4 and 5.5)), but applied to the geometry shown in Figure 5.20.
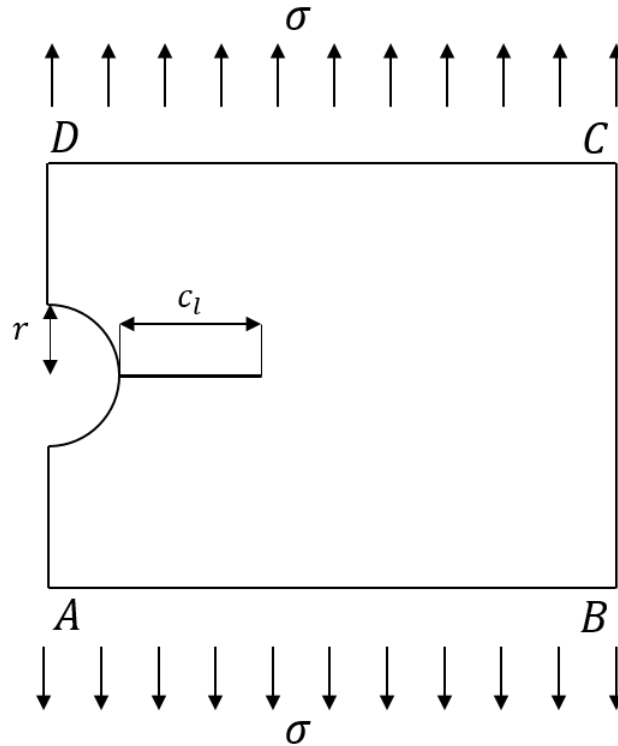
Figure 5.20: Geometry and loading in a plate with a circular hole containing a straight crack under remote tension. Source: prepared by the author.

The boundary conditions for the top, right and bottom edges are essential boundaries, while the left edge is considered as a natural boundary. The physical parameters are presented in Table 5.7. The control points and weights employed for the parametrization of the geometry and the crack are presented in Table 5.8 and Table 5.9 respectively. The knot vector for the parametrization of the geometry and the crack are presented in Table 5.10 and Table 5.11 respectively. This problem is analyzed using XIGA. Quadratic and cubic NURBS are employed for all the simulations.

Four different types of discretization has been considered to measure the accuracy and convergence rate of this numerical simulation. This is accomplish by a h-refinement method, which is a standard NURBS procedure. Table 5.12 resumes the principal information for the discretization. Figure 5.21 shows the four meshes. 4×4 Gaussian quadrature are used for the integration in each element.

Figures 5.22, 5.23 and 5.24 show the loglog L^2, H^1 and Energy norms of the error respectively, obtained for quadratic and cubic XIGA.

Figure 5.25 shows the Stress Intensity Factor K_1 for quadratic and cubic XIGA computed

Table 5.7: Physical Parameters for plate with a circular hole containing a straight crack

Physical Parameters	
Young's Modulus E	10
Poisson Ratio's ν	0.3
Mode 1 Stress Intensity Factor K_I	1
Hole's radio r	0.5
geometry size 1 $AB = CD$	2.5
geometry size 2 $BC = DA$	2
Crack length C_l	1

Source: prepared by the author.

Table 5.8: Control Points and weights for Physical Domain - Numerical Simulation 2

Control Point $\mathbf{X_{ij}}$	weight w_{ij}	Control Point $\mathbf{X_{ij}}$	weight w_{ij}	Control Point $\mathbf{X_{ij}}$	weight w_{ij}
(-1.5,-1.0)	1	(0.0,-1.0)	1	(1.0,-1.0)	1
(-1.5,-0.5)	1	(0.0,-0.85)	1	(1.0,-0.75)	1
(-1.5,-0.5)	1	(0.0,-0.75)	1	(1.0,-0.65)	1
(-1.211,-0.5)	0.866	(0.0,-0.5)	1	(1.0,-0.5)	1
(-1.066,-0.25)	1	(0.0,-0.25)	1	(1.0,-0.35)	1
(-0.922,0.0)	0.866	(0.0,0.0)	1	(1.0,0.0)	1
(-1.066,0.25)	1	(0.0,0.25)	1	(1.0,0.35)	1
(-1.211,0.5)	0.866	(0.0,0.5)	1	(1.0,0.5)	1
(-1.5,0.5)	1	(0.0,0.75)	1	(1.0,0.65)	1
(-1.5,0.5)	1	(0.0,0.85)	1	(1.0,0.75)	1
(-1.5,1.0)	1	(0.0,1.0)	1	(1.0,1.0)	1

Source: prepared by the author.

Table 5.9: Control Points and weights for crack - Numerical Simulation 2

Control Point $\mathbf{X_{ij}}$	weight w_{ij}
(-1,0)	1.0
(0,0)	1.0

Source: prepared by the author.

Table 5.10: Knot vector for Physical Domain - Numerical Simulation 2

Direction	Knot vector
ξ	$\Xi_1 = \left\{0,0,0,1,1,1\right\}$
η	$\Xi_2 = \left\{0,0,0,0.15,0.3,0.3,0.5,0.5,0.7,0.7,0.85,1,1,1\right\}$

Source: prepared by the author.

Table 5.11: Knot vector for crack - Numerical Simulation 2

Direction	Knot vector
$\xi_{\mathbf{c}}$	$\Xi_{1_c} = \{0, 0, 1, 1\}$

Source: prepared by the author.

Table 5.12: Maximum edge size and number of elements for each mesh - Numerical Simulation 2

Mesh	Maximum edge size h	Number of Elements
1	0.0971	5×3
2	0.0571	5×9
3	0.0032	15×9
4	0.0010	45×27

Source: prepared by the author.

by J-integral using the domain method.

Figures 5.26 and 5.27 show the contour plot for the u_x displacement obtained with quadratic XIGA and the exact solution respectively. Figures 5.28 and 5.29 show the contour plot for the u_y displacement obtained with quadratic XIGA and the exact solution, respectively.

Figures 5.30 and 5.31 show the contour plot for the ε_x strain obtained with quadratic XIGA and the exact solution respectively. Figures 5.32 and 5.33 show the contour plot for the ε_y strain obtained with quadratic XIGA and the exact solution respectively. Figures 5.34 and 5.35 show the contour plot for the ε_{xy} strain obtained with quadratic XIGA and the exact solution respectively.

Figure 5.36 shows the contour plot for the equivalent von Mises stress obtained with quadratic XIGA and Figure 5.37 shows the contour plot for the equivalent von Mises stress obtained with the exact solution.
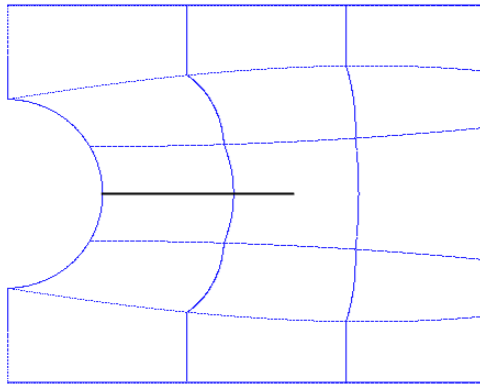
(a) Mesh 1
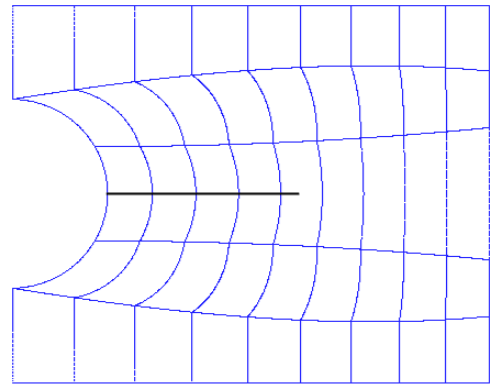
(b) Mesh 2
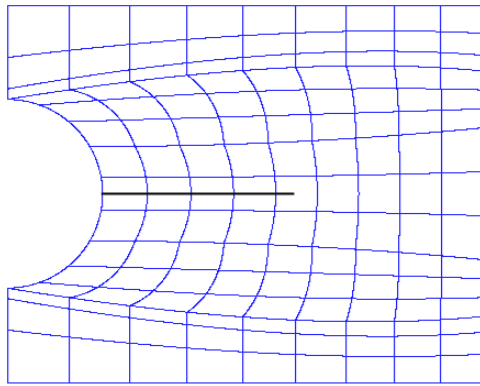
(c) Mesh 3
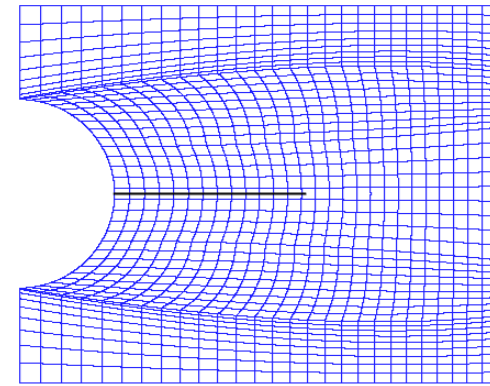
(d) Mesh 4

Figure 5.21: Plate with a circular hole containing a straight crack: Physical meshes produced by h-refinement. The black line represents the crack. Source: prepared by the author.
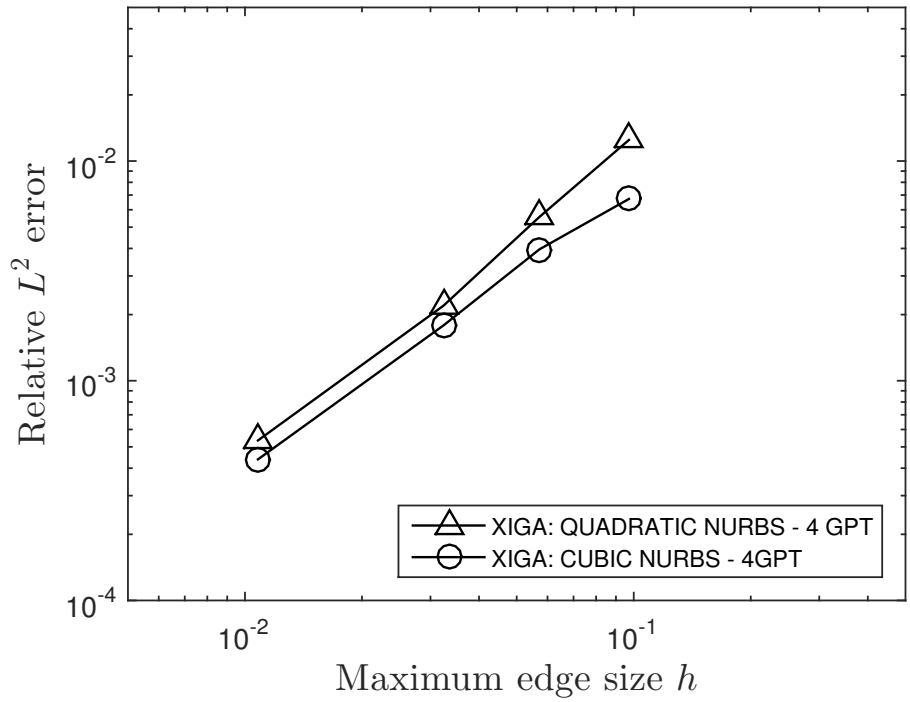
Figure 5.22: Quadratic and cubic XIGA. Convergence in the L^2 norm upon decreasing the element size. Source: prepared by the author.
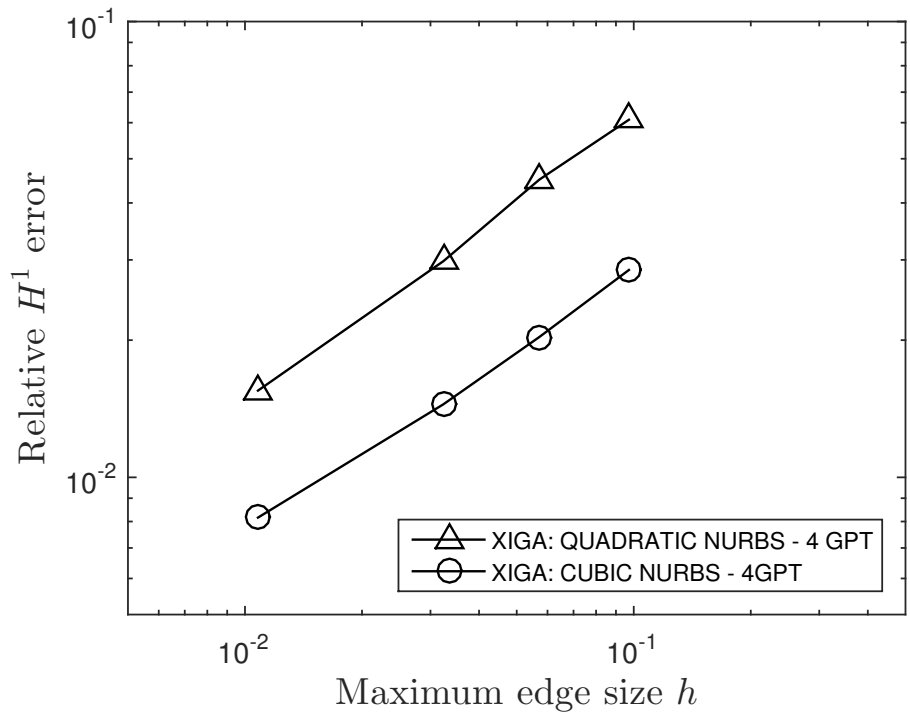
Figure 5.23: Quadratic and cubic XIGA. Convergence in the H^1 norm upon decreasing the element size. Source: prepared by the author.
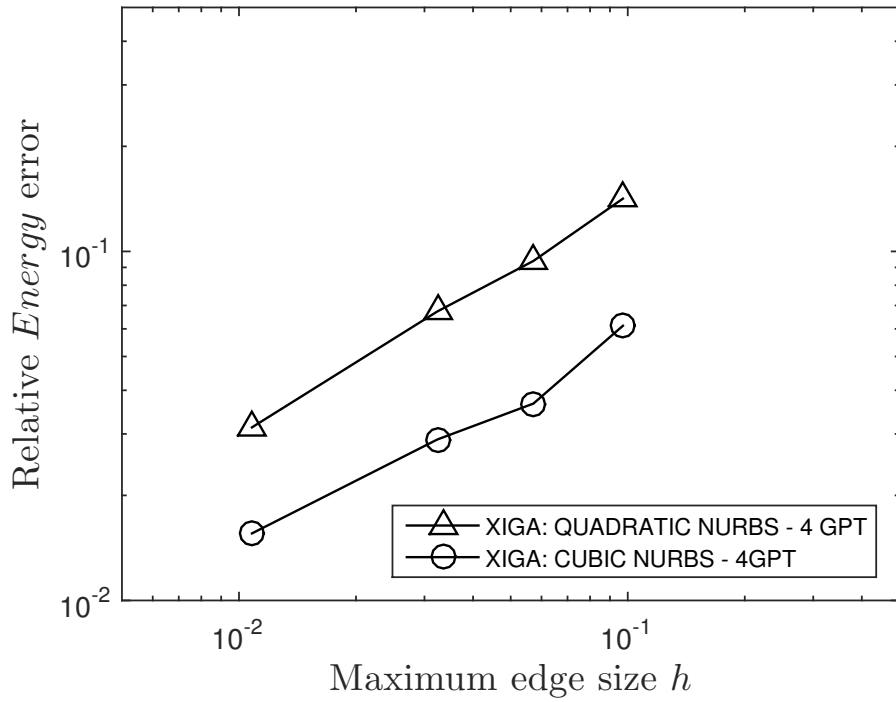
Figure 5.24: Quadratic and cubic XIGA. Convergence in the Energy norm upon decreasing the element size. Source: prepared by the author.
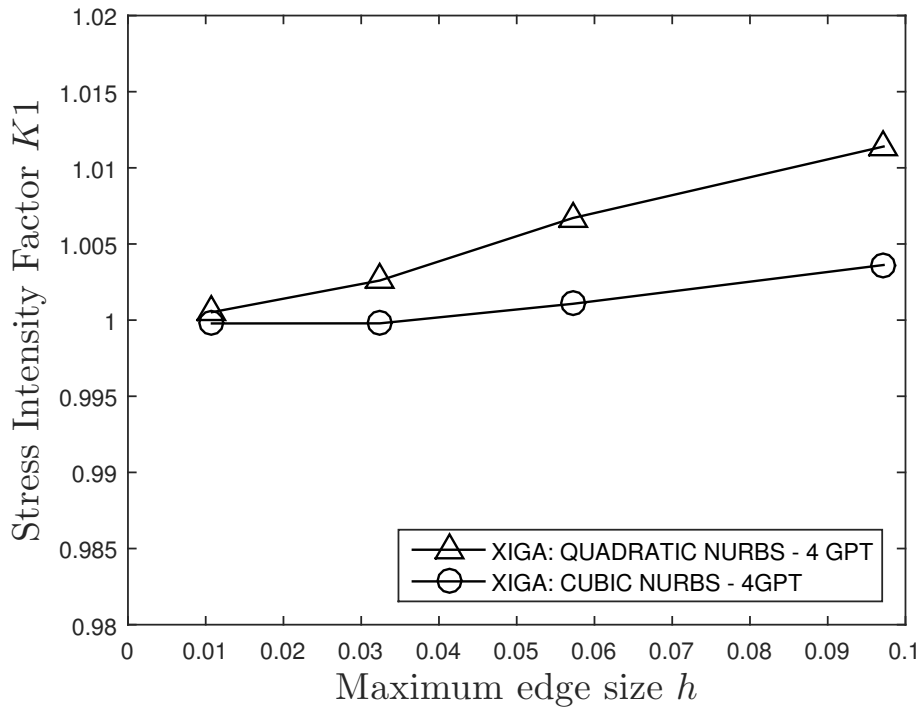
Figure 5.25: Quadratic and cubic XIGA. Convergence of the Stress Intensity Factor K_1 upon decreasing the element size. Source: prepared by the author.
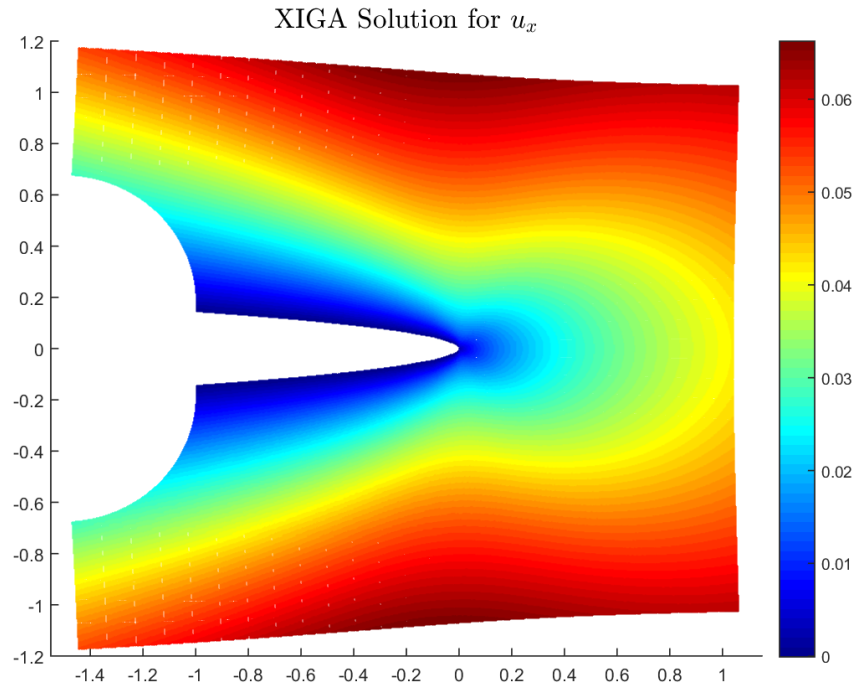
Figure 5.26: u_x displacement contour plot generated by XIGA. Source: prepared by the author.
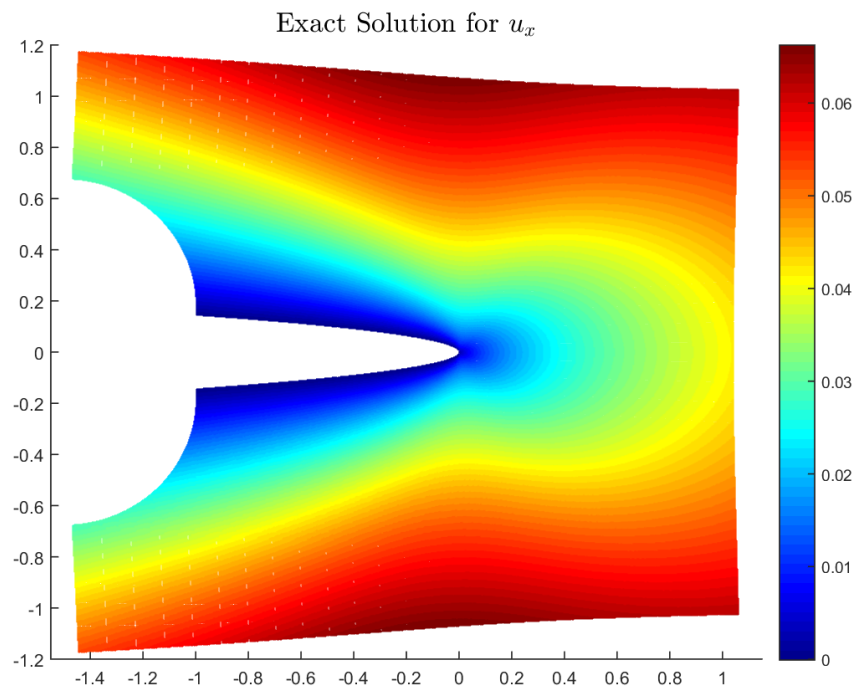
Figure 5.27: u_x displacement contour plot generated by exact solution. Source: prepared by the author.
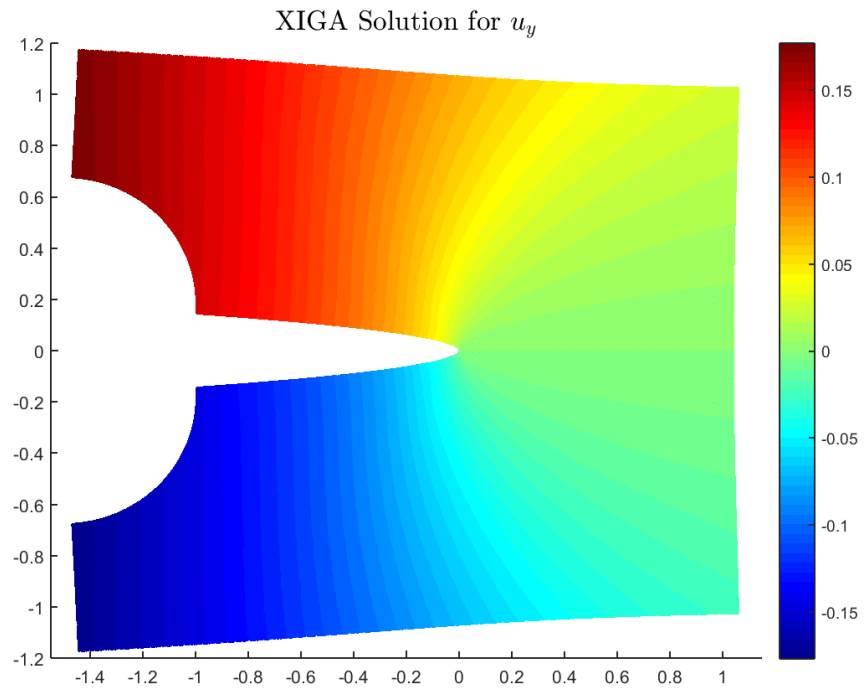
Figure 5.28: u_y displacement contour plot generated by XIGA. Source: prepared by the author.
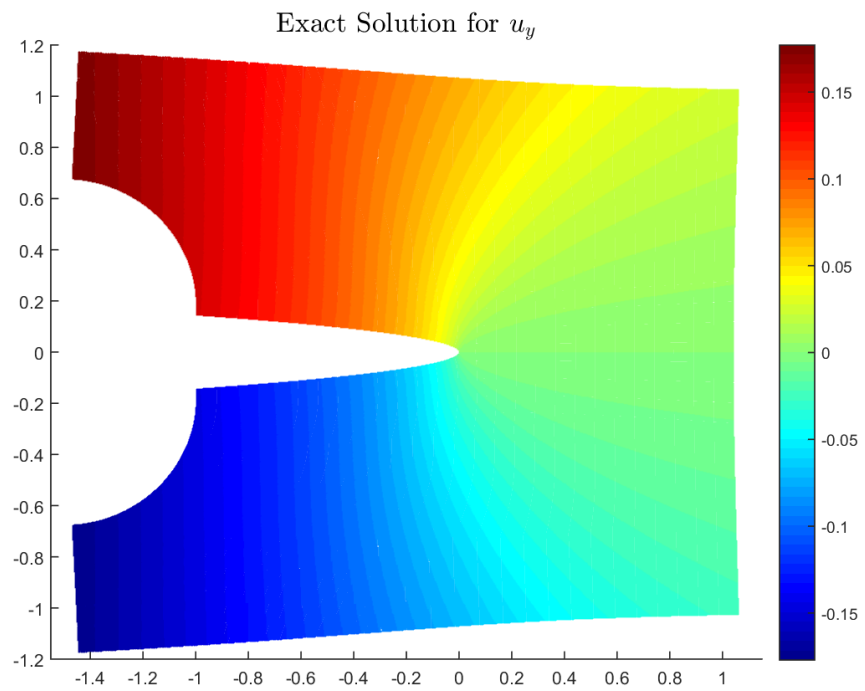
Figure 5.29: u_y displacement contour plot generated by exact solution. Source: prepared by the author.
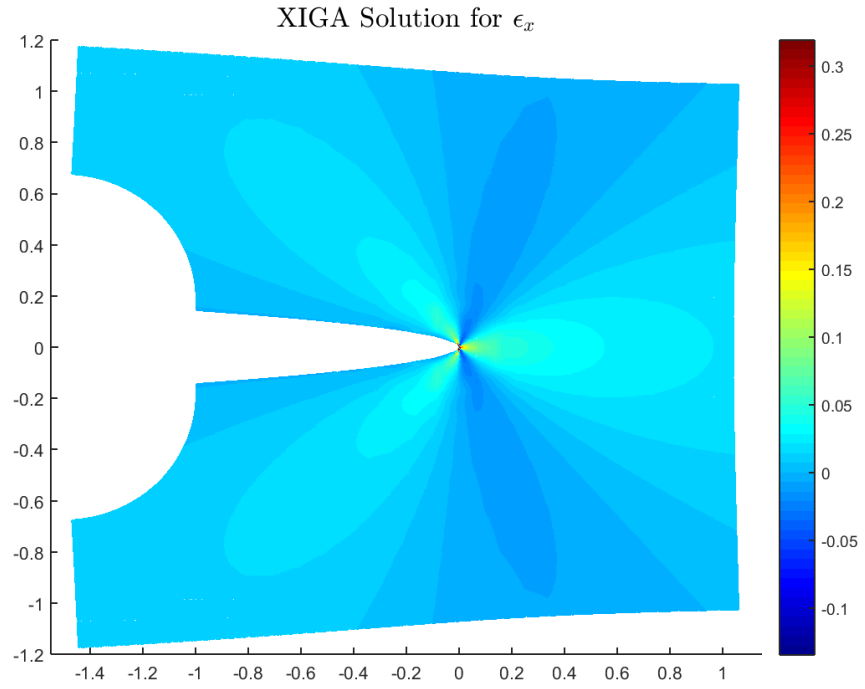
Figure 5.30: ε_x strain contour plot generated by XIGA. Source: prepared by the author.
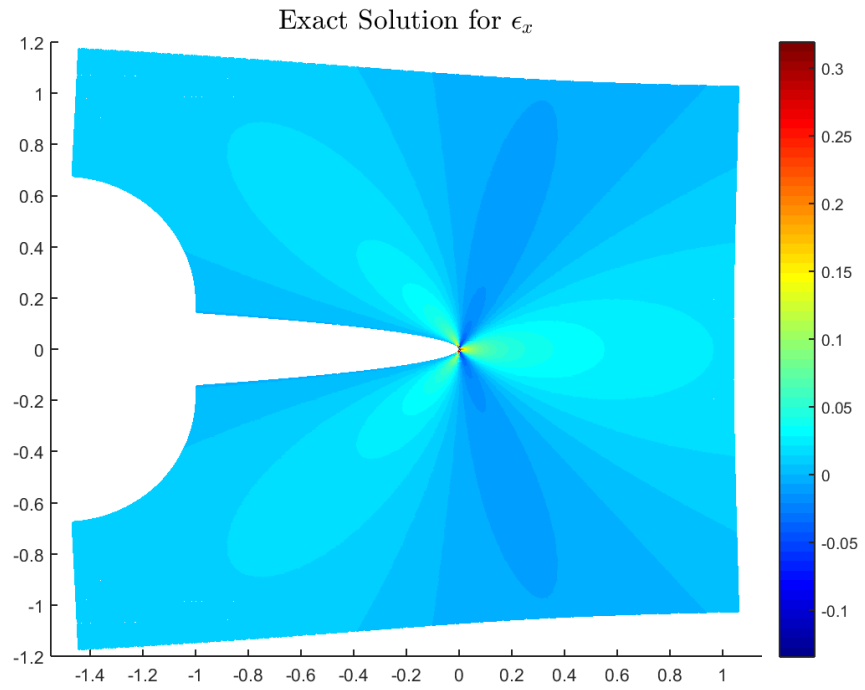
Figure 5.31: ε_x strain contour plot generated by exact solution. Source: prepared by the author.
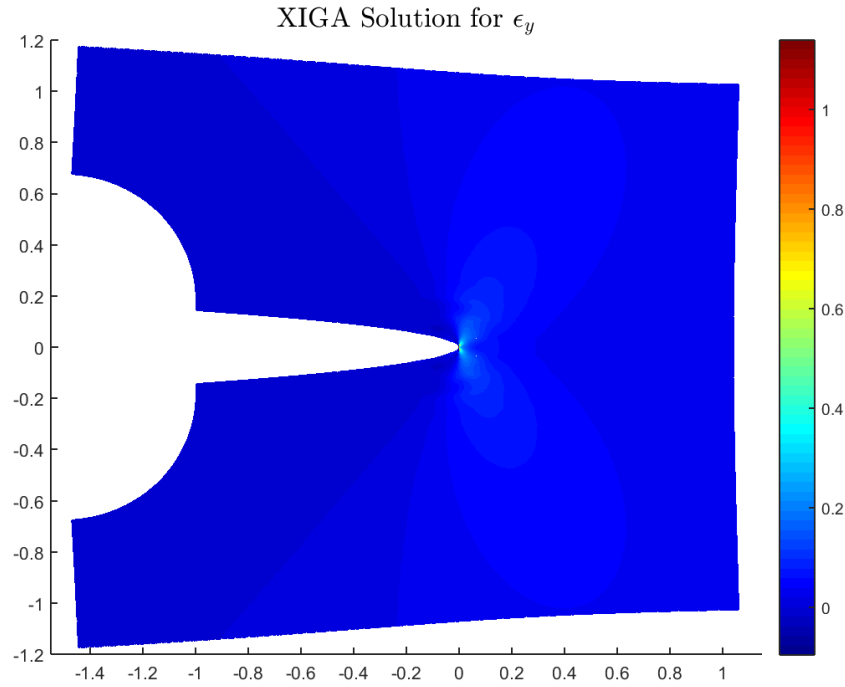
Figure 5.32: ε_y strain contour plot generated by XIGA. Source: prepared by the author.
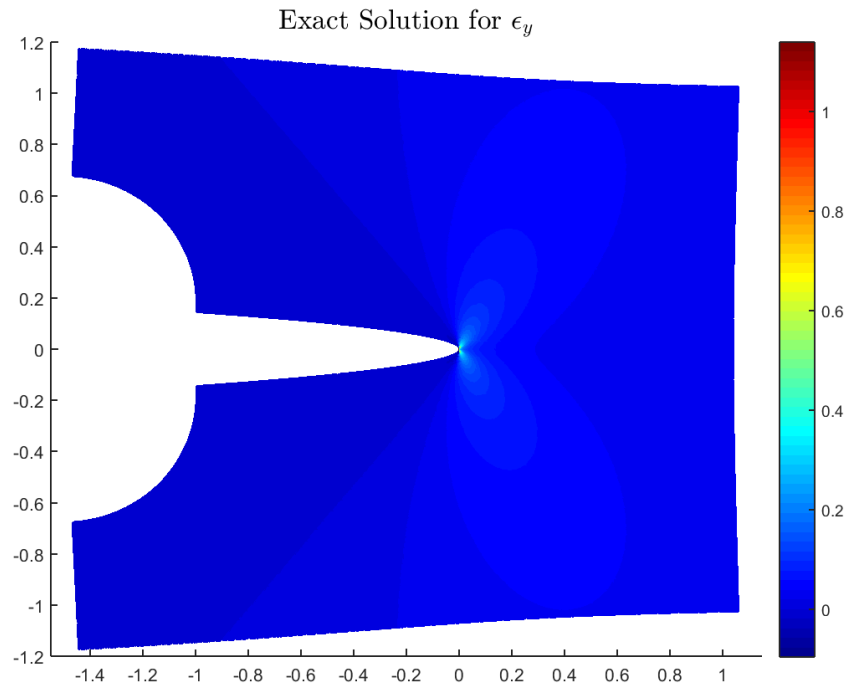
Figure 5.33: ε_y strain contour plot generated by exact solution. Source: prepared by the author.
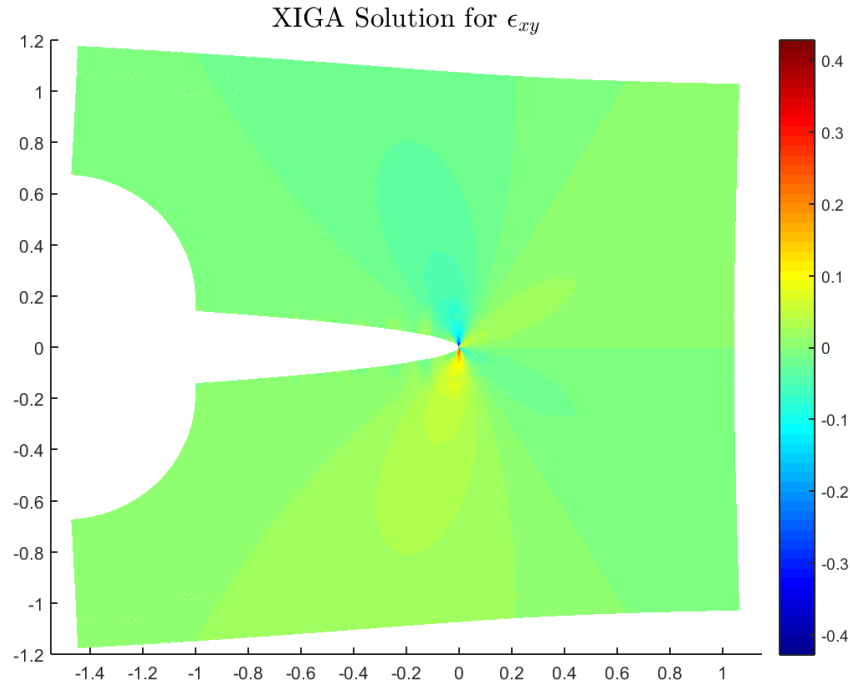
Figure 5.34: ε_{xy} strain contour plot generated by XIGA. Source: prepared by the author.
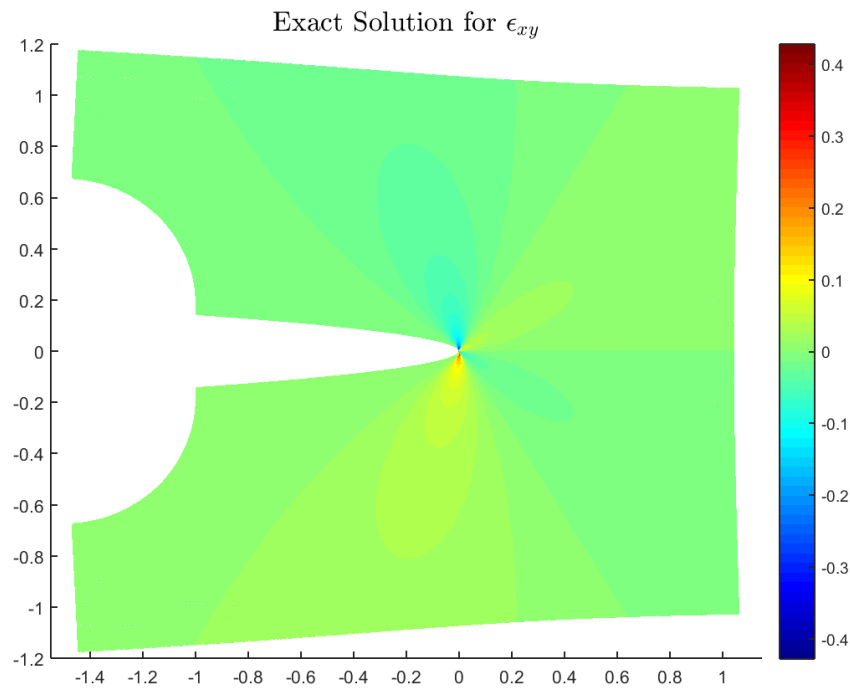
Figure 5.35: ε_{xy} strain contour plot generated by exact solution. Source: prepared by the author.
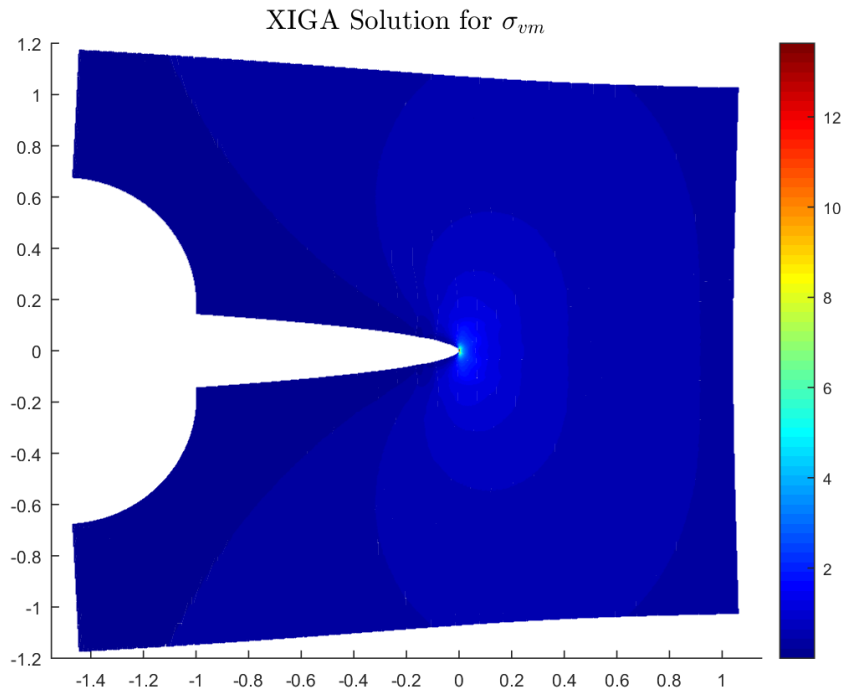
Figure 5.36: Von Mises equivalent stress contour plot generated by Quadratic XIGA. Source: prepared by the author.
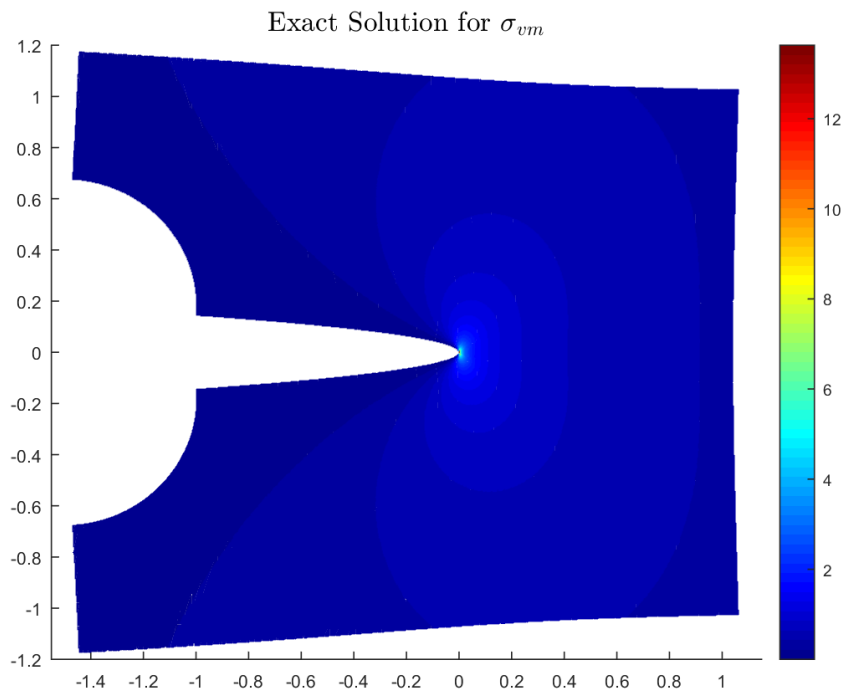
Figure 5.37: Von Mises equivalent stress contour plot generated by exact solution. Source: prepared by the author.

Chapter 6

Analysis and Discussions

6.1 XIGA Implementation

One of the main objectives of this project work is to implement the enrichment criteria from XFEM into an IGA code for solving fracture mechanics problems. In this context, several algorithms have been executed to accomplish this objective. Figure 5.3 exemplifies how all the Crack-Element Identification algorithms (Algorithms 1 and 2) works by identifying the control points that have to be enriched by the Heaviside criteria or crack tip criteria. It can also be seen from Figure 5.3 how the ratio between Crack tip enriched Control points and Heaviside enriched Control points decreases when increasing the refinement of the mesh. The reason for this is that the Crack tip criteria employed (topological enrichment) only enriches the Control points which belong to the element that has crack tip, while the Heaviside criteria enriches the Control points which belong to the elements that are crossed by the crack. By refining the mesh, this will increase the number of Heaviside enriched Control points, while the number of Crack tip enriched Control points will remain the same. It is worth noting that the improvement of the solution is impressive taking into account that the two procedures only differs in the enrichment of 4 Control points. However, this type of enrichment criteria does not allow to reach optimal convergence rate, which will be discussed in the follow paragraphs.

Regarding the integration over the elements which contain the crack, a mapping technique using trimmed NURBS has been implemented in order to perform Gaussian quadrature. The results obtained in the error graphs (Figures 5.4, 5.5 and 5.6 in numerical simulation 1 and Figures 5.22, 5.23 and 5.22 in numerical simulation 2) support this procedure even though both NURBS and enrichment functions are not necessarily polynomials. Nonetheless, it is know that there are better methods for this numerical integration, for example the almost polar integration proposed by Laborde et al. [52] allows to avoid the singularity generated by the $\frac{1}{\sqrt{r}}$ by mapping that to a triangle domain, also Chin et al. proposed the Homogeneous Numerical Integration scheme (HNI) [53] which allows to integrate over an element with a crack tip without element-partitioning. It is expected that better results can be achieved by considering those integration methods.

Regarding the crack parametrization, a NURBS curve parametrization has been implemented for this analysis. It has been useful in both the enrichment criteria and the trimmed NURBS technique since it allows to work with the same concepts as in IGA and also it enables to work with free form cracks.

Two different enrichment methods have been implemented: A "partial" enrichment with only Heaviside criteria and a "complete" enrichment with Heaviside and Crack tip criteria. For each enrichment have been developed Algorithms for the assembly of the Force vector and the stiffness matrix. All this has been done with the purpose of study the improvement of the solution when including the enrichment of the crack tip. It can be seen from Figures 5.4, 5.5 and 5.6 in numerical example 1 how the complete enrichment improves the error of the solution.

6.2   Numerical Simulation 1

The Numerical Simulation 1 - *Infinite plate with a crack under uniform tensile loading* is a classical fracture mechanics problem employed for test the XIGA implementation.

Figure 5.4 shows the relative L^2 error norm with respect to edge size h for four different meshes for a partial enrichment and a complete enrichment. It can be seen that in both cases the L^2 norm decreases when refining the mesh, which is a expected behavior. Also, Both methods show a linear behavior in the log-log plot, where the complete enrichment presents a slope equal to 1.07, while the slope of the partial enrichment is equal to 0.743. It can also be seen that the complete enrichment presents a lower L^2 error compared with the partial enrichment. The optimal slope for this error norm in linear elasticity, considering that Linear NURBS are employed for the simulations, is 2 (which appears as a reference in the plot).

Figures 5.5 and 5.6 show the relative H^1 norm of the error and Energy norm of the error with respect to edge size h for four different meshes for a partial enrichment and a complete enrichment. It can be seen in both graphs that the errors decreases when refining the mesh, which is also a expected behavior. In both plots the error for the complete enrichment is lower than the error for the partial enrichment. In the H^1 norm of error the slopes are 0.482 and 0.494 for the complete and partial enrichment, respectively. In the Energy norm of error the slopes are 0.435 and 0.428 for the complete and partial enrichment, respectively. In both cases, the optimal slope is 1 (which appears as a reference in the plot).

In the three norms, the error for the complete enrichment is lower than the error for the partial enrichment.

Regarding the convergence rates, for both partial enrichment and complete enrichment, the slopes are around 50% lower than the optimal slope mentioned above. First, it needs to be clarified that by optimal convergence rate we refer to the convergence rate expected in classical FEM or IGA, which is $||u - u_h|| = O(h^k)$, where k is the order of the NURBS. The sub-optimal convergence rate obtained is also reported in XIGA [8, 9] and XFEM [52, 54]. The explanation given by Laborde et at. [52] state that the topological enrichment improves

the solution (reduces the error), but it does not change the convergence rate. According to the same authors, the convergence rate for topological enrichment of the H^1 error norm is 0.5 and it is independent of the degree approximation. According to that, the slopes obtained for the H^1 error norm are 3.6% and 1.2% lower than the optimal slopes for the complete and partial enrichment, respectively. The difference can be attributed to the numerical integration, which is not suited for discontinuous functions. Laborde et al. state that better converge rate can be achieved by employing the geometrical enrichment (which consist in to enrich always a fixed are on the domain [54]).

Figure 5.7 shows the Stress Intensity Factor K_I computed using the J-integral domain method for a partial enrichment and a complete enrichment with respect to edge size h for four different meshes. The theoretical value is $K_I = 1$ in this numerical simulation. This plot shows the agreement between the theoretical value and the numerical value for the complete enrichment in the four meshes. On the other hand, the partial enrichment doesn't show a good agreement between the theoretical value and the numerical one for the first three meshes. In the finest mesh, the error for the K_I is 0.16% for the complete enrichment and 4.51% for the partial enrichment.

Figures 5.8, 5.9, 5.10 and 5.11 show the contour plots for the u_x and u_y displacement over the deformed shape using the XIGA methodology and the analytical solution respectively. It can be seen in both cases the agreement between the analytical solution and XIGA solution. These graphs are in line with the results obtained for the L^2 norm of error in Figure 5.4.

Figures 5.12, 5.13, 5.14, 5.15 5.16 and 5.17 show the contour plots for the ε_x, ε_y and ε_{xy} strains over the deformed shape using the XIGA methodology and the analytical solution respectively. The ε_x, ε_y and ε_{xy} plots show an excellent agreement between the analytical solution and the XIGA solution. It can also be seen that the approximation for ε_x, ε_y and ε_{xy} is constant between elements and it is due the fact that linear NURBS are employed in this simulation. All of those graphs are also in line with the results obtained for the H^1 norm of the error and Energy norm of the error in Figures 5.5 and 5.6.

Figures 5.18 and 5.19 show the contour plot for the von Mises stress σ_{vm} over the deformed shape using the XIGA methodology and the analytical solution. It can be seen in both graphics that the maximum von Mises stress focus at the crack tip area. Also, both graphics show an almost identical stress distribution and deformed shape. This result also endorse the XIGA method.

6.3   Numerical Simulation 2

The Numerical Simulation 2 - *Plate with a circular hole containing a straight crack* is solved to test the capability of the XIGA method to deal with complex geometries such as the crack tip in the middle of a circle. This numerical simulation is solved using quadratic and cubic NURBS with 4×4 Gauss points.

Figure 5.22 shows the relative L^2 norm of the error with respect to edge size h for four

different meshes. It can be seen that in the two cases the L^2 norm decreases when refining the mesh, which is a expected behavior. Also, the two different NURBS show a linear behavior in the log-log plot, where the quadratic NURBS exhibits a slope equal to 1.429, while the cubic NURBS exhibits a slope equal to 1.261. It can also be seen that the cubic NURBS presents the lower L^2 error compared with the quadratic NURBS.

Figure 5.23 shows the relative H^1 norm of the error with respect to edge size h for four different meshes using a complete enrichment with quadratic and cubic NURBS with 4×4 Gauss points. It can be seen that in the two cases the H^1 norm decreases when refining the mesh, which is a expected behavior. Also, the two NURBS show a linear behavior in the log-log plot, where the quadratic NURBS exhibits a slope equal to 0.613, while the slope for the cubic NURBS is equal to 0.567. It can also be seen that the cubic NURBS presents a lower H^1 error compared with the quadratic NURBS.

Figure 5.24 shows the relative Energy norm of the error with respect to edge size h for four different meshes using a complete enrichment with quadratic and cubic NURBS with 4×4 Gauss points. It can be seen that in all three cases the Energy norm decreases when refining the mesh, which is a expected behavior. Also, the two NURBS show a linear behavior in the log-log plot, where the quadratic NURBS exhibits a slope equal to 0.668, while the slope for the cubic NURBS is equal to 0.598. It can also be seen that the cubic NURBS presents a lower Energy error compared with the quadratic and cubic NURBS.

In a similar way as in first numerical simulation, the convergence rates for the L_2, H_1 and Energy error norm are sub-optimal and does not improve as expected when increasing the NURBS order in the simulation. In the three error plots is observed that the relative error improves by increasing the order of the NURBS, but the convergence rate does not change as expected. Again, we refer with expected convergence rate as the convergence rate obtained with classical FEM or IGA. This sub-optimal convergence rate and not improvement of the slope with the increase of the NURBS order is also reported by Laborde et al. [52] in XFEM and by Luycker et al. [8] in XIGA. In the case of XFEM, Laborde et al concluded that, even with high-order approximation, the H_1 converge rate is 0.5 and is independent of the order of the approximation employed in XFEM when topological enrichment is employed. On the other hand, Luycker et al. state that even using blending corrections and topological enrichment in XIGA, the convergence rate for H_1 is restricted to 1. Luycker et al improved that by including a modified projection method to generate the blending functions to achieve convergence near the optimal.

In this case, the sub-optimal convergence rate can be attributed to the fact that topological enrichment is employed in the simulations, no blending corrections are performed and the numerical integration may not be able to perform well with crack tip enrichment functions and NURBS. It is expected that optimal convergence can be achieved by including those modifications in the model. Regardless, optimal convergence rates are out of the scope of this final project.

Figure 5.25 shows the Stress Intensity Factor K_I computed using the J-integral domain method using a complete enrichment with quadratic and cubic NURBS with 4×4 Gauss points. The theoretical value is $K_I = 1$ in this numerical simulation. In a same way as in numerical simulation 1, it can be seen that the computed value for the K_I converges to their

theoretical value in each case. In the four mesh, the error between the theoretical value and the obtained by the XIGA is 0.052% for quadratic NURBS and 0.022% for cubic NURBS. This fact validates the results.

Figures 5.26, 5.27, 5.28 and 5.29 show the contour plots for the u_x and u_y displacement over the deformed shape using XIGA with quadratic NURBS and 4×4 Gauss points and the analytical solution respectively. It can be seen the agreement between the analytical solution and XIGA solution. These graphs are in line with the results obtained for the L^2 norm of the error in Figure 5.22.

Figures 5.30, 5.31, 5.32, 5.33 5.34 and 5.35 show the contour plots for the ε_x, ε_y and ε_{xy} strains over the deformed shape using the XIGA with quadratic NURBS and 4×4 Gauss points and the analytical solution respectively. In a same way as in numerical simulation 1, the ε_x, ε_y and ε_{xy} plots show an excellent agreement between the analytical solution and the XIGA solution. Looking at detail, it can be seen some minor irregularities in the strains around the crack tip, however, they are small enough to not undermine the error norms. All of those graphs are also in line with the results obtained for the H^1 and Energy norm of the error in Figures 5.23 and 5.24.

Figures 5.36 and 5.37 show the contour plot for the von Mises stress σ_{vm} over the deformed shape using the XIGA with quadratic NURBS and 4×4 Gauss points and the analytical solution. It can be seen in both graphics that the maximum von Mises stress focus at the crack tip area. Also, both graphics show an almost identical stress distribution and deformed shape.

Chapter 7

Conclusions and Commentaries

An Extended formulation for Isogeometric Analysis has been developed for the study of 2D fracture mechanics problems. In particular, both Heaviside and Crack tip enrichment has been implemented to solve the benchmark problems of an infinite plate with a straight crack under uniform loading and the problem of an infinite plate with a circular hole with two regular straight cracks.

XIGA allows to obtain precisely results in fracture mechanical problems. This is endorsed by all the results obtained for numerical simulations 1 and 2.

The virtues of IGA related to mesh generation are exploit in numerical simulation 2 by creating a geometry with a circular hole. This gives it an advantage compared with the classic FEM mesh process by having an exact geometry of the problem.

The integration scheme used for this work is the Gauss quadrature. In the cases of elements cut by the crack, a trimmed NURBS mapping technique is employed. Despite Gauss quadrature cannot integrate such functions as the employed in the enrichment methodology, the overall results are of great quality. Given that the integration process is out of the scope of this project work there weren't done more efforts into improve that. It is expected to obtain even lower error measurement by improving the integration technique, such as almost polar integration scheme [52] or the Homogeneous Numerical Integration (HNI) method [53].

Regarding the first numerical simulation, it can be seen that both Heaviside enrichment and Heaviside + Crack tip enrichment allow to model the crack opening, being the complete enrichment better than only the Heaviside enrichment. Nevertheless, sub-optimal convergence rate where obtained. It is expected to improve this convergence rates by improving the integration technique, changing from topological enrichment to geometrical enrichment and incorporating the blending modifications proposed by Luycker et al. [8].

Regarding the second numerical simulation, it can be seen how both quadratic and cubic NURBS are employed to model a complex geometry. In this case, L^2 norm of the error, H^1 norm of the error and Energy norm of the error exhibit sub-optimal rate. It is observed that the error norms improve with increasing the NURBS degree, but the convergence does not

improve. As mentioned before, improvements such as the integration process and enrichment modifications can be implemented in order to achieve optimal convergence rate. However, the low error and convergence rate is enough to prove the ability of XIGA to deal with cracks and complex geometries.

In both numerical simulation 1 and 2, the K_I SIF is computed by the domain method using the J-integral with the aim to compare against their theoretical value. In both cases the numerical value are in agree with their theoretical one, endorsing the results of the error measurement.

One of the main drawbacks observed by working with NURBS is the lack of local refinement. This is something desirable when the focus of the analysis is in a small area, such as an abrupt change of geometry. Unfortunately, when employing $h-$refinement in NURBS, the tensor product topology structure of NURBS expands the refinement all over the domain, creating NURBS that can be considered as superfluous. In this context, T-Splines and PHT-Splines have been introduced as generalizations of NURBS that allow local refinement. It is proposed to extend this project work by employing T-Splines or PHT-Splines in order to achieve local refinement.

Bibliography

[1] A. F. Grandt Jr, *Fundamentals of Structural Integrity: Damage Tolerant Design and Nondestructive Evaluation.* John Wiley & Sons, 2003.

[2] M. Ortiz, Y. Leroy, and A. Needleman, "A finite element method for localized failure analysis," *Computer methods in applied mechanics and engineering*, vol. 61, no. 2, pp. 189–214, 1987.

[3] T. Belytschko, Y. Lu, and L. Gu, "Crack propagation by element-free galerkin methods," *Engineering Fracture Mechanics*, vol. 51, no. 2, pp. 295–315, 1995.

[4] N. Moës, J. Dolbow, and T. Belytschko, "A finite element method for crack growth without remeshing," *International journal for numerical methods in engineering*, vol. 46, no. 1, pp. 131–150, 1999.

[5] "Adaptive loops." http://www.itaps.org/tools/services/adaptive-loops.html, 2012.

[6] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Computer methods in applied mechanics and engineering*, vol. 194, no. 39, pp. 4135–4195, 2005.

[7] G. Beer and S. P. A. Bordas, *Isogeometric Methods for Numerical Simulation*, vol. 561. Springer, 2015.

[8] E. De Luycker, D. Benson, T. Belytschko, Y. Bazilevs, and M. Hsu, "X-FEM in isogeometric analysis for linear fracture mechanics," *International Journal for Numerical Methods in Engineering*, vol. 87, no. 6, pp. 541–565, 2011.

[9] S. S. Ghorashi, N. Valizadeh, and S. Mohammadi, "Extended isogeometric analysis for simulation of stationary and propagating cracks," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 9, pp. 1069–1101, 2012.

[10] S. S. Ghorashi, N. Valizadeh, S. Mohammadi, and T. Rabczuk, "Extended isogeometric analysis of plates with curved cracks," in *Proceedings of the eighth international conference on engineering computational technology, Civil-Comp Press, Stirlingshire, UK*, 2012.

[11] G. Beer, B. Marussig, J. Zechner, C. Dünser, and T.-P. Fries, "Boundary Element Analysis with trimmed NURBS and a generalized IGA approach," *arXiv preprint*

arXiv:1406.3499v1, 2014.

[12] G. Beer, B. Marussig, and J. Zechner, "A simple approach to the numerical simulation with trimmed CAD surfaces," *Computer Methods in Applied Mechanics and Engineering*, vol. 285, pp. 776–790, 2015.

[13] Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T. J. Hughes, and G. Sangalli, "Isogeometric analysis: approximation, stability and error estimates for h-refined meshes," *Mathematical Models and Methods in Applied Sciences*, vol. 16, no. 07, pp. 1031–1090, 2006.

[14] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli, "Some estimates for h–p–k-refinement in Isogeometric Analysis," *Numerische Mathematik*, vol. 118, no. 2, pp. 271–305, 2011.

[15] L. Dalcin, N. Collier, P. Vignal, A. Côrtes, and V. M. Calo, "Petiga: A framework for high-performance isogeometric analysis," *Computer Methods in Applied Mechanics and Engineering*, 2016.

[16] M. S. Pauletti, M. Martinelli, N. Cavallini, and P. Antolin, "Igatools: An isogeometric analysis library," *SIAM Journal on Scientific Computing*, vol. 37, no. 4, pp. C465–C496, 2015.

[17] D. Rypl and B. Patzák, "From the finite element analysis to the isogeometric analysis in an object oriented computing environment," *Advances in Engineering Software*, vol. 44, no. 1, pp. 116–125, 2012.

[18] A. Reali, "An isogeometric analysis approach for the study of structural vibrations," *Journal of Earthquake Engineering*, vol. 10, no. spec01, pp. 1–30, 2006.

[19] Y. Bazilevs, V. M. Calo, T. J. Hughes, and Y. Zhang, "Isogeometric fluid-structure interaction: theory, algorithms, and computations," *Computational mechanics*, vol. 43, no. 1, pp. 3–37, 2008.

[20] D. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes, "Isogeometric shell analysis: The Reissner–Mindlin shell," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 5, pp. 276–289, 2010.

[21] C. V. Verhoosel, M. A. Scott, R. de Borst, and T. J. R. Hughes, "An isogeometric approach to cohesive zone modeling," *International Journal for Numerical Methods in Engineering*, vol. 87, no. 1-5, pp. 336–360, 2011.

[22] C. V. Verhoosel, M. A. Scott, T. J. R. Hughes, and R. de Borst, "An isogeometric analysis approach to gradient damage models," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 1, pp. 115–134, 2011.

[23] S. Cho and S.-H. Ha, "Isogeometric shape design optimization: exact geometry and enhanced sensitivity," *Structural and Multidisciplinary Optimization*, vol. 38, no. 1, pp. 53–70, 2009.

[24] S.-H. Ha, K. Choi, and S. Cho, "Numerical method for shape optimization using T-spline based isogeometric method," *Structural and Multidisciplinary Optimization*, vol. 42, no. 3, pp. 417–428, 2010.

[25] R. Sevilla, S. Fernández-Méndez, and A. Huerta, "NURBS-enhanced finite element method (NEFEM)," *International Journal for Numerical Methods in Engineering*, vol. 76, no. 1, pp. 56–83, 2008.

[26] G. Legrain, "A NURBS enhanced extended finite element approach for unfitted CAD analysis," *Computational Mechanics*, vol. 52, no. 4, pp. 913–929, 2013.

[27] R. N. Simpson, S. P. A. Bordas, J. Trevelyan, and T. Rabczuk, "A two-dimensional isogeometric boundary element method for elastostatic analysis," *Computer Methods in Applied Mechanics and Engineering*, vol. 209, pp. 87–100, 2012.

[28] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. Hughes, S. Lipton, M. A. Scott, and T. W. Sederberg, "Isogeometric analysis using T-splines," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 5, pp. 229–263, 2010.

[29] M. R. Dörfel, B. Jüttler, and B. Simeon, "Adaptive isogeometric analysis by local h-refinement with T-splines," *Computer methods in applied mechanics and engineering*, vol. 199, no. 5, pp. 264–275, 2010.

[30] N. Nguyen-Thanh, H. Nguyen-Xuan, S. P. A. Bordas, Bordas, and T. Rabczuk, "Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids," *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 21, pp. 1892–1908, 2011.

[31] P. Wang, J. Xu, J. Deng, and F. Chen, "Adaptive isogeometric analysis using rational PHT-splines," *Computer-Aided Design*, vol. 43, no. 11, pp. 1438–1448, 2011.

[32] N. Nguyen-Thanh, K. Zhou, X. Zhuang, P. Areias, H. Nguyen-Xuan, Y. Bazilevs, and T. Rabczuk, "Isogeometric analysis of large-deformation thin shells using rht-splines for multiple-patch coupling," *Computer Methods in Applied Mechanics and Engineering*, vol. 316, pp. 1157–1178, 2017.

[33] G. Bhardwaj, I. Singh, and B. Mishra, "Numerical simulation of plane crack problems using extended isogeometric analysis," *Procedia Engineering*, vol. 64, pp. 661–670, 2013.

[34] S. S. Ghorashi, N. Valizadeh, S. Mohammadi, and T. Rabczuk, "T-spline based XIGA for fracture analysis of orthotropic media," *Computers & Structures*, vol. 147, pp. 138–146, 2015.

[35] K. Gürlebeck, T. Lahmer, and F. Werner, "Extended isogeometric analysis based crack identification applying multilevel regularizing methods," *19th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering*, 2012.

[36] T. Q. Bui, S. Hirose, C. Zhang, T. Rabczuk, C.-T. Wu, T. Saitoh, and J. Lei, "Extended

isogeometric analysis for dynamic fracture in multiphase piezoelectric/piezomagnetic composites," *Mechanics of Materials*, vol. 97, pp. 135–163, 2016.

[37] T. Yu, T. Q. Bui, S. Yin, D. H. Doan, C. Wu, T. Van Do, and S. Tanaka, "On the thermal buckling analysis of functionally graded plates with internal defects using extended isogeometric analysis," *Composite Structures*, vol. 136, pp. 684–695, 2016.

[38] S. Yin, T. Yu, T. Q. Bui, P. Liu, and S. Hirose, "Buckling and vibration extended isogeometric analysis of imperfect graded reissner-mindlin plates with internal defects using nurbs and level sets," *Computers & Structures*, vol. 177, pp. 23–38, 2016.

[39] A. P. Boresi and R. J. Schmidt, *Advanced Mechanics of Materials*. John Wiley & Sons, Inc., New York, 2003.

[40] M. H. Sadd, *Elasticity: Theory, Applications, and Numerics*. Academic Press, 2009.

[41] S. Mohammadi, *Extended Finite Element Method: for Fracture Analysis of Structures*. John Wiley & Sons, 2008.

[42] X. Peng, E. Atroshchenko, P. Kerfriden, and S. P. A. Bordas, "Linear elastic fracture simulation directly from cad: 2d NURBS-based implementation and role of tip enrichment," *International Journal of Fracture*, pp. 1–24, 2016.

[43] J. R. Rice, "A path independent integral and the approximate analysis of strain concentration by notches and cracks," *Journal of applied mechanics*, vol. 35, no. 2, pp. 379–386, 1968.

[44] F. Z. Li, C. F. Shih, and A. Needleman, "A comparison of methods for calculating energy release rates," *Engineering Fracture Mechanics*, vol. 21, no. 2, pp. 405–421, 1985.

[45] G.-R. Liu and S. S. Quek, *The Finite Element Method: A Practical Course*. Butterworth-Heinemann, 2013.

[46] J. N. Reddy, *An Introduction to The Finite Element Method*, vol. 2. McGraw-Hill New York, 1993.

[47] V. P. Nguyen, C. Anitescu, S. P. A. Bordas, and T. Rabczuk, "Isogeometric analysis: An overview and computer implementation aspects," *Mathematics and Computers in Simulation*, vol. 117, pp. 89–116, 2015.

[48] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*. John Wiley & Sons, 2009.

[49] J. M. Melenk and I. Babuška, "The partition of unity finite element method: basic theory and applications," *Computer methods in applied mechanics and engineering*, vol. 139, no. 1, pp. 289–314, 1996.

[50] L. Berger-Vergiat, H. Waisman, B. Hiriyur, R. Tuminaro, and D. Keyes, "Inexact Schwarz-algebraic multigrid preconditioners for crack problems modeled by extended

finite element methods," *International Journal for Numerical Methods in Engineering*, vol. 90, no. 3, pp. 311–328, 2012.

[51] L. Piegl and W. Tiller, *The NURBS Book*. Springer Science & Business Media, 2012.

[52] P. Laborde, J. Pommier, Y. Renard, and M. Salaün, "High-order extended finite element method for cracked domains," *International Journal for Numerical Methods in Engineering*, vol. 64, no. 3, pp. 354–381, 2005.

[53] E. B. Chin, J.-B. Lasserre, and N. Sukumar, "Modeling crack discontinuities without element-partitioning in the extended finite element method," *International Journal for Numerical Methods in Engineering*, vol. 110, no. 11, pp. 1021–1048, 2017.

[54] É. Béchet, H. Minnebo, N. Moës, and B. Burgardt, "Improved implementation and robustness study of the X-FEM for stress analysis around cracks," *International Journal for Numerical Methods in Engineering*, vol. 64, no. 8, pp. 1033–1056, 2005.

Appendix A

Weak form formulation

Recalling the strong form of the elastostatics equation in the absence of body forces is:

$$\bigtriangledown \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = 0 \tag{A.1}$$

In index notation it can be written as:

$$\sigma_{ij,j}(\boldsymbol{u}) = 0 \tag{A.2}$$

with the following boundary conditions:

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \bar{t} \text{ on } \Gamma_{\bar{t}} \tag{A.3}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{c^+} \tag{A.4}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{c^-} \tag{A.5}$$

$$\boldsymbol{u} = \bar{u} \text{ on } \Gamma_{\bar{u}} \tag{A.6}$$

where $\boldsymbol{n}$ is the unit outward normal, $\boldsymbol{\sigma}$ is the stress tensor and $\boldsymbol{u}$ is the displacement vector.

Considering an arbitrary function $\boldsymbol{w}$, called the weight function, we multiply all the equation with $\boldsymbol{w}$ and integrate over the domain Ω:

$$\int_{\Omega} \sigma_{ij,j}(\boldsymbol{u}) w_i \mathrm{d}\Omega = 0 \tag{A.7}$$

I

Integrating by parts, we obtain:

$$\int_\Omega (\sigma_{ij}(\boldsymbol{u})w_i)_{,j}\, \mathrm{d}\Omega - \int_\Omega (\boldsymbol{u})\sigma_{ij}w_{i,j}\mathrm{d}\Omega = 0 \tag{A.8}$$

Applying the divergence theorem to the first integral, we obtain:

$$\int_\Gamma (\sigma_{ij}(\boldsymbol{u})w_i) \cdot n_j\mathrm{d}\Gamma - \int_\Omega \sigma_{ij}(\boldsymbol{u})w_{,j}\mathrm{d}\Omega = 0 \tag{A.9}$$

Substituting with the traction definition $\boldsymbol{t} = \boldsymbol{\sigma n}$, we obtain:

$$\int_\Gamma t_i w_i\mathrm{d}\Gamma - \int_\Omega \sigma_{ij}(\boldsymbol{u})w_{i,j}\mathrm{d}\Omega = 0 \tag{A.10}$$

Any tensor can be written as a sum of symmetric and antisymmetric parts, so we write $\bigtriangledown\boldsymbol{w}$ as:

$$w_{i,j} = \frac{1}{2}\left(w_{i,j} + w_{j,i}\right) + \frac{1}{2}\left(w_{i,j} - w_{j,i}\right) \tag{A.11}$$

recalling that the product of a symmetric and an antisymmetric tensor is 0 and that $\boldsymbol{\sigma}$ is a symmetric tensor, we have that:

$$\sigma_{ij}(\boldsymbol{u})w_{i,j} = \sigma_{ij}(\boldsymbol{u})\left(\frac{1}{2}\left(w_{i,j} + w_{j,i}\right) + \frac{1}{2}\left(w_{i,j} - w_{j,i}\right)\right) \tag{A.12}$$

$$\sigma_{ij}w_{i,j} = \sigma_{ij}(\boldsymbol{u})\frac{1}{2}\left(w_{i,j} + w_{j,i}\right) + \sigma_{ij}(\boldsymbol{u})\frac{1}{2}\underbrace{\left(w_{i,j} - w_{j,i}\right)}_{}{}^{\!\!\!0}$$

recalling that:

$$\varepsilon_{ij}(\boldsymbol{u}) = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{A.13}$$

we can write Equation (A.12) as:

$$\sigma_{ij}(\boldsymbol{u})w_{i,j} = \sigma_{ij}(\boldsymbol{u})\varepsilon_{ij}(\boldsymbol{w}) \tag{A.14}$$

and replacing Equation (A.14) on Equation (A.10) we obtain:

$$\int_\Gamma t_i w_i\mathrm{d}\Gamma - \int_\Omega \sigma_{ij}(\boldsymbol{u})\varepsilon_{ij}(\boldsymbol{w})\mathrm{d}\Omega = 0 \tag{A.15}$$

II

which can be written as:

$$\int_\Omega \sigma(\boldsymbol{u}) : \varepsilon(\boldsymbol{w}) \mathrm{d}\Omega = \int_\Gamma \bar{t} \cdot \boldsymbol{w} \mathrm{d}\Gamma \tag{A.16}$$

which is known as the weak form of the elastostatics equation. It is worth notice that since this is an integral equation, the requirements for $\boldsymbol{u}$ and $\boldsymbol{w}$ are the follow:

a. $\boldsymbol{u}$ belongs to the space of all the vector functions whose derivatives are square-integrable on Ω (known as the Hilbert space $H^1(\Omega)$) which satisfy the essential boundary conditions.

b. $\boldsymbol{w}$ belongs to the space of all the vector functions whose derivatives are square-integrable on Ω (known as the Hilbert space $H^1(\Omega)$) which vanish on Ω_u and are discontinuous on Ω_c.