A NURBS-enhanced Discontinuity-Enriched Finite Element Method

by

Elena De Lazzari

to obtain the degree of Master of Science
in Mechanical Engineering
at the Delft University of Technology,
to be defended publicly on Friday October 20th, 2017 at 9:00 AM.

Student number:	4518578	
Project duration:	December 10th, 2016 – October 3rd, 2017	
Thesis committee:	Dr. ir. A. M. Aragón,	TU Delft, 3mE, supervisor
	Prof. dr. ir. F. van Keulen,	TU Delft, 3mE
	Dr. Ir. F. P. van der Meer,	TU Delft, CEG
	Dr. S. R. Turteltaub,	TU Delft, AE

TU Delft

Abstract

Generalized finite element methods have proved a great potential in the mesh-independent modeling of both weak and strong discontinuities, such as the ones encountered when treating materials with inclusions or cracks. By removing the constraint of a conforming mesh, more freedom is offered to modeling exact geometries by means of splines. However, very few studies have been published which combine Non-Uniform Rational B-Splines (NURBS) to interface-enriched methods, addressing uniquely weak discontinuities.

Therefore, the aim of this thesis is to propose a NURBS-based enhancement to the Discontinuity-Enriched Finite Element Method (DE-FEM) in two dimensions and to discuss the potential of its application. The main advantage of this method is the possibility to study problems that present discontinuities with arbitrary smooth shapes, while maintaining exact geometries throughout the analysis: in this way, the equivalence between design and computational geometry is preserved. To this purpose, a suitable NURBS-based analysis technique is selected and implemented within the framework offered by the group's finite element library, Hybrida.

The capabilities of the NURBS-enhanced DE-FEM to solve several weakly discontinuous problems are assessed for composites of different complexities. Furthermore, a novel study is presented which extends this technique to the treatment of strong discontinuities, in the context of fracture mechanics. The accuracy, convergence properties and numerical efficiency of the proposed method are investigated, in particular in comparison with the standard DE-FEM. Based on these observations, further insights are provided into the convenience and the limitations of adopting NURBS enhancements within the DE-FEM formulation. Lastly, some recommendations about possible directions of improvement are provided.

Acknowledgements

This project is the result of months of hard work and personal involvement. However, all of this wouldn't have been possible without the contribution of some people who supported me along this journey.

First I would like to express my gratitude to my supervisor, Dr. ir. Alejandro Aragón, for the valuable suggestions and discussions that inspired me throughout these months. His sincere interest in my work and his enthusiasm in leading the team provided an essential contribution to my personal motivation and to the development of my thesis.

What I have accomplished in this project would not have been possible, to this extent, without the framework offered by Hybrida and without all the people who dedicated energy and passion to its growth. Some special words deserve to be spent for Ir. Jian Zhang, who offered patient help throughout my project, and to Ir. Sanne van den Boom, who supported me with the implementation since the very beginning. I am very grateful to both for the time they dedicated to providing feedback and corrections to my report, as much as for inputs that helped me constantly improve my work.

The motivation and energy I found during these months were also due, to a large extent, to the sparkling environment that surrounded me. Therefore, I would like to say a big thank you to the office crew, for adding entertainment to my daily work and for sharing drinks, meals and cheer. In particular, I would like to mention Ryan van Dommelen, Bart Holtzer, Lili Maxime Hauzer, Jelmer de Zeeuw, Heleen Payens and Michael Mengolini, for the exchange of perspectives, the tea breaks, the suggestions about my work and the commitment to helping me learn Dutch. I would also like to dedicate some words to the doctoral and post-doctoral students of the PME department, who were always willing to provide feedback and inputs and who offered warm company during the weeks in which we happened to share the office.

Moreover, an important contribution to this work came from Antonio Mattia Arbues, my spiritual counsellor, who helped me learn how to use graphics editors and to whom I owe some taste for what a good layout should look like.

At the end of this long way through my studies, I would like to thank my family. I am grateful to my parents, for raising me in a home full of books and bolts, and to my sister, for being my number one fan.

To conclude, the two years of my Master's would not have been the same without the friends I met in Delft, for making this experience worth being remembered, and without the ones that were always there since before. In particular, I would like to thank Cristina and my life companions from Shanghai, with whom this all started and will eventually get to an end.

Elena De Lazzari
Delft, October 2017

Contents

List of Figures

List of Tables

Nomenclature and abbreviations

Nomenclature

In this work, light letters conventionally represent scalars, lower-case bold letters represent vectors and upper-case bold letters represent matrices.

$\mathbf{b}$	control point
$\mathbf{c}(s)$	spline curve in physical coordinates
E	Young's modulus
$\bar{f}$	body force
h	mesh size
$\mathbf{J}$	Jacobian matrix
K_I	mode I stress intensity factor
K_{II}	mode II stress intensity factor
M	interaction energy integral (M-integral)
N	FEM shape function, B-spline basis function
p	order of a spline
R	NURBS basis function
s	parametric coordinate for splines, knot
$\mathbf{u}$	solution field, displacement field
$\mathbf{u}^h$	FEM approximation of the solution field
w	weight (for Gauss points or splines)
$\mathbf{x}$	coordinates in the physical space
α	semi-angle spanned by an arc crack
$\boldsymbol{\alpha}$	(weak) enriched degree of freedom
$\boldsymbol{\beta}$	strong enriched degree of freedom
ε	strain
ϵ_{L_2}	L_2-norm of the error
ϵ_E	error in the energy norm
η	reference domain ordinate
λ	first Lamé parameter
μ	second Lamé parameter
ν	Poisson's ratio
ξ	reference domain abscissa
σ	stress, applied boundary traction
$\mathbf{\Phi}$	NURBS-enhanced mapping
χ	strong enrichment function
ψ	(weak) enrichment function

Abbreviations

CAD	Computer-Aided Design
DE-FEM	Discontinuity-Enriched Finite Element Method
DOF	Degree Of Freedom
FEA	Finite Element Analysis
FEM	Finite Element Method
GFEM	Generalized Finite Element Method
HIFEM	Hierarchical Interface-enriched Finite Element Method
IGA	IsoGeometric Analysis
IGFEM	Interface-enriched Generalized Finite Element Method
NeDEFEM	NURBS-enhanced Discontinuity-Enriched Finite Element Method
NEFEM	NURBS-Enhanced Finite Element Method
NIGFEM	NURBS-enhanced Interface-enriched Generalized Finite Element Method
NURBS	Non-Uniform Rational B-Spline
p-FEM	*p*-version of the Finite Element Method
SIF	Stress Intensity Factor
XFEM	eXtended Finite Element Method
X-IGA	eXtended IsoGeometric Analysis

Pronaque cum spectent animalia cetera terram,
Os homini sublime dedit caelumque videre
Iussit et erectos ad sidera tollere vultus.

Ovidius

To my artist.

1 Introduction

Discontinuities are often present in engineering problems. Typical examples are offered by composites, where jumps in the gradient field, also known as *weak* discontinuities, are encountered at the interface between two heterogeneous phases. Fracture problems are another classical example, where the field itself presents C^{-1}-continuity across the crack. Such a case is referred to as *strong* discontinuity. Across discontinuities, not only material properties can differ, but also the physics that dominates the problem can change, as in the case of fluid-structure interaction problems.

In continuum mechanics, the presence of discontinuities strongly influences the behavior of a structure. This is particularly evident in metamaterials, a special class of composites where voids or inclusions in patterned configurations result globally in novel material properties, such as the ones displayed by structures with negative Poisson ratios or phononic crystals, which are able to screen acoustic waves within determined ranges of frequencies. Besides cases in which discontinuities result in enhanced material characteristics, inclusions and fractures often have a detrimental effect: the presence of cracks and internal flaws plays in fact a dominant role in lowering structures' strength and durability. Therefore, being able to quantify the effect of discontinuities, also in relation with their shape and distribution, takes on major importance in understanding a structure's characteristics and behavior under specific conditions.

The study of discontinuous problems can be performed through an experimental approach, requiring tests and measurements on real samples. When conventional computational methods are not reliable, experimental techniques can be a good option, but present several drawbacks. Besides being usually quite costly in terms of both equipment and time, one frequent issue is associated with accuracy. It is generally difficult to reproduce the ideal conditions of the problem exactly: one could just think about the precision of a specimen's dimensions compared to its nominal size. Furthermore, the measurements obtained from a test may present disturbances and imprecisions and they may be hard to obtain for a certain target parameter or for challenging test conditions. Repeatability and non-destructiveness may also represent constraints when carrying out measurements in an experimental setting and, often, a consistent number of tests have to be carried out to obtain statistically relevant results.

The disadvantages of experimental analysis have led to the development of computational methods, that analyse engineering problems by means of numerical techniques. Discontinuities usually require an *ad hoc* treatment in numerical analysis: in the framework of Finite Element Analysis (FEA), for example, a conventional approach is building a conforming finite element mesh. This procedure often results in a costly and laborious process, especially in cases when interfaces evolve in time, as in crack growth problems or shape optimization, or when discontinuities present complex shapes, for which good mesh refinements are needed to achieve acceptable quality, especially in low-order approximations. These drawbacks have drawn increasing attention towards methods that can represent interfaces independently of the mesh: hence the development of techniques such as meshless methods or generalized finite element methods.

The rest of the chapter will first examine the standard finite element formulation and its limitations in representing discontinuities. Subsequently, a literature review of generalized finite element methods will be covered. Particular focus will be dedicated, then, to illustrating techniques to model exact geometries, followed by a discussion of relevant examples of how these can be included into generalized finite element formulations, to achieve accurate mesh-independent representations of curved discontinuities.

1.1. The Finite Element Method

The most widespread approach to engineering analysis is the Finite Element Method (FEM), which is founded on the principle of creating a computational model of a physical problem to be solved for a determined field. Such a problem is usually described by means of differential equations, for which a closed-form solution is often not available. This requires approximating the unknown field, by defining a finite set of unknown degrees of freedom that can be computed numerically. This approximation of the field is constructed over a discretization of the problem's geometry, which takes the form of a mesh of finite elements. The degrees of freedom are associated with the mesh nodes and these nodal values are interpolated into a continuous field throughout the physical domain by space-dependent shape functions:

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} N_i(\mathbf{x})\mathbf{u}_i \tag{1.1}$$

With this approach, the differential equations that define the problem are reduced to a system of algebraic equations.

In order to recover correct nodal values, the shape functions $N_i(\mathbf{x})$ have to be chosen to meet specific criteria. In particular, each shape function needs to satisfy the Kronecker delta property, by assuming value one at its associated node and zero at all the others. Lagrangian shape functions, of first or higher order, are a classical choice, for which the general expression of the p-th order function associated with node x_j is:

$$L_j^p(x) = \prod_{\substack{i=0 \\ i\neq j}}^{p} \frac{x - x_i}{x_j - x_i} \tag{1.2}$$

Lagrangian shape functions are a good general purpose option for modeling solution fields; however, they limit the capabilities of the standard finite element method of capturing discontinuities within elements. These limitations can be overcome by generating meshes that conform to the problem's interfaces, by exploiting the discontinuities in the shape functions or their gradient that occur by construction across element edges. The generation of a matching mesh, however, is far from trivial when it comes to ensuring good element quality and it usually results in a cumbersome process. This is especially true in problems where geometries undergo consistent modifications, for example in optimization or transient analysis, where grid deterioration may require remeshing.

1.2. Extended and Generalized Finite Element Methods

To overcome the necessity of creating good quality matching grids, a new class of methods was developed to allow modeling discontinuities independently of the mesh, namely the eXtended Finite Element Method (XFEM) [18] or the Generalized Finite Element Method (GFEM) [11]. Inspired by the possibility of including *a priori* knowledge of the finite element space in the formulation [2], these approaches introduce enriched terms with convenient discontinuity properties, in addition to the standard FEM formulation, that are able to model jumps in the solution field or in its gradient. The discontinuous character of these enrichments depends on the choice of suitable shape functions, such as step or hat functions, in accordance with the type of discontinuity of the field. The enrichment functions are associated with enriched degrees of freedom, located at the existing nodes of the *representing elements*, *i.e.* the elements crossed by a discontinuity. The generalized finite element formulation takes the form [18]:

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} N_i(\mathbf{x})\mathbf{u}_i + \sum_{i=1}^{n_e} N_i(\mathbf{x}) \sum_{j=1}^{n_f} \psi_{ij}(\mathbf{x})\boldsymbol{\alpha}_j \tag{1.3}$$

where $\boldsymbol{\alpha}_j$ is the j-th enriched degree of freedom and $\psi_{ij}(\mathbf{x})$ the ij-th enrichment function evaluated in $\mathbf{x}$. The enrichment functions are localized in the formulation by means of the standard FEM shape functions $N(\mathbf{x})$, associated with the subset of n_e enriched nodes, which form a partition of unity on the representing elements, as in Figure 1.1.

In the simplest version of this method, the geometry is approximated piecewise linearly and the shape functions $N(\mathbf{x})$ are chosen to be Lagrangian shape functions. The enrichment functions, instead, are constructed to represent suitable types of discontinuities or singularities: these can be, for instance, step or distance functions (Figure 1.2). In general, the enrichment functions of the XFEM are non-local by construction, as they do not vanish far from the representing elements.

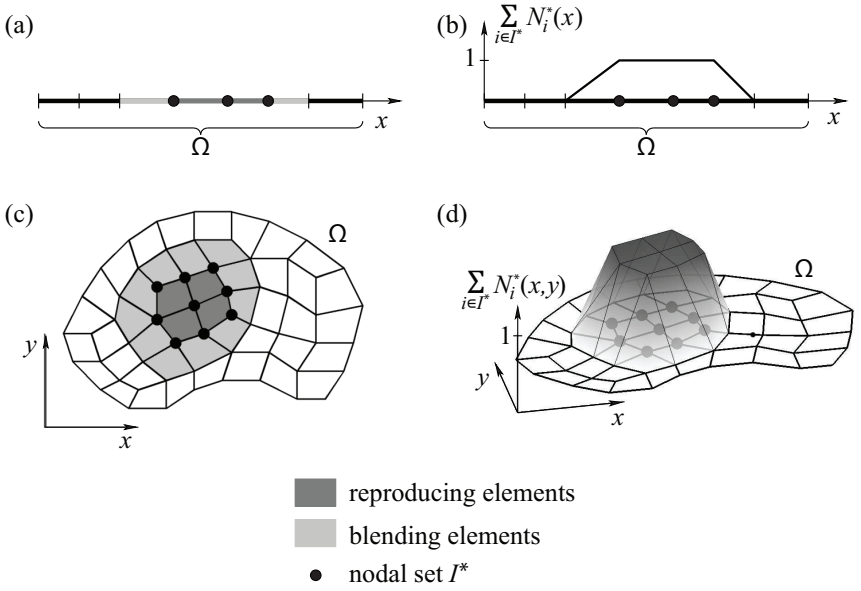


Figure 1.1: One- and two-dimensional example of discretized domains (*a, c*) and the associated shape functions (*b, d*) used to include the enriched terms [12]. The dark-colored elements are the representing elements, where the basis builds a partition of unity. In the blending elements, in light gray, the partition of unity is incomplete.

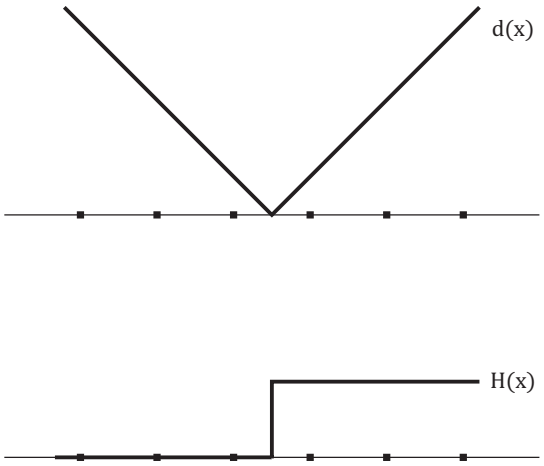


Figure 1.2: Examples of weak and strong enrichment functions, as a distance and a Heaviside function respectively. With this construction, the enrichment functions assume non-zero values in the elements that do not include the discontinuity.

The numerical integration is not performed directly on one element split by an interface: applying a conventional quadrature rule without accounting for discontinuities would in fact result in a decrease in accuracy. Instead, each representing element is decomposed into sub-domains, called integration elements; on each, a quadrature rule — commonly, Gauss quadrature — is applied.

Extended finite element methods have been successfully applied to a variety of problems, including propagating and branched cracks [9, 18], plates with holes and inclusions [5, 35] and materials with microstructures [19]. These examples showcase the capabilities of generalized finite element methods of modeling discontinuities and singularities, where matching meshes in the classical FEM can fail [11] or become too expensive to compute. The versatility of the XFEM lies in the possibility of selecting enrichment functions with suitable properties, with shapes that can model steps, kinks or crack tips, while fully exploiting the existing formulation of the classical FEM. The necessity of remeshing is avoided in the majority of cases.

While providing a valid approach to address the above-mentioned problems, the introduction of enrichments in the GFEM comes with some relevant drawbacks. One of the major issues arises in correspondence of the *blending elements*, the elements adjoining the representing elements, where only part of the nodes are enriched. There, as can be observed also in Figure 1.2, the enrichment functions are non-zero and the stan-

dard shape functions used to localize the enrichments, illustrated in Figure 1.1, do not complete a partition of unity [5]. This results in spurious terms in the field approximation, that can compromise the convergence properties of the solution. Usually, the accuracy in the blending elements is restored by means of corrective terms or by constructing enrichment functions that vanish outside the representing elements.

Secondly, handling boundary conditions is less straightforward when the interface is not matched by the elements' edges and, in particular, including Dirichlet conditions requires complex strategies. One of the most popular methods consists in incorporating essential boundary conditions into the problem's weak form by introducing Lagrange multipliers [20]. These techniques often lead to oscillations in the solution field and method's accuracy is then compromised.

An additional remark can be made by looking at the physical interpretation of the problem's degrees of freedom (DOFs). While in the standard FEM the DOFs represent the value of the field at the corresponding location, this duality with the physical field is lost in the GFEM, where nodes carry also information associated with non-conformed interfaces.

1.3. Interface- and Discontinuity-Enriched Finite Element Methods

The Interface-enriched Generalized Finite Element Method (IGFEM) [34] has been introduced to provide a solution to most of the issues originated by the generalized finite element method. As the name of this technique suggests, the enriched entities are not the nodes of the original mesh, but newly generated interface nodes, collocated at the intersection between the interface profile and the elements' edges.

A key feature of this method consists in the choice of the enrichment functions, which are local by construction, meaning that they vanish at the mesh nodes and in correspondence of all the edges which are not crossed by discontinuities. Consequently, the enrichments can be directly included into the FEM formulation, without resorting to the partition of unity property of the standard shape functions.

The basic formulation for the field approximation is the following:

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} N_i(\mathbf{x})\mathbf{u}_i + \sum_{j=1}^{n_{en}} s\psi_j(\mathbf{x})\boldsymbol{\alpha}_j \tag{1.4}$$

where $\boldsymbol{\alpha}_j$ is the j-th interface enriched node and $\psi_j(\mathbf{x})$ its associated enrichment function, able to capture weakly discontinuous fields. s is introduced as a scaling factor to improve the conditioning of the stiffness matrix when an enriched node is in close proximity of a mesh node, resulting otherwise in a very steep enrichment function.

The capabilities of this method have been extended to the treatment of strong discontinuities in the recently proposed Discontinuity-Enriched Finite Element Method (DE-FEM) [1], which introduces an additional term able to capture the C^{-1}-continuity in the solution field:

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} N_i(\mathbf{x})\mathbf{u}_i + \sum_{j=1}^{n_{we}} \psi_j(\mathbf{x})\boldsymbol{\alpha}_j + \sum_{k=1}^{n_{se}} \chi_k(\mathbf{x})\boldsymbol{\beta}_k \tag{1.5}$$

A strong enrichment function χ is constructed as a linear combination of Lagrangian shape functions on each side of the discontinuity, representing a jump in the field across the interface. An example of such an enrichment function is illustrated in Figure 1.3. The enriched degrees of freedom $\boldsymbol{\beta}_k$ assume the physical meaning of representing the magnitude of this jump at their respective enriched nodes.

The numerical integration, in both methods, is performed at a subdomain level with respect to the elements crossed by discontinuities. Soghrati [32] proposes a hierarchical method to both construct the enrichment functions and perform the integration, in the Hierarchical Interface-enriched FEM (HIFEM). The elements are recursively sub-triangulated from the first to the last discontinuity, as illustrated in Figure 1.4. Conversely, the quadrature points are first mapped onto the local coordinate system of the last hierarchical integration element and then inversely mapped until the root element, where the integration is performed.

Both IGFEM and DE-FEM successfully address the main deficiencies of the GFEM. The local construction of enrichment functions eliminates the necessity to utilize a partition-of-unity technique to localize the enrichments and, at the same time, solves the issues encountered in blending elements. Additionally, the standard degrees of freedom maintain the physical meaning of being equal to the value of the field at the mesh nodes. Hence, Dirichlet boundary conditions can be incorporated in a straightforward manner, without the need of more complex techniques requiring, for instance, Lagrange multipliers or penalty terms [1].

An observation about mesh-independence can be drawn from the formulation for IGFEM and DE-FEM here illustrated. In general, these methods allow modeling discontinuities independently of the chosen grid.

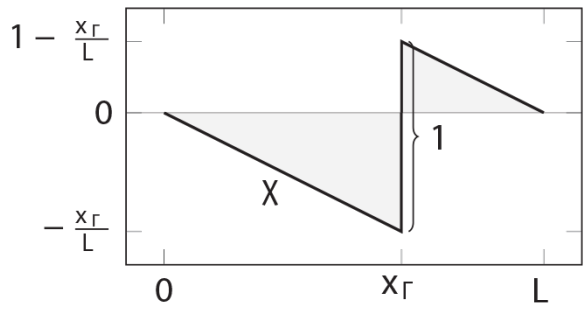


Figure 1.3: Example of DE-FEM strong enrichment function for a one-dimensional element [1], displaying a unit jump in correspondence of the strong discontinuity at x_Γ.

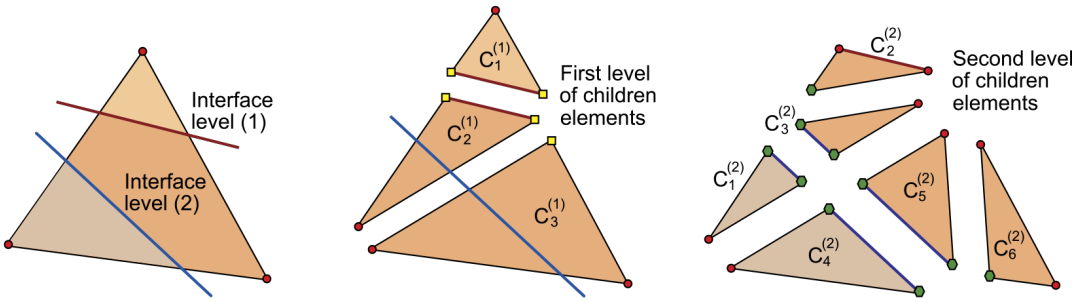


Figure 1.4: Hierarchical subtriangulation of a parent element into integration elements for multiple interfaces [32].

However, the use of isoparametric approximations with low-order Lagrangian shape functions introduces a discrepancy between computational and exact geometry: poor quality in the approximation, in particular in the linear case, still requires mesh refinements.

1.4. Finite Element Methods based on exact geometries

The capability of generalized finite element methods to represent mesh-independent discontinuities suggests the possibility to construct enrichment functions that can fit the exact interface geometries when these are nonlinear. This idea is not new to the world of finite elements: the literature offers multiple examples of methods aimed at creating high fidelity geometric models, ranging from higher order polynomial to spline boundary descriptions. A review of some outstanding methods will be illustrated in the next paragraphs.

1.4.1. The *p*-version of the Finite Element Method

In the *p*-version of the Finite Element Method, also known as *p*-FEM, higher-order polynomials are used to perform the analysis. In the isoparametric *p*-FEM, in particular, the same parametrization is utilized for both geometry and solution field. In first-order elements, this leads to a polygonal geometry, while, in the case of second-order elements, a parabolic interpolation of the edges is obtained. When a different parametrization is used for boundaries, this interpolation method leads to a misrepresentation of the geometries, even if higher-order functions are used [37]. Along with the increased accuracy in modeling the solution field, it becomes then necessary to represent arbitrary boundaries with more precision, without having to refine the mesh further.

Szabo' *et al.* [37] proposed a method to apply an arbitrary parametric description of the element boundaries, as a more accurate alternative to isoparametric *p*-FEM. This parametrization is introduced into the finite element formulation by the *blending function method*. According to this technique, the standard FEM mapping is augmented by a term that expresses the difference between the curved parametrization and the associated element edge, that could be, for instance, straight. This term is multiplied by a blending function, that prevents the mapping of opposite edges from affecting each other, by assuming unit values on its corresponding edge and vanishing at the facing side.

For the simple example illustrated in Figure 1.5, the formulation for a regular quadrilateral is augmented

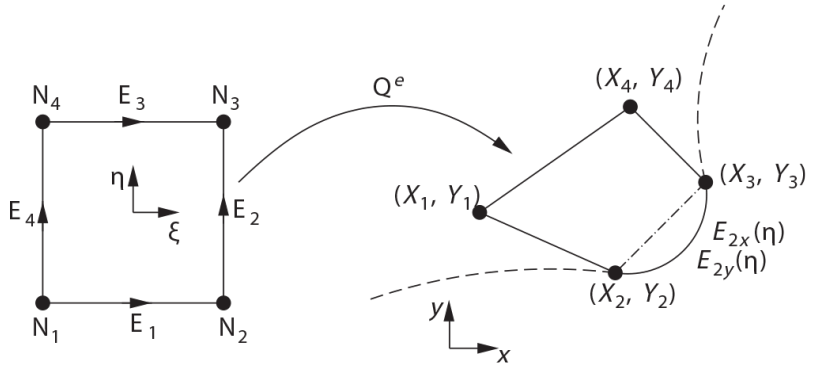


Figure 1.5: Blending function method, illustrated for a quadrilateral element with one curved edge [37].

by a term depending on the parametrization of the curved edge E_2:

$$\mathbf{x} = \sum_{i=1}^{4} N_i(\xi,\eta)\mathbf{X}_i + \left(\mathbf{E}_2(\eta) - \frac{1-\eta}{2}\mathbf{X}_2 - \frac{1+\eta}{2}\mathbf{X}_3\right)\frac{1+\xi}{2} \tag{1.6}$$

The term with the curvilinear description of edge E_2, added to the standard Lagrangian shape functions, is blended by $\frac{1+\xi}{2}$: this term goes to zero at $\xi = -1$, leaving the opposite edge E_4 unaffected.

The blending function method, from a case as simple as the one in equation 1.6 and Figure 1.5, can be extended to multiple parametric edges in a straightforward manner. Furthermore, due to the generality of the formulation, it can be applied to arbitrary parametrizations of the edges and it is suitable for higher-order elements, as in the focus of the authors' work.

1.4.2. Isogeometric Analysis

Differently from the blending function method, IsoGeometric Analysis (IGA) was conceived to handle a specific class of parametric curves called splines. These constitute geometries in Computer-Aided Design (CAD) and are constructed from basis splines, or B-splines. B-spline basis functions are piecewise polynomials in parametric coordinates, recursively defined from order zero according to the Cox-de Boor formula [23]:

$$\begin{aligned} N_{i,0}(s) &= \begin{cases} 1 & \text{if } s_i \le s < s_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(s) &= \frac{s - s_i}{s_{i+p} - s_i} N_{i,p-1}(s) + \frac{s_{i+p+1} - s}{s_{i+p+1} - s_{i+1}} N_{i+1,p-1}(s) \end{aligned} \tag{1.7}$$

where $N_{i,p}(s)$ is the i-th basis function of order p, evaluated at the parametric coordinate s. A predefined set of coordinates s_i, known as *knots*, delimits the intervals for the definition of the polynomial basis.

From these basis functions, B-spline curves are constructed in the physical space as:

$$\mathbf{x} = \mathbf{c}(s) = \sum_{i=1}^{n} N_{i,p}(s)\mathbf{b}_i \tag{1.8}$$

Here, a set of coordinates $\mathbf{b}_i$ are used as degrees of freedom to control the curve, hence the name of *control points*. The B-spline basis functions behave as blending functions, by merging the contribution of their associated control points within the curve.

B-splines have several limitations when it comes to modeling higher–order curves; for example, they cannot represent conic sections exactly. For this reason, computer-aided design is generally based on Non-Uniform Rational B-Splines, commonly known as NURBS. As the name suggests, these splines are constructed as rational functions of B-splines, for which non-uniform spacing between knots is allowed. NURBS basis functions are expressed as:

$$R_{i,p}(s) = \frac{N_{i,p}(s)\, w_i}{\sum_{j=1}^{n} N_{j,p}(s)\, w_j} \tag{1.9}$$

being w_i the weight factor associated to the i-th B-spline basis function.

From these basis functions, NURBS can be constructed in the physical space with a simple mapping, similar to the one applied for B-splines, by means of control points $\mathbf{b}_i$. Such a curve is expressed as:

$$\mathbf{x} = \mathbf{c}(s) = \sum_{i=1}^{n} R_{i,p}(s)\mathbf{b}_i \tag{1.10}$$

The formulation illustrated above for curves finds a straightforward extension to the representation of spline-based surfaces and volumes. For NURBS surfaces, for example, the basis is expressed as

$$R_{ij,pq}(s,t) = \frac{N_{i,p}(s)M_{j,q}(t)w_{ij}}{\sum_{k=1}^{n}\sum_{l=1}^{m} N_{k,p}(s)M_{l,q}(t)w_{kl}} \tag{1.11}$$

and the surface itself is constructed, again from a set of control points, according to:

$$\mathbf{x} = \mathbf{c}(s,t) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij,pq}(s,t)\mathbf{b}_{ij} \tag{1.12}$$

The spline-based models used for design are considered exact geometries, as they are meant to be high-fidelity representations of the shapes of real objects. Engineering analysis, on the other side, requires the discretization of the domain into finite elements, where the approximation of the solution field is computed. The incongruity between computational and design geometries represents a source of error, on top of the numerical and field approximation errors, that could be completely eliminated by carrying out finite element analysis on the exact shapes available *a priori*.

The wish to integrate CAD with FEA motivated the development of isogeometric analysis [8, 14], where the analysis is performed on the exact spline-based geometries used for the design. The solution field is modeled by means of the same NURBS basis, following an isoparametric approach that can compute high-order fields with great accuracy.

In isogeometric analysis, the domain is still subdivided into elements, where not only the boundary elements but also the internal ones are described by means of splines. The mesh nodes are located in correspondence of the points mapped by the knots defining the NURBS geometry and the field is interpolated by means of spline basis functions. In particular, in the case of a 1D problem, the field is expressed as [8]:

$$u(x) \approx u^h(x(s)) = \sum_{i=1}^{n} R_i(s)\mathbf{b}_i \tag{1.13}$$

where the NURBS basis functions $R_i(s)$ interpolate the values of the control variables $\mathbf{b}_i$. These can be assimilated to the concept of control points, which is also evident by comparing equations 1.13 and 1.10.

The choice of non-interpolatory degrees of freedom results in the impossibility to directly associate the control variables with the field value at the mesh nodes. This leads to several complications when it comes to handling essential boundary conditions [8], as interpolating the perscribed Dirichlet condition by the control variables often yields insufficient accuracy. In that case, fitting algorithms or a weak enforcement of the condition with additional corrective terms are applied. However, the enforcement of Dirichlet boundary conditions is never exact.

Broad research has involved isogeometric analysis, proving its great capabilities to model complex fields and geometries. The literature reports a wide variety of examples, including linear and nonlinear structural problems [8, 14], fluid-dynamics and fluid-structure interaction [4, 8, 14], and the study of vibrations [7, 8]. All these studies report an increase in the accuracy in computing the numerical solution, in particular comparing higher–order NURBS with p-FEM. The variation diminishing property of NURBS, for which higher–order curves tend to a smoother behavior, prevents the formation of the oscillations which often characterize high-order polynomial interpolation [14]. Several advantages are also presented by the possibility of increasing the curves' order or refining the mesh — for example, by knot insertion (see Appendix A, Section A.1.3) — without any modification to the exact geometry.

Despite its resonance in the academic world, the isogeometric approach has not gained popularity in industrial and commercial applications. One major reason is its full incompatibility with the standard finite-element formulation of most existing codes [16]. Moreover, its high computational cost does not make it an attractive alternative to the classical FEM, even in the case of very refined meshes.

1.4.3. The NURBS-Enhanced Finite Element Method

The NURBS-Enhanced Finite Element Method (NEFEM) introduced by Sevilla *et al.* [25, 26] abandons the isoparametric idea of isogeometric analysis, by applying a spline-based geometrical representation while maintaining the Cartesian shape functions of the standard FEM. A NURBS-enhanced geometric mapping is applied to boundary elements, generally characterized by one NURBS edge, while inner elements are treated as in the standard FEM.

The theory of the NEFEM is presented for triangular meshes, for which two different mappings are proposed [25], from either a rectangular or a triangular parametric domain. These will be briefly summarized in the next paragraphs, for a general triangular element of nodes $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, with a NURBS edge between nodes $\mathbf{x}_1$ and $\mathbf{x}_2$, as illustrated in Figures 1.6 and 1.7.

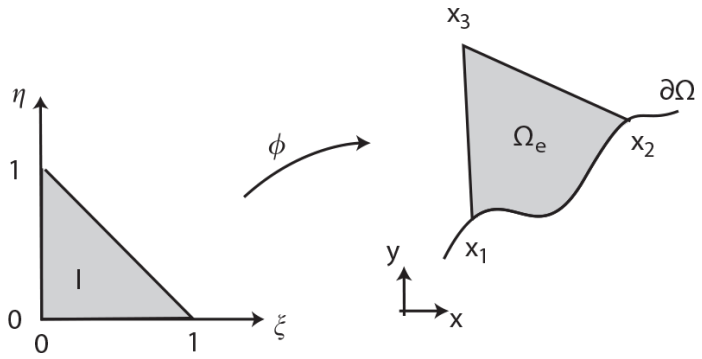


Figure 1.6: NEFEM mapping of a NURBS-edged triangular element from a triangular reference domain [25].

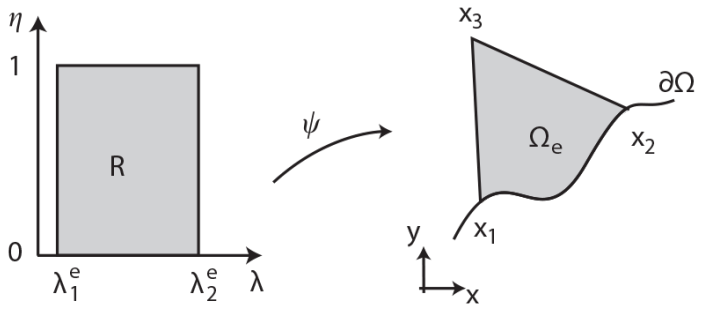


Figure 1.7: NEFEM mapping of a NURBS-edged triangular element from a rectangular reference domain [25].

The most intuitive method consists in mapping the NURBS-edged triangle from a triangular parametric domain. The transformation is defined as follows:

$$\begin{aligned} &\phi : T \to \Omega_e \\ &(\xi,\eta) \mapsto \phi(\xi,\eta) := \frac{1-\xi-\eta}{1-\xi}\mathbf{c}(\xi) + \frac{\xi\eta}{1-\xi}\mathbf{x}_2 + \eta\mathbf{x}_3 \end{aligned} \tag{1.14}$$

where $\mathbf{c}(\xi)$ is the parametrization of the spline.

Alternatively, the NURBS-enhanced element can be mapped from a parametric rectangle, according to:

$$\begin{aligned} &\psi : R \to \Omega_e \\ &(\xi,\eta) \mapsto \phi(\lambda,\eta) := (1-\eta)\mathbf{c}(\lambda) + \eta\mathbf{x}_3 \end{aligned} \tag{1.15}$$

where the bottom edge of the rectangle is mapped directly onto the NURBS edge and the top is collapsed onto node $\mathbf{x}_3$, opposite to the curve.

The use of a rectangular reference domain is demonstrated to bring numerous advantages. As can be noticed by comparing the mappings in 1.14 and 1.15, the rectangular case is simpler and the coordinate ξ, in particular, results only in the parametrization of the NURBS edge. Moreover, the linearity in the parametrization in the interior direction suggests the possibility of decoupling the number of quadrature points in the

two directions ξ and η, considering that the exact integration along the latter can be performed very cheaply [25]. Therefore, fewer integration points are required, compared to a triangular quadrature scheme. Based on these advantages, a rectangular parametric domain is generally preferred, which results in a convenient choice also when the element presents two or three NURBS edges.

In combination with high-order shape functions, the NEFEM has been proven effective capabilities to simulate problems with curved boundaries with minimal mesh refinement. Examples include the Poisson problem [27] and electromagnetic scattering [25, 27].

With the NEFEM, the significant drawbacks of the isogeometric analysis are substantially solved. First of all, NURBS are introduced at the mere level of the geometric mapping, preserving therefore the majority of the standard finite element formulation. This also implies that the spline-based parametrization is only active at the level of boundary elements, which contains the additional computational cost associated with NURBS drastically.

However, one evident drawback that appears from this formulation is the complexity of mesh generation. To the standard FEM process of creating a good-quality, conforming grid, an additional burden is brought by accounting for NURBS-based boundaries. Although this issue does not emerge as a difficulty in the fixed-boundary problems addressed in the NEFEM literature, meshing in cases of moving boundaries would become a challenging task.

1.5. NURBS-enhanced Generalized Finite Element Methods

The possibility of representing interfaces independently of the mesh offered by generalized finite element methods has inspired several researchers to use accurate geometries for the enrichments.

A combination of the XFEM and the IGA is presented by De Luycker *et al.* [10] and Ghorashi *et al.* [13], as the so-called eXtended IsoGeometric Analysis (X-IGA). This method yields very accurate results in fracture problems, where crack tip fields can be better predicted, without the need of remeshing for propagating fractures.

Legrain [16] introduces the NURBS-enhanced approach presented by Sevilla *et al.* [27] in the XFEM. The NURBS-enhanced XFEM results to be an accurate method when applied to structural problems, more efficient than isogeometric analysis and more versatile than the blending function method in handling discontinuities. Numerical examples carried out with high order field approximations, in conjunction with the spline-based geometric model, show the capability of the approach of yielding accurate results even for large elements.

Recent publications present the application of NURBS enhancements to interface-enriched finite element methods. The approach followed by Tan *et al.* [38] and Safdari *et al.* [24], called NURBS-enhanced IGFEM (NIGFEM), utilizes NURBS for both the geometric mapping and the construction of spline-based enrichment functions, with a methodology that resembles the isogeometric analysis, although limited to the enrichments. Therefore, the enrichment functions are NURBS basis functions, while the enriched degrees of freedom are the control variables associated with the interface.

An alternative method is presented by Soghrati *et al.* [33], who follow the approach of the NEFEM when constructing integration elements. The use of NURBS is restricted to the geometric mapping, which is performed from a rectangular reference domain as in Sevilla *et al.* [25], on which standard Lagrangian shape functions are defined for approximating the field. The enriched degrees of freedom are associated with the new nodes created at the intersection between interfaces and mesh lines, like in the classical IGFEM.

Both methods show outstanding performance, limited to the study of weak discontinuities, in structural and thermal problems with composite materials [24, 33] and in the thermal study of microvascular systems [38]. The latest publication in this field also illustrates the application to shape optimization for materials with inclusions [21]. To my knowledge, no research has been published for what concerns the study of problems with strong discontinuities, in the context of NURBS-enhanced IGFEM or DE-FEM.

1.6. Objectives

The main objective of this thesis is to propose a NURBS-based enhancement to the Discontinuity-Enriched Finite Element Method, in order to model two-dimensional problems with weak and strong discontinuities with arbitrary smooth shapes. This method potentially allows more freedom in modeling curved boundaries, by maintaining exact geometries throughout the analysis. In this way, the error originated by the geometric discretization can be eliminated and compatibility between design and computational geometries is preserved.

To this purpose, a suitable NURBS-based analysis technique will be selected and an appropriate implementation for the group's finite element library, Hybrida, will be proposed. Preliminary studies will be carried out to validate the method, in particular for the recovery of constant states of stress and the satisfaction of optimal convergence properties. Moreover, the equivalence of standard and NURBS-enhanced DE-FEM in the study of problems with straight discontinuities, in terms of accuracy, will be verified.

Subsequently, the study of diverse structural problems with curved discontinuities will be aimed at assessing the accuracy and robustness of the NURBS-enhanced DE-FEM, also in comparison with the standard FEM and DE-FEM. One main focus point will be the extension of the proposed technique to a novel study, for the modeling of strong discontinuities in the context of fracture mechanics.

Overall, this project aims at providing further insight into the potential and the limitations of adopting such NURBS-enhanced formulation in combination with the DE-FEM, for the study of problems with arbitrarily-shaped discontinuities. In particular, the convenience of utilizing a NURBS-based approach against a linearly approximated one will be discussed. This evaluation will consider factors such as accuracy and computational efficiency. Some recommendations will be also formulated, to provide an indication of the most suitable problems for this method and of open directions for future improvement.

1.7. Thesis outline

After this introduction to the project's topic and goals, the rest of this thesis is structured as follows.

In Chapter 2, the NURBS-enhanced DE-FEM for the treatment of weak and strong discontinuities is presented, including its theoretical formulation and in the details regarding its implementation. The method is validated with benchmark problems, in comparison with the results obtained via the standard DE-FEM. Subsequently, the study is extended to several applications of structural problems with curved inclusions and cracks.

Chapter 3 contains a reflection about the work that led to this thesis. Particular focus is dedicated to the methodology followed and the personal challenges faced during the development of the project.

A more detailed description of spline-based geometric operations, their implementation and the validation process is included in the Appendices.

2

A NURBS-enhanced Discontinuity-Enriched Finite Element Method

2.1. Formulation and implementation

2.1.1. Problem description

In this study, a NURBS-based enhancement to the DE-FEM is proposed for two-dimensional discontinuous problems. In particular, the focus of the applications will be on boundary value problems for linear elasticity, for which the derivation of the FEM formulation will be illustrated in this section, starting from the equilibrium equations in strong form. In a domain V with boundary S, the displacement $\mathbf{u} : \bar{\Omega} \rightarrow \mathbb{R}^2$ is to be found such that:

$$\mathbf{D}^T \boldsymbol{\sigma} + \bar{\mathbf{f}} = \mathbf{0} \quad \text{in } V \tag{2.1}$$

$$\mathbf{N}^T \boldsymbol{\sigma} = \bar{\mathbf{t}} \quad \text{on } S_\sigma \tag{2.2}$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } S_u \tag{2.3}$$

where $\boldsymbol{\sigma} : \Omega \rightarrow \mathbb{R}^2 \times \mathbb{R}^2$ is the Cauchy stress tensor and $\bar{\mathbf{f}} : \Omega \rightarrow \mathbb{R}^2$ the applied body force. The boundary conditions express, respectively, equilibrium and continuity at the boundaries where the traction $\bar{\mathbf{t}} : S_\sigma \rightarrow \mathbb{R}^2$ is prescribed (equation 2.2) and where the displacement $\bar{\mathbf{u}} : S_u \rightarrow \mathbb{R}^2$ is imposed (equation 2.3). S_σ and S_u are the portions of the boundary of V where Neumann and Dirichlet boundary conditions are imposed, respectively.

In order to find an approximate solution to this boundary value problem, the Galerkin method is applied, which requires first the definition of the problem in its weak, or variational, form. Upon defining the space of the admissible solutions

$$\mathscr{U} = \{\mathbf{u} \in \mathscr{W}, \mathbf{u} = \bar{\mathbf{u}} \text{ on } S_u\} \tag{2.4}$$

where $\mathscr{W}$ is a space constructed according to the regularity of $\mathbf{u}$, a second space is defined for the weight functions, so that a homogeneous condition is prescribed on the Dirichlet boundary:

$$\mathscr{V} = \{\mathbf{v} \in \mathscr{W}, \mathbf{v} = \mathbf{0} \text{ on } S_u\} \tag{2.5}$$

The weak form for the boundary value problem, obtained by multiplying equation 2.1 by the weight function and integrating, is then: Find $\mathbf{u} \in \mathscr{U}$ such that:

$$\int_V \boldsymbol{\sigma} : \boldsymbol{\varepsilon}(\mathbf{v}) dV = \int_V \bar{\mathbf{f}} \cdot \mathbf{v} dV + \int_{S_\sigma} \bar{\mathbf{t}} \cdot \mathbf{v} dS \qquad \forall \mathbf{v} \in \mathscr{V} \tag{2.6}$$

with the strain $\boldsymbol{\varepsilon}(\mathbf{v})$ being defined as:

$$\boldsymbol{\varepsilon}(\mathbf{v}) = \mathbf{D}\mathbf{v} = \frac{1}{2}(\nabla \mathbf{v} + (\nabla \mathbf{v})^T) \tag{2.7}$$

After substituting into equation 2.6 the FEM approximation $\mathbf{u} = \mathbf{Fq}$ for the solution field $\mathbf{u}$ and its dual for the weight function $\mathbf{v}$, simplifying the arbitrary terms depending on $\mathbf{v}$ yields:

$$\int_V \mathbf{B}^T \mathbf{HBq} dV = \int_V \mathbf{F}^T \bar{\mathbf{f}} dV + \int_{S_\sigma} \mathbf{F}^T \bar{\mathbf{t}} dS \tag{2.8}$$

often written in compact form by introducing a global stiffness matrix $\mathbf{K}$ and a global force vector $\mathbf{f}$:

$$\mathbf{Kq} = \mathbf{f} \tag{2.9}$$

The system in equation 2.9 is solved for the vector of the nodal values of the solution field $\mathbf{q}$, by inverting the global stiffness matrix.

2.1.2. DE-FEM formulation

The Discontinuity-Enriched Finite Element Method, as illustrated in Chapter 1, is capable of representing both weak and strong discontinuities independently of the mesh, by assigning enriched degrees of freedom to newly constructed nodes at the intersection between discontinuities and element edges [1].

The initialization of a problem includes the generation of a non-conforming mesh, here assumed to consist of triangular elements, and the construction of the discontinuities, either explicitly or implicitly. In the first case, the equation of a discontinuity is given in cartesian or parametric coordinates, while in the latter it is expressed by means of a level-set function. Before solving the problem, a series of geometric operations is performed by looping over each discontinuity, in order to find intersections with the elements. Enriched nodes are created at those locations and the parent elements are subtriangulated in a hierarchical fashion.

For the DE-FEM approximation of the solution field [1]:

$$\mathbf{u}(\mathbf{x}) \approx \mathbf{u}^h(\mathbf{x}) = \sum_{i=1}^{n} N_i(\mathbf{x})\mathbf{u}_i + \sum_{i=1}^{n_{we}} \psi_i(\mathbf{x})\boldsymbol{\alpha}_i + \sum_{i=1}^{n_{se}} \chi_i(\mathbf{x})\boldsymbol{\beta}_i \ , \tag{2.10}$$

Lagrangian shape functions $N_i(\mathbf{x})$ are constructed for the standard FEM component of the field. In the standard DE-FEM, the enrichment functions $\psi_i(\mathbf{x})$ and $\chi_i(\mathbf{x})$ are linear combinations of Lagrangian shape functions, locally constructed on the representing elements, such that kinks or jumps are incorporated at the discontinuity location. Eventually, the contribution of each element to the system matrices is computed by Gauss quadrature and the system is solved for the degrees of freedom at original mesh and enriched nodes.

One limitation of the standard DE-FEM lies in the choice of simple linear combinations of isoparametric Lagrangian shape functions to construct enrichment functions. In the most common case of linear shape functions, this implies that geometries can be at most approximated by line segments, which contributes to the overall approximation error with additional geometric error. However, the DE-FEM formulation, as in equation 2.10, can accommodate a wider choice of enrichment functions and, in particular, functions that can preserve the exact geometries of discontinuities. To this purpose, a NURBS-based enhancement is proposed, which allows more accuracy and flexibility in the treatment of discontinuities with arbitrary shapes.

The implementation of NURBS-based enhancements presents the advantage of being fully compatible with the existing DE-FEM framework: the structure of the method previously illustrated is substantially unchanged. NURBS-based discontinuities are defined explicitly, from parametric coordinates, according to the formulation illustrated in equations 1.9 and 1.10. This construction procedure is represented in Figure 2.1, for the case of a NURBS-based circle. As can be observed from both this illustration and from the mathematical formulation, the creation of a NURBS-based discontinuity requires the definition of a knot vector, weights and control points, in a number that satisfies the conditions for the desired polynomial order and continuity properties (see Appendix A).

The main differences with respect to the standard DE-FEM appear at the level of geometrical operations and coordinate mapping. In fact, the way these are performed undergoes relevant modifications, in order to construct NURBS-based integration elements and enrichment functions $\psi_i(\mathbf{x})$ and $\chi_i(\mathbf{x})$. These will be the focus of the following sections.

2.1.3. Construction of integration elements

Following the creation of the NURBS-based discontinuities, a geometric engine loops over the elements, to create discontinuity nodes and integration elements. When searching for the intersected elements, a preliminary check excludes from further investigation the elements lying outside the discontinuity's bounding box,

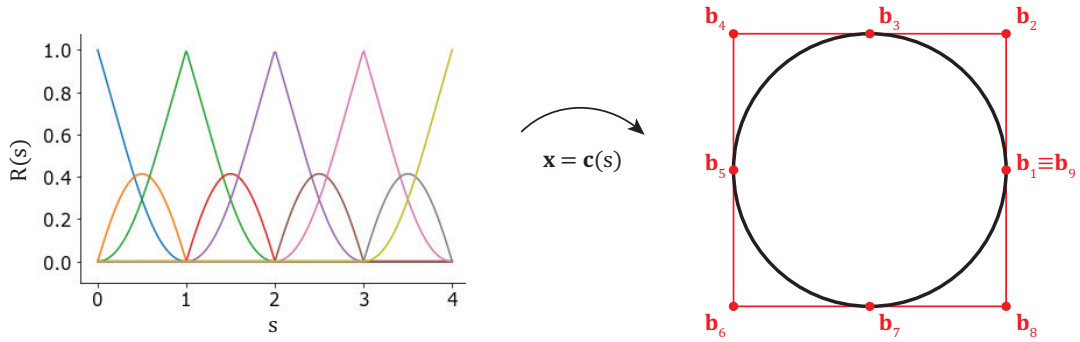


Figure 2.1: Example of construction of a NURBS-based interface, from the basis functions in parametric coordinates to the curve in the physical space. In the case illustrated, representing a second-order NURBS-based circle, the knot vector is $[0,0,0,1,1,2,2,3,3,4,4,4]$. The weights are equal to $1/\sqrt{2}$ for points with even index and have unit value for the odd ones.

being in 2D the rectangle circumscribing the curve. If the condition is satisfied, the level-set function, defined as a signed distance function, is computed at every node of the element. In particular, the sign of the level-set function is checked, which indicates the side of a discontinuity to which a point belongs. Interface elements experience a variation in sign among nodes.

When an interface element is encountered, intersection points are searched for in every element edge with discordant nodes. This method is based on the assumption, holding in general for the DE-FEM, that every edge is crossed not more than once by a discontinuity. Intersection points are found by a Newton-Raphson solver, which computes the parametric coordinate on the discontinuity for which the equation of a specified mesh segment is satisfied.

In correspondence of the intersection points between discontinuities and element edges, new nodes are created: one single node for weak discontinuities and double nodes on the edges of strong discontinuities. Crack tips represent an exception: in those cases, the location for the tip node is found by checking whether the end point of the discontinuity is contained within an element where only one edge is intersected.

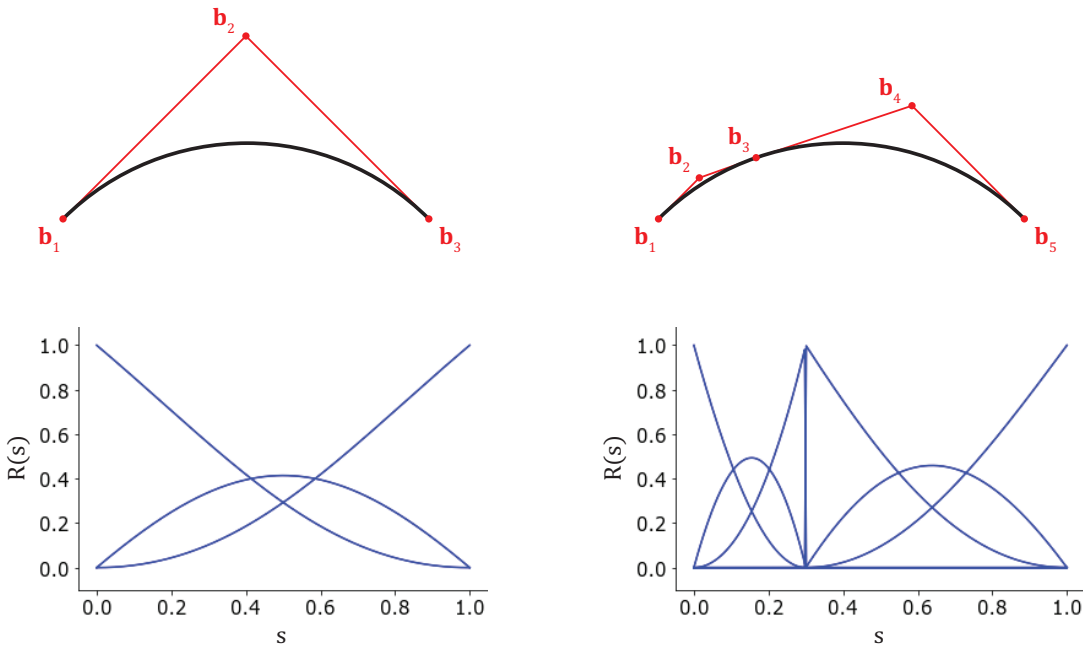
In addition to the creation of enriched nodes, the NURBS segment contained within each element is extracted [38]. This step, not required in linear approximations, where an interface segment is uniquely defined by two nodes, is necessary to be able to utilize information associated with the spline segment in successive stages of the analysis. In a NURBS curve, each point results from the contribution of several basis functions, blended within the range of a few control points. Conversely, when storing data associated with the segment of discontinuity enclosed by one element, it is convenient to describe that portion of the curve by only self-contained information. This would mean, in practice, reducing the continuity of the spline parametrization across the enriched nodes, without modifying the actual geometry of the interface, so that external data do not affect the representation of the inner segment and, therefore, can be neglected. Such an operation can be performed by means of the knot insertion algorithm [23, 38], which introduces knots at a given location with sufficient multiplicity to achieve C^{-1}-continuity. An illustration of the effects of knot insertion on the NURBS parametrization is shown in Figure 2.2, while the algorithm is described with more detail in Appendix A (Section A.1.3).

Upon the creation of enriched nodes, every representing element is split into children elements by means of Delaunay triangulation. Examples of subtriangulation are illustrated in Figure 2.3, for both elements internal to a discontinuity and containing one of the ends. The data describing the discontinuity segment are associated with the children elements which share the spline edge.

2.1.4. NURBS-based enrichment functions

Once integration elements are created, both standard shape and enrichment functions can be constructed, by following a hierarchical approach. For the elements at the root level and for the children elements with only one node on the discontinuity, standard Lagrangian shape functions are used. Instead, for the integration elements that share a NURBS edge, NURBS-based enrichment functions are required.

The literature offers two main approaches to construct NURBS-enhanced shape functions: either the use of NURBS basis functions [24, 38] or the construction of standard Lagrangian shape functions, which are subsequently mapped onto a curved element via a NURBS-enhanced mapping [26, 33]. For the NURBS-



(a) (b)

Figure 2.2: Effect of knot insertion on the continuity of a NURBS's parametrization. To the initial curve, illustrated in Figure 2.2a together with its control points and shape functions, three coincident knots are inserted at $s = 0.3$. Figure 2.2b shows the change in control points and parametric construction, in particular the loss of continuity across the added knots, while the shape of the NURBS curve undergoes no variation.

Figure 2.3: Examples of subtriangulation of elements crossed by NURBS-based discontinuities into integration elements: on the left, an internal discontinuity element is illustrated, while the right image represents the case of an element containing a crack tip. The darker color indicates the integration elements sharing a NURBS edge, that will need an *ad hoc* construction for the enrichment functions.

enhanced DE-FEM, the second approach is preferable: it is not only consistent with the order and type of shape functions associated to the non-enriched degrees of freedom, but it also maintains the simplicity in the formulation that is sought for in the DE-FEM. Additionally, the use of Lagrangian shape functions is compatible with the assignment of degrees of freedom to the intersection nodes placed on the discontinuity, as in the standard DE-FEM, instead of imposing the choice of non-interpolatory control variables as degrees of freedom.

The enrichment functions are constructed on a square reference domain, which requires fewer integration points than the choice of a triangular reference domain [26], at least for elements with one or two NURBS edges. These are expected to represent the vast majority of cases and, for this study, only integration elements with one NURBS edge are assumed.

In the NURBS-enhanced DE-FEM, the reference domain has been selected as the standard domain adopted for the definition of Gauss points, namely the reference square $[-1,1]\times[-1,1]$. This not only eliminates a first unnecessary coordinate change, but also allows utilizing directly the shape functions for standard four-noded rectangular elements, which are defined in the same reference domain. The mapping takes then the form:

$$\boldsymbol{\Phi} = \boldsymbol{\Psi}\circ\boldsymbol{\Theta}: \quad (1-t)\mathbf{c}(s) + t\mathbf{x}_3 = \Big(1-\frac{1}{2}(1+\eta)\Big)\,\mathbf{c}\Big(s_a + \frac{s_b - s_a}{2}(1+\xi)\Big) + \frac{1}{2}(1+\eta)\mathbf{x}_3 \tag{2.11}$$

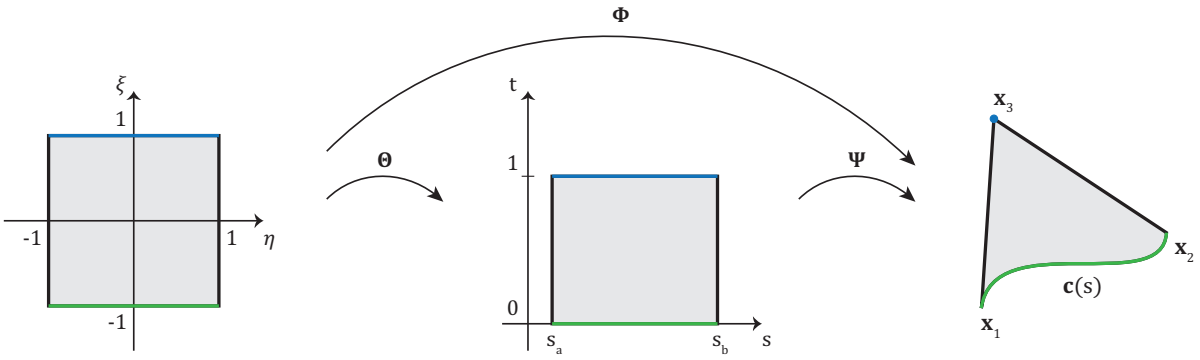


Figure 2.4: Mapping of NURBS-enhanced integration elements.

where $\mathbf{x} = \mathbf{c}(s)$ is the expression of the NURBS-based discontinuity, following the formulation in equations 1.9 and 1.10. In this mapping, the top edge of the square eventually collapses into the single vertex $\mathbf{x}_3$ in the triangular integration element, as highlighted in Figure 2.4. Hence, the shape function associated with that node will be obtained by the sum of the shape functions associated with the top nodes of the square reference domain [26]:

$$\begin{aligned} N_1^{\triangle}(\boldsymbol{\Phi}(\xi,\eta)) &= N_1^{\square}(\xi,\eta) &&= \frac{(1-\xi)(1-\eta)}{4}, \\ N_2^{\triangle}(\boldsymbol{\Phi}(\xi,\eta)) &= N_2^{\square}(\xi,\eta) &&= \frac{(1+\xi)(1-\eta)}{4}, \\ N_3^{\triangle}(\boldsymbol{\Phi}(\xi,\eta)) &= N_3^{\square}(\xi,\eta) + N_4^{\square}(\xi,\eta) = \frac{(1+\xi)(1+\eta)}{4} + \frac{(1-\xi)(1+\eta)}{4} &&= \frac{1+\eta}{2} \end{aligned} \tag{2.12}$$

The derivatives of these shape functions with respect to the coordinates (ξ,η) are computed trivially:

$$d\mathbf{N} = \frac{\partial \mathbf{N}^{\triangle}}{\partial \boldsymbol{\xi}} = \frac{1}{4}\begin{bmatrix} -(1-\eta) & -(1-\xi) \\ (1-\eta) & -(1+\xi) \\ 0 & 2 \end{bmatrix} \tag{2.13}$$

2.1.5. Numerical integration

The finite-element approximation of the solution field allows reducing the problem to a system of algebraic equations, for which the system matrices are found according to equation 2.8. In particular, this phase requires integrating regular and enrichment shape functions over the integration elements, which is performed by numerical quadrature.

In general, it is not possible to integrate NURBS exactly by Gauss quadrature [24] and, to the present, the numerical integration of these curves is an open field of research. Techniques have been developed which are specifically suited for spline-based geometries: one example is represented by the *half-point rule* [15], which makes more efficient use of the smoothness of splines across knot spans. This allows placing a lower number of integration points than with Gauss quadrature, down to one integration point every two B-spline basis functions. The efficiency of this method, however, depends on higher-order continuity across knots: when this hypothesis is not verified, Gauss integration is still an optimal quadrature rule [15]. The proposed half-point rule has been only tested for B-splines, not NURBS [38], and it specifically aims at improving integration in isogeometric analysis, where elements are exactly coincident to knot spans and where NURBS-based geometries constitute the entirety of the domain.

In the case of a NURBS-enhanced finite element methods, the mismatch between element boundaries and knot spans, the decreased continuity of NURBS across element nodes and possibly inside elements, and the need to still maintain Gauss integration for standard elements do not favour the adoption of a different quadrature technique for only a limited number of elements with spline-based geometries. In fact, higher-order Gauss-Legendre quadrature rules still represent a convenient method, in terms of both accuracy and efficiency [25], and are widely used in the literature for the integration of NURBS [8, 26, 33, 38].

Gauss points (ξ_i, η_i) for NURBS-enhanced elements are mapped from their square reference element onto the integration elements via the mapping $\mathbf{\Phi}$. The choice of the reference domain results convenient for this operation, as Gauss points are generally defined within the range $[-1,1]$. Following this quadrature rule, the integral of a generic function $f(x,y)$ over an integration element Ω can be then approximated as [26]:

$$\int_\Omega f(x,y)\,dx\,dy = \int_R f(\mathbf{\Phi}(\xi,\eta))|\mathbf{J}_\mathbf{\Phi}|\,d\xi\,d\eta \approx \sum_i^{n_\xi}\sum_j^{n_\eta} \hat{w}_i\,\bar{w}_j\,f(\xi_i,\eta_j)|\mathbf{J}_\mathbf{\Phi}(\xi_i,\eta_j)| \tag{2.14}$$

where $|\mathbf{J}_\mathbf{\Phi}|$ is the determinant of the Jacobian of the transformation $\mathbf{\Phi}$ between reference and physical coordinates, simply computed by chain rule. The integration order in the two directions has been decoupled [28], based on considerations on the linearity of the mapping along η: hence $n_\xi \neq n_\eta$. This allows reducing the number of Gauss points considerably.

2.1.6. Comparison between standard and NURBS-enhanced DE-FEM

The NURBS-enhanced DE-FEM preserves all the key advantages of the standard DE-FEM formulation: the enrichments are local by construction, with enriched degrees of freedom assigned to discontinuity nodes. In particular for strong discontinuities, the interpretation of the enriched degrees of freedom as the extent of the crack opening is preserved. On the other hand, the original mesh nodes still satisfy the Kronecker delta property. With the introduction of NURBS-based enhancements, the DE-FEM formulation remains completely unchanged, while the only substantial differences lie in the nature of the geometric operations needed to construct integration elements and in the mapping of the enrichment functions.

The NURBS-based enhancements allow modeling exact geometries even for the coarsest meshes, meaning that no refinement is needed for the sole purpose of improving shape representation. However, this comes at the price of the increased complexity and computational cost of handling geometric operations for recursively defined parametric curves. A direct implication involves the need for suitable data structures to store NURBS information: this step is completely missing in the DE-FEM based on linear approximations, where every segment is uniquely defined by its bounding nodes. In the case of splines, instead, additional information needs to be associated with NURBS-based features, to keep track of the parametrization and avoid repeated mapping operations.

These issues are not only limited to preliminary geometric manipulation and analysis: spline geometries need a special treatment also in the postprocessing stage. In the case of linear approximations, the visualization of the solution field requires uniquely nodal data to be interpolated. Additional postprocessing is required in the case of discontinuity-enriched FEA, in order to incorporate the enriched contributions appropriately. Adopting spline geometries adds complexity to these operations, as nodal values are not sufficient for a correct representation of the boundaries. In general, common software for data visualization handles simple polygons with more ease; consequently, additional tasks for reconstructing NURBS geometries or sampling and interpolating values along the discontinuities may be required.

A drawback of the NURBS-enhanced method comes also with the computation of level-set functions. In the standard DE-FEM, these are already given in the initialization steps or have a trivial computation from the explicit definition of the discontinuities. When using NURBS, instead, level-set functions are known explicitly only if a cartesian expression for the discontinuity can be derived, which is not easy to perform in an automatic way. Otherwise, computing the value of a level-set function at a point requires more elaborate operations, including a minimization problem to find the closest point on the curve, from which the distance is calculated.

Because of the expensive computation of level-set functions and crossing points, a systematic search for intersections throughout all the elements of a domain is definitely not the most suitable approach for NURBS-based discontinuities. However, this is not required by the formulation either: in fact, when a spline-based discontinuity is present in the domain, it is given as a continuous curve and the intersected elements all belong to a connected set. This does not necessarily hold for implicitly defined discontinuities: in this case, the level-set function employed for their definition may become null at several locations of the domain, not necessarily connected to each other. This circumstance justifies a thorough search for intersected elements throughout the domain, which should rather be avoided otherwise.

Overall, the convenience and attractiveness of introducing spline-based modeling would depend on how the improved geometric representation can counterweight the drawbacks of a more complex and costly formulation and, in particular, on the actual possibilities of minimizing mesh refinement.

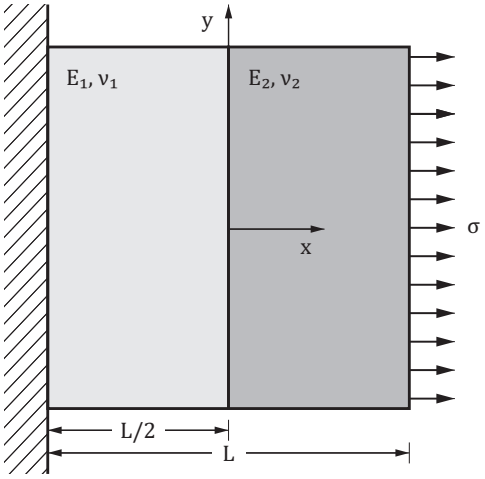


Figure 2.5: Bimaterial test: definitions for the boundary value problem.

2.2. Validation

2.2.1. Bimaterial test

A finite element method based on first-order shape functions is expected to recover a linear displacement field and the associated constant state of stress exactly, which is verified by means of a bimaterial test. The boundary value problem for this test has been defined for a square bimaterial block under tension, as illustrated in Figure 2.5. The hypothesis of plane strain is assumed in this and all the following problems. For what concerns the materials, the Young's modulus E_1 of the left side assumes a value of 10 Pa, while for the second material $E_2 = 1$ Pa. For both, the Poisson's ratio is null. The uniaxial stress σ has module 2 Pa.

For this problem, the linear displacement field is given by [22]:

$$\mathbf{u} = \begin{pmatrix} \frac{\sigma x}{E_1} + \frac{\sigma L}{2E_1} \\ 0 \end{pmatrix} \quad \text{for} -\frac{L}{2} \leq x \leq 0 \tag{2.15}$$

$$\mathbf{u} = \begin{pmatrix} \frac{\sigma x}{E_2} + \frac{\sigma L}{2E_1} \\ 0 \end{pmatrix} \quad \text{for} \quad 0 < x \leq \frac{L}{2} \tag{2.16}$$

In equations 2.15 and 2.16, L measures the edge of the square domain (here 2 m long), assuming that the interface is located at $x = 0$. No displacement is expected along the y direction, due to the null Poisson's ratio assigned to both materials. The stress is expected to have a constant value throughout the whole domain, equal to the applied traction.

As showed in Figure 2.7, the NURBS-enhanced DE-FEM can successfully recover constant states of stress (here 2 Pa) up to machine precision. Moreover, the representation (Figure 2.6) of the linear displacement field in x and the null displacement in y (with an error in the order of the numerical precision) reflect the exact solution. Tests run on various meshes show good performance for different refinements: in all cases, the constant state of stress in recovered and the same deformation field is computed.

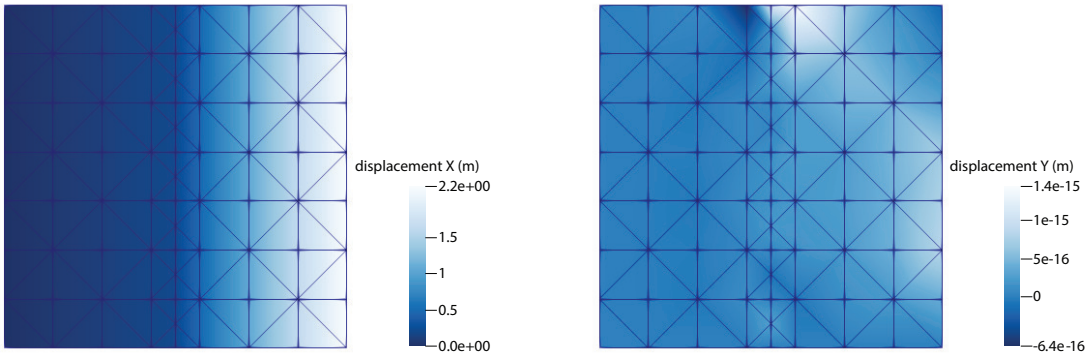
2.2.2. Convergence test

The accuracy and convergence properties of the NURBS-enhanced DE-FEM have been verified by means of two convergence tests, which assess the improvement rate in the error for successive mesh refinements.

Two measures of the error are monitored, namely the L_2-norm of the error (ϵ_{L_2}) and the error in the energy norm (ϵ_E). These are respectively defined as:

$$\epsilon_{L_2} = \sqrt{\int_\Omega ||\mathbf{u} - \mathbf{u}^h||^2 d\Omega} \tag{2.17}$$

$$\epsilon_E = \frac{||\mathbf{u} - \mathbf{u}^h||_{E(\Omega)}}{||\mathbf{u}||_{E(\Omega)}} = \frac{\sqrt{\int_\Omega (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h)^T \mathbf{H} (\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^h) d\Omega}}{\sqrt{\int_\Omega \boldsymbol{\varepsilon}^T \mathbf{H} \boldsymbol{\varepsilon} d\Omega}} \tag{2.18}$$



(a) (b)

Figure 2.6: Bimaterial test: linear x-displacement field (a) and null y-displacement field (b) for a coarse structured mesh. For the y-displacement, in particular, the field presents an error in the order of the numerical precision with respect to the exact zero of the analytical solution, as can be observed from the legend.

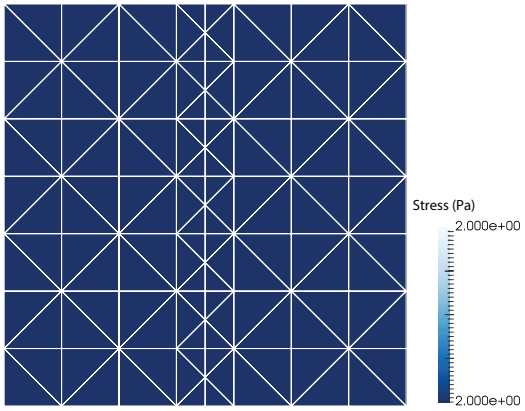


Figure 2.7: Bimaterial test: constant stress field.

The first convergence problem studied is similar to the boundary value problem studied for the bimaterial test under constant stress (see also Figure 2.5), with the addition of a source term in the loading. In particular, a body force $\bar{f} = 2$ Pa is applied along x in the direction of the positive abscissas. The exact solution for the displacement field for this problem is known [22]:

$$\mathbf{u} = \begin{pmatrix} \frac{(x+1)(2\sigma - L(x-3))}{2E_1} \\ 0 \end{pmatrix} \qquad \text{for } -\frac{L}{2} \leq x \leq 0 \tag{2.19}$$

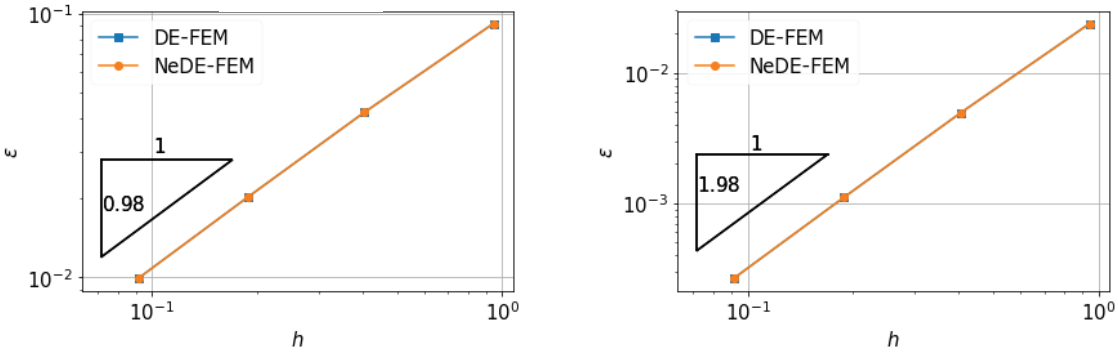
$$\mathbf{u} = \begin{pmatrix} \frac{E_2(3L+2\sigma)+E_1 x(2(\sigma+L)-Lx)}{2E_1E_2} \\ 0 \end{pmatrix} \quad \text{for} \quad 0 < x \leq \frac{L}{2} \tag{2.20}$$

For this problem, which presents a straight interface, the NURBS-enhanced DE-FEM is expected to yield the same performance of the standard DE-FEM in terms of error, which indeed can be observed in Figure 2.8 as an additional validation. The graph also shows that an optimal convergence rate is achieved for both the energy norm and the L_2-norm of the error for successive mesh refinements.

The second convergence test has been run based on Eshelby's inclusion problem, illustrated in Figure 2.9. The problem consists in the static analysis of a circular plate containing a circular inclusion under the imposed boundary displacement

$$\mathbf{u}_\Gamma = \begin{pmatrix} u_r \\ u_\theta \end{pmatrix} = \begin{pmatrix} r_u \\ 0 \end{pmatrix} \tag{2.21}$$

where $r_u = r_2$ is assigned.



(a) Energy norm of the displacement error.

(b) L_2-norm of the displacement error.

Figure 2.8: Convergence test for the bimaterial boundary value problem: comparison of the error for the standard DE-FEM and the NURBS-enhanced DE-FEM with respect to the mesh size.

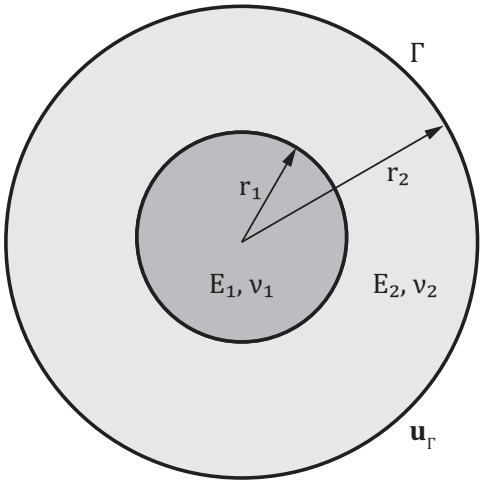


Figure 2.9: Eshelby's inclusion problem definition.

The exact displacement field for this problem is given as [22]:

$$\mathbf{u} = \begin{pmatrix} u_r \\ u_\theta \end{pmatrix} = \begin{pmatrix} r\frac{r_2^2+\alpha(r_1^2-r_2^2)}{r_1^2} \\ 0 \end{pmatrix} \qquad \text{for } 0 \le r \le r_1 \tag{2.22}$$

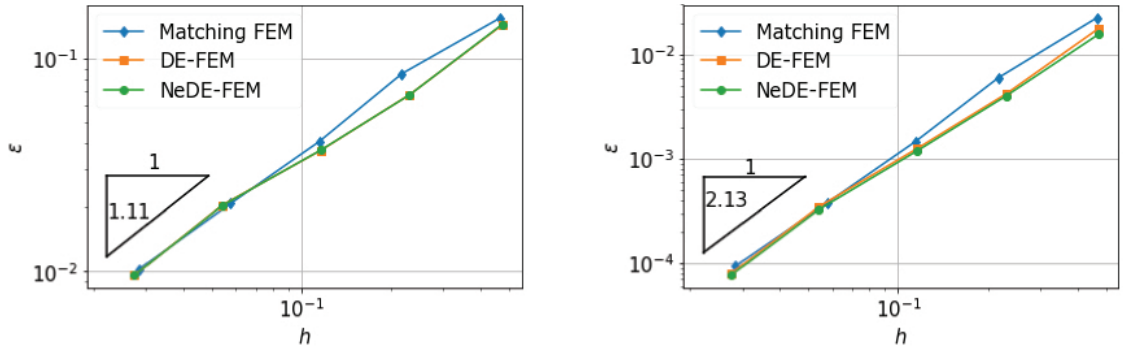
$$\mathbf{u} = \begin{pmatrix} u_r \\ u_\theta \end{pmatrix} = \begin{pmatrix} r\left(\alpha + (1-\alpha)\frac{r_2^2}{r^2}\right) \\ 0 \end{pmatrix} \qquad \text{for } r_1 \le r \le r_2 \tag{2.23}$$

with the parameter α, dependent on the Lamé parameters λ and μ for the two materials, being defined as:

$$\alpha = \frac{(\lambda_1+\mu_1+\mu_2)r_2^2}{(\lambda_2+\mu_2)r_1^2 + (\lambda_1+\mu_1)(r_2^2-r_1^2) + \mu_2 r_2^2} \tag{2.24}$$

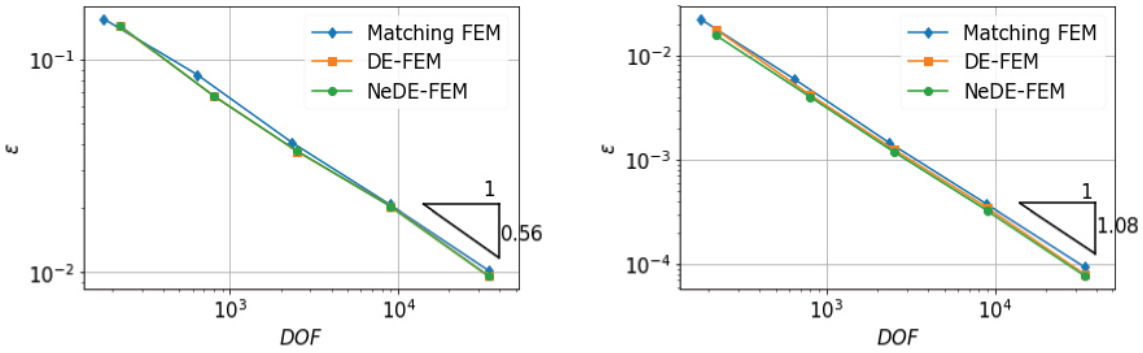
In the convergence test here reported, the material parameters are chosen as follows: $E_1 = 1$ Pa and $\nu_1 = 0.25$ for the inclusion, $E_2 = 10$ Pa and $\nu_2 = 0.3$ for the matrix. The radius r_1 of the inclusion is 0.901 m long, while the outer radius $r_2 = r_u$, whose value is used also in the refinition of the boundary condition, measures 2 m.

The convergence test for Eshelby's inclusion compares three different methods: the standard FEM with matching meshes, the standard DE-FEM with linear approximation of the interfaces and the NURBS-enhanced DE-FEM. In this specific case, only the inner boundary is modeled with discontinuity enrichments; the outer boundary, instead, simply consists in a piecewise linear approximation, as from the original mesh, also in the NURBS-enhanced method.



(a) Energy norm of the displacement error. (b) L_2-norm of the displacement error.

Figure 2.10: Convergence test for Eshelby's inclusion problem with respect to the mesh size: comparison between the standard FEM with matching meshes, the standard DE-FEM and the NURBS-enhanced DE-FEM. The terminal convergence rate is computed for the NURBS-enhanced DE-FEM.



(a) Energy norm of the displacement error. (b) L_2-norm of the displacement error.

Figure 2.11: Convergence test for Eshelby's inclusion problem with respect to the number of degrees of freedom: comparison between the standard FEM with matching meshes, the standard DE-FEM and the NURBS-enhanced DE-FEM. The terminal convergence rate is computed for the NURBS-enhanced DE-FEM.

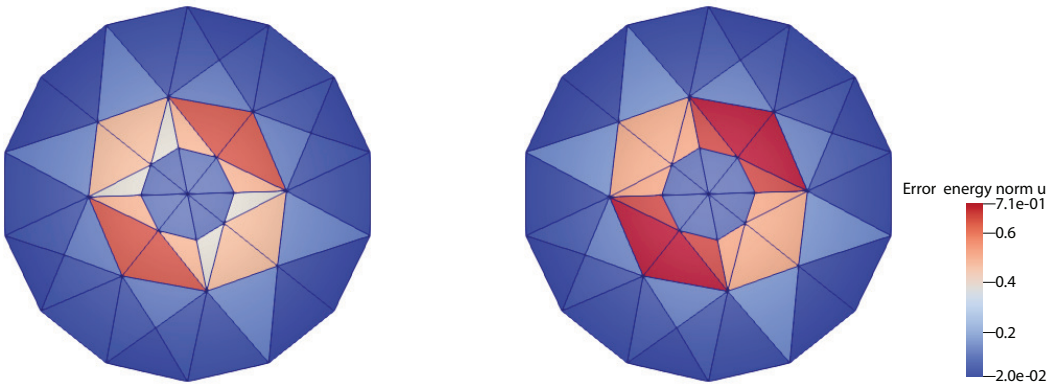
Figures 2.10 and 2.11 compare the energy norm and L_2-norm of the error with respect to the mesh size and number of degrees of freedom, respectively, with levels of refinement ranging from medium-coarse to very fine.

The curves in the convergence plots show that the NURBS-enhanced DE-FEM reaches an optimal convergence rate. While the increase in accuracy compared to the standard FEM with matching meshes is evident, due to the creation of smaller integration elements at the interface, the improvement in the error compared to the linearly approximated DE-FEM is only slight, decreasing with mesh refinements. This result is not surprising: the error in Figures 2.10 and 2.11 is averaged over all the mesh elements, making the improvements in the NURBS-enhanced elements less and less dominant for increasing refinements, as the linearly approximated boundary tends towards the shape of a circle. Moreover, penalizing factors are the low order of the shape functions, which relegates the increased accuracy of the NURBS-based formulation mostly to the interface elements, and the linear approximation of the external boundary.

To verify the greater accuracy of the NURBS-enhanced method to model structures with curved interfaces, Eshelby's inclusion problem has been tested for a coarser mesh than the ones utilized in the convergence test and for a smaller inclusion, of radius $r_1 = 0.5$ m, resulting in higher curvature. While the inclusion, for the linearly approximated DE-FEM, results in an octagon, the circular shape is preserved by the NURBS description of the curve. The improvements in the representation of the field are particularly evident on an element-wise evaluation of the error, which has been visualized in Figures 2.12 and 2.13. It has to be noted that, for the error representation, the software ParaView was used, in which the elements were constructed from nodal data; for this reason, all elements, including NURBS-enhanced integration elements, are visualized as simple triangles.

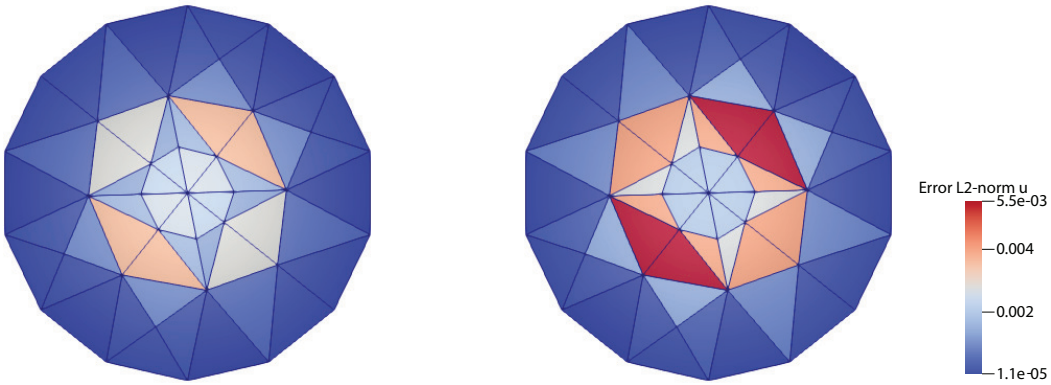
By the comparison in Figure 2.12, in the linearly approximated model, the integration elements adjacent to the interface show a peak in the error, which does not correspond to a similar error distribution in the

NURBS-enhanced case, due to the missing component originated from the approximation of the geometry. The propagation of this error to the outer elements can be observed in the comparative visualization of the L_2-norm of the error (Figure 2.13). Here, again, the spline-based model yields better results, where the field approximation error is a more dominant component. In this case, the contribution of geometric errors is mostly originated by the linear elements in the matrix, that do not match the radial symmetry of the problem. In Table 2.1, the improvement in the error is quantified, by giving a comparison between the results obtained with the two methods, for both an element-wise and a globally averaged computation of the error in the solution field. The results clearly show an improvement in the FEM approximation, which is especially visible in this study due to the very coarse mesh chosen.



(a) (b)

Figure 2.12: Eshelby's inclusion problem for the coarsest mesh: visualization of the energy norm of the displacement error by element for the NURBS-enhanced DE-FEM (a) and for the standard DE-FEM (b).



(a) (b)

Figure 2.13: Eshelby's inclusion problem for the coarsest mesh: visualization of the L_2-norm of the displacement error by element for the NURBS-enhanced DE-FEM (a) and for the standard DE-FEM (b).

Table 2.1: Eshelby's inclusion problem for the coarsest mesh: comparison of the error in the displacement field for the standard and the NURBS-enhanced DE-FEM.

Error	NeDE-FEM	DE-FEM	Improvement
maxϵ_E (element-wise)	$6.6\cdot10^{-1}$	$7.1\cdot10^{-1}$	7.0%
ϵ_E (global)	$1.6\cdot10^{-1}$	$1.7\cdot10^{-1}$	5.9%
maxϵ_{L_2} (element-wise)	$3.6\cdot10^{-3}$	$5.5\cdot10^{-3}$	34.5%
ϵ_{L_2} (global)	$4.6\cdot10^{-2}$	$5.6\cdot10^{-2}$	17.7%

2.2.3. Computing time

The study of Eshelby's inclusion problem allows formulating some observations on the increase in the computational time due to the NURBS-based enhancement. Prior to this evaluation, it must be pointed out that several strategies to improve the computational efficiency of the method have been implemented: these mostly consist in the creation of suitable data structures to avoid recomputing frequently used information associated with splines, for example coordinates obtained by direct or inverse mapping. Moreover, whenever a cartesian expression can be derived for level-set functions, this can be explicitly provided together with the explicitly defined discontinuity.

In the case of Eshelby's problem, the inclusion presents a simple circular shape, for which an explicit expression for the level-set function can be computed trivially. This allows comparing the computing time for the NURBS-enhanced method with both explicit and implicit level-set function, also relatively to the standard DE-FEM. The results are presented in Figure 2.14. The graph shows that the NURBS-based method is always computationally more expensive than the standard DE-FEM, which is expected considering the increased complexity of the geometric operations involved.

Profiling the code allows getting specific information on the most frequent and costly operations performed: for example, a significant amount of computing time is required for the calculation of cartesian points from parametric coordinates, when not sufficiently recurrent to justify storing their value. This operation by itself is very simple, but it needs to be repeated very frequently, for instance by the Newton-Raphson solver, and it often can't be avoided.

Although the computational cost undergoes a visible increase, it grows at the same rate as the standard DE-FEM when the level-set function is explicitly given, maintaining the computing time in the same order of magnitude, at least for the tested refinements. What is evident, instead, is the dramatic drop in efficiency due to the full computation of the level-set function, when not explicitly defined. This makes the NURBS-enhanced method disadvantageous, in particular, when general or more complex shapes are treated, as long as the search for intersections is performed systematically for all elements in the bounding box. This increased computational cost would be observed also in the standard DE-FEM, whenever discontinuities were defined by means of splines, even if eventually approximated by linear segments. However, considering that the aim of NURBS-based formulations is to allow more freedom to model arbitrary shapes, it would be convenient to adopt a search strategy that limits the computation of level-set functions only to the strictly necessary cases, for example by reusing known information from neighboring elements.

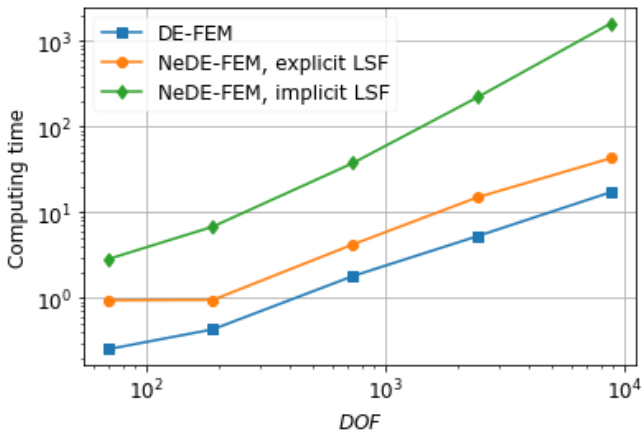


Figure 2.14: Computing time for the solution of Eshelby's inclusion problem for different methods and mesh refinements.

2.2.4. Straight crack

The validation studies for the NURBS-enhanced DE-FEM have been completed by comparing its accuracy in the study of strong discontinuity problems with the results obtained with the standard DE-FEM. As a test case, the problem of a plate under tension presenting a straight internal crack has been selected, for which the stress intensity factors (SIFs) at the two crack tips are computed.

Stress intensity factors are parameters associated with the modes of crack deformation (in two dimensions: opening and sliding, in Figure 2.15), that provide a measure of the respective singular stress fields at

(a) (b)

Figure 2.15: Opening (2.15a) and sliding (2.15b) modes of crack deformation [36].

the crack tips. The SIF associated with the opening mode is denoted as K_I, while K_{II} is associated with the sliding mode.

A common way to compute mixed-mode stress intensity factors is derived from the interaction energy integral, or M-integral [39], expressing the conservation of energy for two superposed equilibrium states, an actual and an auxiliary state (the latter denoted hereafter with the subscript *aux*). For the purposes of finite element analysis, the M-integral can be conveniently computed from its domain integral formulation [30], which reads:

$$M = \lim_{\Gamma \to 0} \int_A (-\sigma_{ik} \varepsilon^{aux}_{ik} \delta_{1j} + \sigma_{ij} u^{aux}_{i,1} + \sigma^{aux}_{ij} u_{i,1}) q_{1,j} \, dA \tag{2.25}$$

The integral in equation 2.25 is evaluated over the area A delimited by an arbitrary path, which encloses the crack tip. The weight function q is introduced, which assumes values ranging from 1, in the elements completely contained by the path, to 0, at the outer nodes of the intersected elements. By taking the weight function's derivative $q_{1,j}$, only the latter bring an actual contribution to the integral.

By computing the M-integral with equation 2.25 and appropriately choosing the auxiliary equilibrium states, the SIFs can be calculated by means of the equation [39]:

$$M = \frac{2}{E^*}(K_I K_I^{aux} + K_{II} K_{II}^{aux}) \quad \text{with } E^* = \begin{cases} \frac{E}{1-\nu^2} & \text{for plane strain} \\ E & \text{for plane stress} \end{cases}, \tag{2.26}$$

Usually, a pure mode I auxiliary state is first selected with $K_I^{aux} = 1$ and $K_{II}^{aux} = 0$ to compute K_I, followed by a similar computation with a pure mode II auxiliary state to extract K_{II}.

Following this procedure, the problem of a horizontal crack in a plate under biaxial tension has been studied, whose definition is illustrated in Figure 2.16. Under the hypothesis of an infinite plate with external traction applied at infinity and unloaded crack edges, the problem results in a mode I deformed configuration, due to the symmetry of the loading with respect to the crack axis [36]. In practice, the plate has been modeled as finite-dimensional, with applied traction $\sigma = 1$ Pa and material properties $E = 1000$ Pa and $\nu = 0$. The mesh constructed for this problem presents a medium level of refinement, as visible from Figure 2.17, with the crack crossing a total of 20 triangular elements. The objective, in fact, is not as much to achieve very high precision in the computed results, but rather to show that the accuracy of the NURBS-enhanced DE-FEM in computing the stress intensity factors is the same as the standard DE-FEM.

The results for the two methods, with the M-integral being computed over a circular area of radius 0.04, are reported in Table 2.2 and compared with the analytical solutions. Due to the symmetry of the problem, the values of the stress intensity factors computed are reported only for the right crack tip. By comparing the results for the two numerical methods, it can be observed that the SIFs are identical up to the third digit, for both mode I and mode II. This validates the capabilities of the NURBS-enhanced method to handle strong discontinuities, at the same level of the linearly approximated one. The accuracy for this example is acceptable, considering the relatively coarse mesh used, the low order of the field approximation and the limited size of the domain used for the plate.

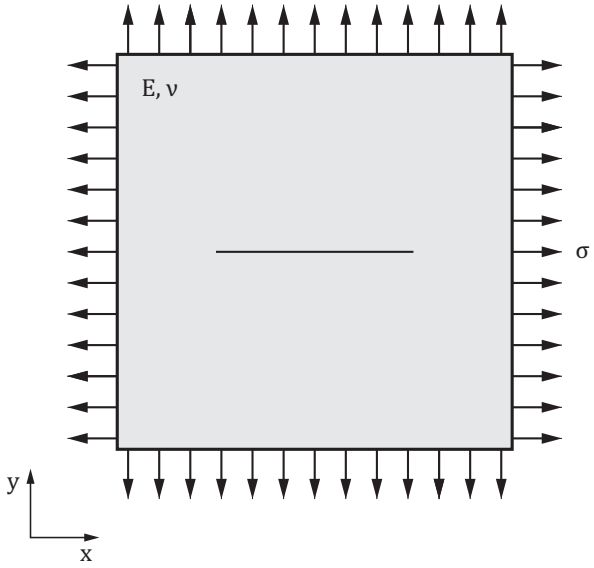


Figure 2.16: Straight crack on a plate under tension: problem definition.

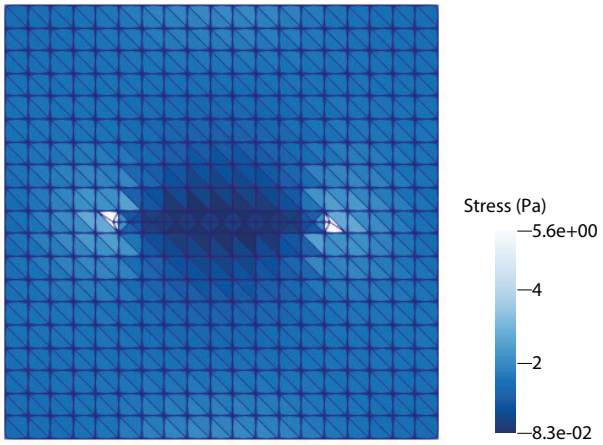


Figure 2.17: Straight crack on a plate under tension: mesh and computed stress field.

Table 2.2: Straight crack in a plate under biaxial tension: analytical and computed stress intensity factors.

Method	K_I	K_{II}
Analytical	1.0	0.0
Standard DE-FEM	0.99642	0.04920
NURBS-enhanced DE-FEM	0.99391	0.04734

2.3. Application problems

2.3.1. Plate with four-lobed inclusion

The objective of implementing a spline-based method, like the NURBS-enhanced DE-FEM proposed in this work, is to allow modeling arbitrary geometries with more freedom and accuracy. This could be a necessity when designing structures with complex holes or composites with irregular interfaces, due to specific constraints or desired global properties. Upon verification of the performance of the NURBS-enhanced DE-FEM when handling circular and straight discontinuities, the following application aims at showing its capabilities to solve problems with interfaces of more complex, arbitrary shapes.

The example presented here consists of a square matrix with a four-lobed inclusion of softer material. The

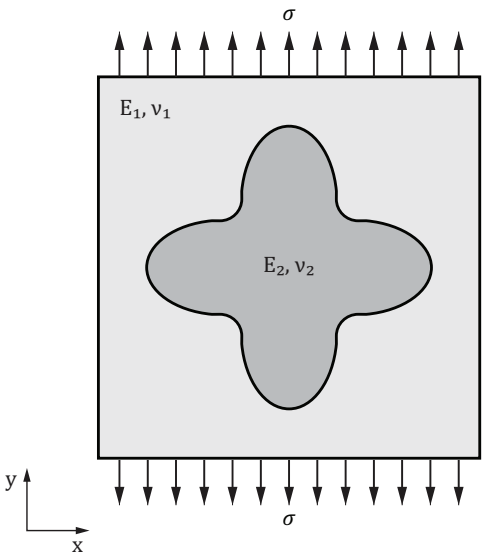
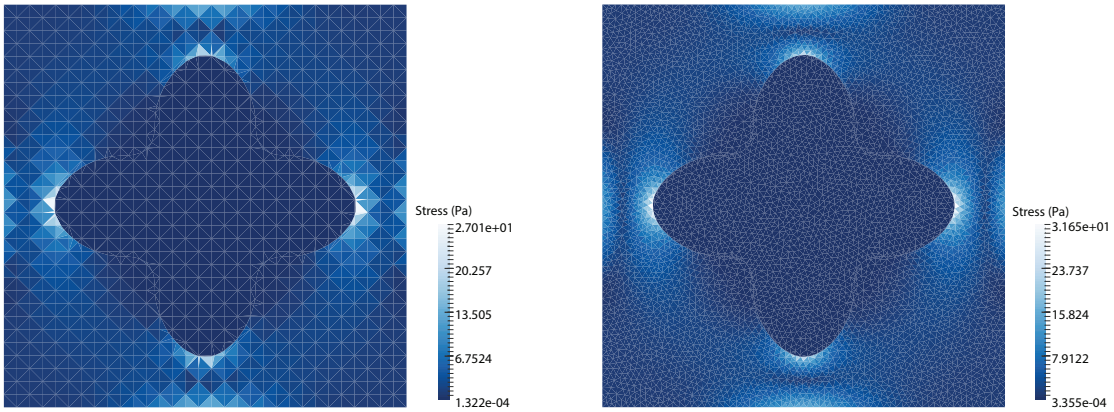


Figure 2.18: Plate with a four-lobed inclusion: problem definition.



(a) (b)

Figure 2.19: Stress field for the problem of a plate with a four-lobed inclusion: comparison of the results for two different meshes.

plate is in traction along the y-direction on the top and bottom boundaries, as illustrated in Figure 2.18. The applied traction σ has unit magnitude; matrix and inclusion have a Young's modulus of 1000 Pa and 0.1 Pa respectively, while for both Poisson's coefficient is equal to 0.3.

The computed stress fields are reported for two levels of mesh refinement in Figure 2.19, while Figure 2.20 reports the deformed configuration, where the displacement has been scaled by factor 5. These results allow observing that the most stressed region is the portion of the matrix adjacent to the left and right lobes, where most of the tensile deformation for the stiffer material is concentrated. As an effect to this deformation in tension, the materials contract in the opposite direction, resulting in a compression stress which is particularly evident in the matrix, at the interface with the top and bottom lobes. A noticeable effect of having a hard material with a softer inclusion lies also in the presence of distortion at the centre of the plate's edges, where the matrix bends to allow larger displacement at the inclusion, while maintaining a low level of deformation closer to the corners.

From this example, it can be concluded that the proposed NURBS-enhanced DE-FEM can effectively handle discontinuities with complex shapes and variations in the local convexity at the interface. This capability is not straightforward, considering that computing level-set functions and intersection points involves an optimization routine and the use of a nonlinear solver, respectively. The two examples demonstrate that the construction of integration elements can be carried out successfully for different mesh refinements, even

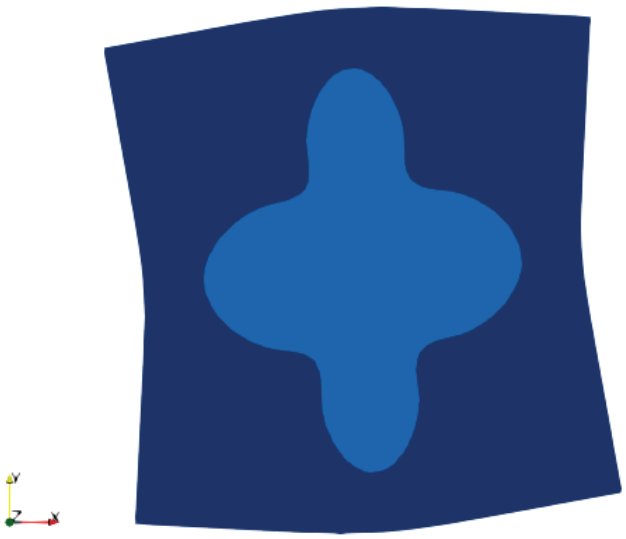


Figure 2.20: Deformed configuration for the plate with a four-lobed inclusion. The displacement has been magnified by a factor 5 for a better representation. The slight tilting, emphasized by the scaling factor, is an effect of the asymmetry of the mesh.

when mesh nodes are located very close to the interface or mesh lines are almost tangential to the discontinuity.

The comparison between the outputs of the two simulations shows that, even with the larger mesh size, the magnitude of the stress field is predicted with sufficient precision throughout the domain: the simulations converge to the same stress profile with increasing refinements. However, the overall representation of the stress field in Figure 2.19a is coarsely tessellated, thus requiring refinements if a more accurate approximation is desired. This is not due to the quality of the geometric modeling itself, but by the low order of the shape functions, that can only achieve a linear approximation of the solution field.

2.3.2. Circular arc crack

The last application under study aims at showing the capabilities of the proposed NURBS-enhancement to the DE-FEM in the treatment of curved crack problems. The chosen example consists in a plate under biaxial tension with a circular arc crack, for which an analytical expression for the stress intensity factors is given [6, 31]. The problem definition is illustrated in Figure 2.21. As in the validation example of the straight crack, unit surface traction has been applied along the normal outward direction, while the material parameters are chosen as $E = 1000$ Pa and $\nu = 0$.

Under the assumption of an infinite plate subjected to uniform tension at infinity [6, 31], the stress intensity factors are expressed analytically as:

$$K_I = \sqrt{\pi a}\left(\left(\frac{\sigma_{yy}+\sigma_{xx}}{2} - \frac{\sigma_{yy}-\sigma_{xx}}{2}\sin^2(\alpha/2)\cos^2(\alpha/2)\right)\frac{\cos(\alpha/2)}{1+\sin^2(\alpha/2)} + \frac{\sigma_{yy}-\sigma_{xx}}{2}\cos(3\alpha/2) - \sigma_{xy}(\sin(3\alpha/2)+\sin^3(\alpha/2))\right) \tag{2.27}$$

$$K_{II} = \sqrt{\pi a}\left(\left(\frac{\sigma_{yy}+\sigma_{xx}}{2} - \frac{\sigma_{yy}-\sigma_{xx}}{2}\sin^2(\alpha/2)\cos^2(\alpha/2)\right)\frac{\sin(\alpha/2)}{1+\sin^2(\alpha/2)} + \frac{\sigma_{yy}-\sigma_{xx}}{2}\sin(3\alpha/2) + \sigma_{xy}(\cos(3\alpha/2)+\cos(\alpha/2)\sin^2(\alpha/2))\right) \tag{2.28}$$

In our case, uniform biaxial traction is applied, therefore $\sigma_{xx} = \sigma_{yy} = \sigma$ and $\sigma_{xy} = 0$.

In order to obtain values for the stress intensity factors which are independent of the specific choices for dimensions and loading, it is often convenient to normalize them, by dividing them by $\sigma\sqrt{\pi a}$. In the rest of the section, the notations K_I and K_{II} will refer to the normalized stress intensity factors.

Numerical results have been computed for five different choices of α, ranging between 10° and 50°. The arc cord keeps a constant value of 0.5 m, with the radius changing to satisfy the assigned angle. In each

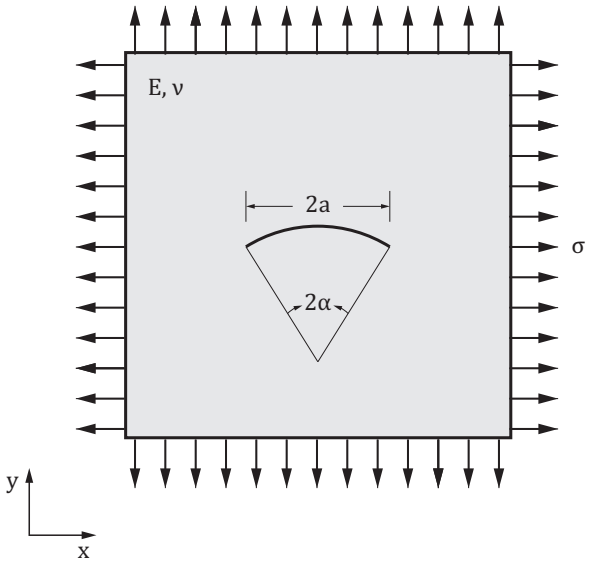


Figure 2.21: Definitions for the problem of a plate under biaxial tension, with a circular arc crack spanning an angle 2α.

case, the crack crosses approximately 25 mesh elements, on a square mesh 3 times wider than the crack. The values for the normalized stress intensity factors have been computed with the NURBS-enhanced DE-FEM, by taking a circular area of radius 0.2 m to evaluate the M-integral. The results are reported in Table 2.3 and visualized in Figure 2.22, in comparison with the analytical values from equations 2.27 and 2.28.

Table 2.3: Curved crack in a plate under biaxial tension: normalized stress intensity factors, obtained analytically [31] and by FEA.

Angle α	10^o	20^o	30^o	40^o	50^o
K_I (analytical)	0.98868	0.95598	0.90528	0.84128	0.76897
K_I (NURBS-enhanced DE-FEM)	0.97544	0.94443	0.89420	0.83631	0.77261
Error (%)	-1.34	-1.21	-1.22	-0.59	0.47
K_{II} (analytical)	0.08650	0.16857	0.24257	0.30620	0.35857
K_{II} (NURBS-enhanced DE-FEM)	0.08031	0.16454	0.22016	0.27055	0.29939
Error (%)	-7.15	-2.39	-9.24	-11.64	-16.50

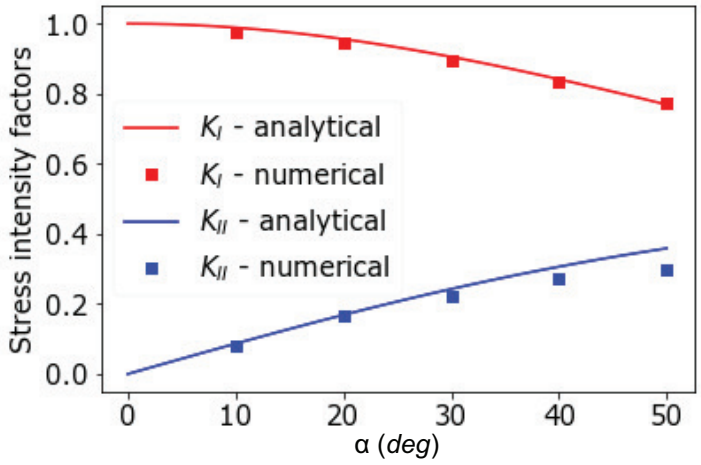


Figure 2.22: Normalized stress intensity factors for a circular arc crack on a plane under biaxial tension: analytical and numerical results.

By comparing analytical and numerical results, it can be seen that the simulations are very accurate, in

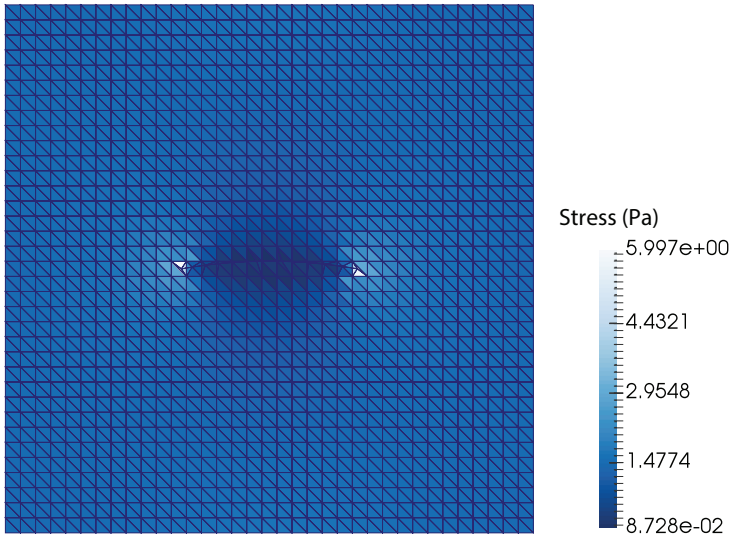


Figure 2.23: Stress field for a plate with a circular arc crack, for $\alpha = 10^o$. A concentration of stress is observed at the crack tips, while the edges of the crack tend to an unloaded profile, as expected from theoretical models.

particular in the computation of the stress intensity factor for mode I, which presents an error in the order of 1%. Overall, the computational values underestimate the stress intensity factors and, in particular, tend to lose accuracy for increasing curvature of the crack, as observed in the error for K_{II}.

An interpretation of these results can be formulated by considering how the stress and displacement fields change with the crack's shape: an increase in its curvature is expected to be reflected to a certain degree in the profile of the field. On the other side, the FEM approximation on a relatively coarse mesh has limited possibilities to model such a stress or displacement distribution. In this perspective, it is reasonable to see a decrease in precision for increasing α, especially in the computation of K_{II}: the SIF associated with the sliding deformation mode, which is the more affected by the curved profile of the crack.

In general, it has been observed that other factors potentially affecting the accuracy of the computed solutions are the aspect ratio of the integration elements and the configuration of the discontinuity with respect to the element edges, for example in the case of tangent lines. Overall, the method is robust with respect to these conditions, thanks to corrective factors for high slopes in the enrichment functions; however, the creation of enriched nodes too close to existing mesh nodes may result, in some cases, in nonphysical stress concentrations at a crack's edges. The occurrence of tangent discontinuities or close-to-coincident enriched and mesh nodes is more frequent for very refined meshes, which was observed also by Safdari *et al.* [24], and for shapes that resemble the pattern of the mesh. In these cases, a slight increase in the error may occur. This phenomenon can be observed by considering the problems here simulated with $\alpha = 10^o$ and $\alpha = 20^o$, where, for the first case, a worsening of the accuracy in the approximation is noticed. In fact, the crack with $\alpha = 10^o$ is almost tangent to the mesh lines (see Figure 2.23) and experiences a slightly larger error, with particularly visible effects for K_{II}.

From this example, it can be concluded that the proposed NURBS-enhanced DE-FEM can lead to accurate results when modeling curved cracks and computing the associated stress intensity factors, although the method can be sensitive to poor aspect ratios in the integration elements and to the low order of the field approximation. With the flexibility of NURBS-based shapes, not only simple curves like circumference arcs can be treated: applications can be extended to arbitrary, more complex shapes.

2.4. Conclusions and recommendations

2.4.1. Conclusions

In this chapter, a NURBS-based enhancement has been proposed for the DE-FEM, for the treatment of both weak and strong discontinuities. The formulation employed is completely compatible with the standard DE-FEM and needs limited modifications, mostly for the treatment of geometries and the mapping from the splines' parameter space. In the finite element library, this required the creation of a specific element type, to which NURBS-enhanced shape functions and integration were associated, but the structure of the

implementation remained substantially the same as for the standard DE-FEM.

The NURBS-enhanced DE-FEM proposed has been tested for different types of structural problems and successfully validated for the exact recovery of constant states of stress and for optimal convergence rates. The examples included display the versatility of the method for handling discontinuities of varying complexity, including interfaces with high curvature and multiple changes in concavity. The NURBS-enhanced DE-FEM shows good robustness with respect to different types of meshes and in presence of nodes located in close proximity of the discontinuity.

The spline-based method has been proved to be equivalent to the standard DE-FEM when modeling straight interfaces and cracks. Moreover, as expected, the NURBS-enhanced DE-FEM yields more accurate results than the DE-FEM based on linear approximations when the analysis is carried out on problems with curved discontinuities, especially for the coarsest meshes, due to the exact geometric representation of the boundaries.

The novel study to extend the method to the treatment of curved strong discontinuities has led to successful outcomes. The simulations performed on fracture problems with arc cracks point out very accurate results in the computation of the stress intensity factors, with an error in the order of 1% for K_I, when cracks cross as little as 25 mesh elements. Better accuracy is observed for smaller curvatures, as an effect of the approximation in the elements located further from the crack.

In the current implementation, only linear shape functions have been used, which represent the actual limitation to the maximum accuracy that can be achieved. In the standard DE-FEM, mesh refinements are required to improve the quality of both the solution field and the geometry. In the NURBS-enhanced DE-FEM, instead, geometries are exact even with the coarsest meshes; nonetheless, refinements are generally needed to achieve an acceptable representation of stresses and displacements. Often, this requires meshes that are fine enough to attenuate the advantages of employing an exact geometry compared to a very refined linear approximation.

The introduction of NURBS-based enhancements results in an overall increase in the computational cost, due to the more elaborate geometric operations involved. The most significant contribution to this burden comes from the computation of the level-set function – if not explicitly known – when searching for intersected elements. However, a wide range of problems dealing with curved discontinuities concerns shapes for which level-set functions can be defined explicitly, for example in presence of circular inclusions. In these cases, the computational cost of a NURBS-enhanced simulation is maintained in the order of that of the standard DE-FEM. In general, though, the increased cost of the geometric operations associated with NURBS leads to a significant slowing down of the computation, making the NURBS-enhanced version of the DE-FEM less attractive than the standard DE-FEM for problems with complex interfaces on very refined meshes, but more suitable for the treatment of problems with relatively coarse meshes and geometries for which a cartesian expression can be derived.

2.4.2. Recommendations

The necessity of refining the mesh to improve the field representation, as previously mentioned, brings significant limitations to the application of this method, which results in only moderate improvements in modeling the geometries, against a relevant increase in the computational effort. Based on this insight, it is suggested to extend the application of NURBS-based enhancements to *p*-FEM: this method's capabilities of representing higher-order fields, combined with the possibility offered by the NURBS-based mapping to model geometries that cannot be interpolated exactly by higher-order polynomials, would result in a method that can compute non-linear fields in presence of complex-shaped discontinuities with minimal mesh refinement. In this perspective, the increased computational cost of performing geometric operations on NURBS would be balanced by reducing the overall number of degrees of freedom in a more costly finite element method.

A more general recommendation about the speed of the current implementation can be drawn from the consideration that a consistent part of the computational effort is rooted into calculating level-set functions and finding intersections between discontinuities and element edges. These operations are performed for each element independently, which suggests the possibility of parallelizing this stage of the computation for speed-up. Besides, during the development of the code, introducing strategies to reduce the number and cost of the operations, such as dictionaries to store recurrent data, has brought outstanding improvements in the computing time. A more thorough revision of the code, specifically aiming at increasing its efficiency, would likely bring a further reduction in the computational burden of utilizing NURBS-based geometries.

Apart from this, it can be observed that a full element-wise search for intersections is not the most suitable approach to finding representing elements: due to the continuity of a NURBS curve across mesh lines, it is

expected to find elements crossed by discontinuities next to other intersected elements. Consequently, it is advised to continue the research in the direction of fast-marching methods as a more efficient option. An alternative recommendation is to make more effective use of the information available from the computation of level-set functions. Instead of only considering the sign of the level-set function at a point to assign it to one side of a discontinuity, the absolute value of the level-set function can be also actively exploited to estimate the position of the discontinuity. This would allow excluding from the search elements located in areas where the level-set function cannot possibly experience any change in sign.

For what concerns the application of NURBS-based enhancements to two-dimensional problems, some extensions and improvements can be still considered. In the first place, the present study has not involved problems with multiple discontinuities crossing each other. For this purpose, the implementation of specific functions with nonlinear solvers would be needed, to find the coordinates corresponding to the crossing point in the different parametrizations. In addition, it is recommended to introduce more flexibility in the procedure for the creation of integration elements, which is currently performed automatically by Delaunay triangulation. The possibility of interacting with the algorithm and enforcing additional criteria for the choice of the triangulation could, in fact, prevent the creation of tangled elements and even allow splitting integration elements with splines [33]. Moreover, it is suggested to invest further research in the application of boundary conditions on enriched nodes belonging to NURBS-based discontinuities, which has not been validated in this work. This would not only allow handling a greater variety of problems, but also offer the possibility to reduce simulations on the basis of symmetry considerations, decreasing the overall computational cost.

The last recommendation concerns the postprocessing of NURBS-enhanced simulations. Some work had been dedicated to the representation of meshes including spline-based discontinuities and integration elements, exported in the form of LaTeXplots, which was carried out successfully. However, that implementation is not supported by the current version of Hybrida and, therefore, the study to include a visualization of fields and errors accounting for curved elements could not be completed. In general, however, ParaView resulted to be a convenient tool for plotting the outputs, which generally demonstrated good versatility and flexibility in data visualization for daily use.

3

Reflection

3.1. Timeline

At the beginning of the thesis, a timeline was set for the main tasks. The project had an expected duration of 9 months, not including short breaks, which has been overall respected: I started working on my literature study at TU Delft in mid-December 2016 and the thesis submission is expected by the beginning of October 2017. The main phases in which this project was structured and accomplished are four: the preliminary review of the literature, the formulation and implementation of the proposed method, the application and, eventually, the reporting phase. The planning for the project was set to reflect this structure. The expected timeline, as it was defined at the beginning of the work, is reported in Figure 3.1. From the original planning, the actual execution followed a slightly different timeline, which is included in Figure 3.2.

The comparison between the two shows that most of the goals were completed within the expected time; however, a few tasks were hard to estimate in the beginning, mostly because only hands-on experience could provide a complete idea of the actual challenges of the project. This holds, in particular, for the implementation of NURBS-enhanced analysis and the extension of the work to an original study. In the first case, the implementation itself followed the planned schedule; however, it required several weeks for perfectioning and debugging, that were accounted for only partially. Similarly, estimating the time requirements for the application problems in the very beginning presented some difficulties, as the planning had to be based on assumptions on the success of the studies. The decision of the research direction, in particular, involved considering several possible applications and running preliminary tests, before choosing the most promising option; with this respect, the outcomes could be hardly predicted with sufficient accuracy. Moreover, what was not taken into account in the initial planning was the necessity to modify the code to comply with the updates the finite element library and, with that, repeat simulations that were already completed. This, in particular, can be observed in the planning of validation and application problems in Figure 3.1, compared to the actual execution time.

3.2. Research direction

When I selected my graduation project, I was particularly attracted by the possibility of working with geometry and programming, which I have always found very interesting. Since the beginning, one main research question was defined: implementing a NURBS-based enhancement for the DE-FEM, to allow modeling weak discontinuities based on exact geometries. From that, the project was open to expansion in further directions and applications.

Choosing the research direction has been probably one of the most delicate tasks throughout this work. On one side, I did not have an outstanding personal preference for a specific discipline or problem to study; on the other, it would have been interesting to choose an application for which I could provide some novel insight. The latter meant also that it would have been harder to estimate the outcome of such a study beforehand.

After the implementation of the NURBS enhancements was complete and validated, some time has been dedicated to conduct some research and preliminary tests to explore different possibilities. These included applying the method to shape optimization, to the study of acoustic metamaterials and to fracture problems.

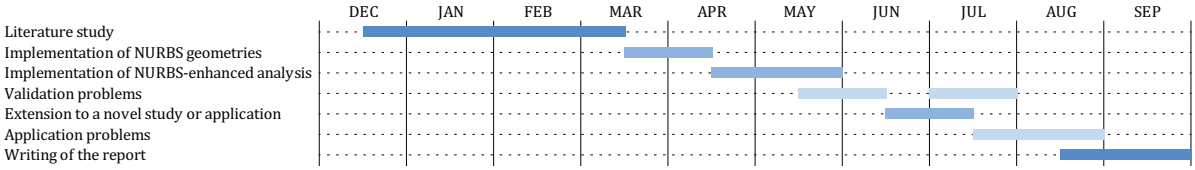


Figure 3.1: Original planning for the thesis project.

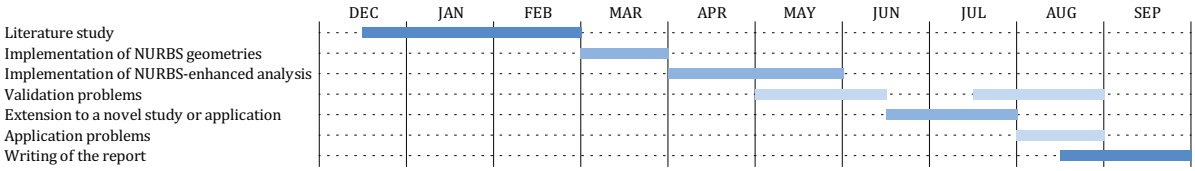


Figure 3.2: Timeline for the actual execution of the thesis project.

The first topic had a certain interest because spline geometries are generally good candidates for shape optimization, resulting in smooth geometries for a limited number of design variables. However, not having sensitivities available in the current DE-FEM implementation was a significant drawback, that limited the possibilities of achieving satisfying results, especially for problems with multiple design variables. For this reason, this direction was eventually discarded. Among the other two, the study of phononic crystals would have required the validation of the NURBS-enhanced DE-FEM for dynamic problems and the extension to a substantially different physics, whereas the study of fractures would have maintained the focus on structural mechanics, while extending the application to strong discontinuities and requiring some additional implementation, for example in the reconstruction of crack tips and in the computation of SIFs.

The application to the study of cracks was eventually chosen, which would have allowed exploiting the whole range of discontinuities that the DE-FEM is able to handle also in the NURBS-enhanced implementation.

3.3. Methodology

The initial planning involved a reflection about the methodology to be followed throughout the work, which accounted for the specific characteristics and requisites of each stage. The approach was kept open to revision along the project, to meet specific goals and necessities.

Upon the completion of the literature study, the research questions were formulated, which guided the work for the rest of the project. At that point, the main tasks required for the implementation were identified, based on the findings from the literature review and on the modules that needed to be incorporated into the existing code. These tasks were broken down into small, specific steps, such as the construction of B-splines and the implementation of the intersection algorithm. For each of these steps, validation tests have been designed along with the implementation, aiming at verifying that theoretical results could be satisfied and that examples from the literature could be reproduced accurately. This validation process has been described more in detail, with the associated tests, in Appendix B and has helped building the code progressively on strong foundations and facilitated the debugging work significantly.

For the NURBS-enhanced analysis as a whole, the first validation examples were performed in comparison with the standard DE-FEM for straight discontinuities, case in which the two methods are expected to yield substantially identical results. Most of the debugging work took place at this validation stage: it allowed not only assessing the implementation as a whole, with all the modules working together, but also verifying how my code was interacting with Hybrida's environment.

The application problems to be studied were chosen based on the objectives for the thesis and on the specific research direction chosen: for the points to be proved or examined, suitable test cases were identified that could provide quantitative results and, along with these, the necessary insights. Factors that were considered for this purpose were the availability of analytical solutions and the possibility of studying the problems with different methods for comparison. In most cases, multiple simulations were run to identify the possible sources of error and evaluate the robustness and sensitivity of the implemented method with respect to the variation of several conditions, such as the mesh refinement, the size of the domain or the radius for the computation of the stress intensity factors. However, a thorough sensitivity study with respect to these

conditions was not in the objectives for this thesis.

3.4. Working with Hybrida

The entirety of the implementation of the proposed method has been carried out within Hybrida, the finite element library being developed within the Structural Optimization and Mechanics group under the lead of professor Aragón. As the name hints, Hybrida is mostly implemented in Python with elements of C++ and it is the result of numerous contributions, including the work of several doctoral and master students. Working within this framework has represented a great opportunity for the development of this project, while bringing along several challenges.

My first encounter with Hybrida presented some obstacles: even having experience with Python or object-oriented programming, which was mostly not my case, can be generally insufficient to start immediately modifying and expanding the code with confidence. It took a few months to become accustomed with the architecture of the code and, in particular, with the thinking process behind the implementation of the DE-FEM modules. However, all these challenges represented above all an added opportunity for learning: this was my first real chance to experience working on the development of a complex and structured code, requiring me to adopt the group's working methods and tools and to interact with the other contributors. By attending regular meetings and participating in discussions about Hybrida's structure and operations, I got valuable insights about how to introduce improvements into my own implementation and about where to leave room for the modifications, accounting for future research directions in the group.

Another main characteristic of working within the Hybrida team has been the dynamism of the whole context. Having approximately 4 people simultaneously involved in the development of the DE-FEM framework meant not just having the opportunity for an active discussion of ideas, inputs and doubts, but also a continuous process of improvement for the code and its structure. In fact, while I was working on the spline-based implementation, the framework it was built on was being modified in parallel. In a few occasions, this resulted in my code being not compatible with the latest version of Hybrida any more, which sometimes required rethinking how to perform specific operations in order to comply with it. More in general, the dynamic and versatile character of the DE-FEM implementation constrained my code to be flexible, not intrusive and robust.

A final consideration can be included about the documentation of my work. In my view, contributing to a broader framework means giving other people the possibility to understand, interact with and modify my code in the future. As I directly experienced throughout this project, these tasks can become quite challenging and time-consuming if who implemented a certain module left almost no documentation behind: even if reading a file makes it obvious *what* a portion of code does, it is not always automatic to discern *what for* and *why*. Hence, my implementation has been always accompanied by accurate documentation, which worked as a good reminder for me throughout the project and, more in general, helps the code's readability and understanding in case of future need.

3.5. Points of personal improvement

Working on this thesis has been both a considerable learning process and a personal challenge. This project started from discovering the theory behind splines, that allowed me to understand how the curves that I used to know from design software are constructed, based on their mathematical formulation. Both the study of the literature and the work with Hybrida led me through a deeper understanding of the finite element method under multiple perspectives, touching not only different formulations but also, above all, the practical aspects behind the development of finite element programs. One of the biggest challenges and gratifications has been definitely represented by the implementation of the code, due not only to my almost nonexistent experience with object-oriented programming, but also to the work within a complex and continuously evolving software framework in parallel with other contributors. From the personal perspective, the debugging tasks have been an excellent training for my patience.

By looking at my work upon its completion, I would probably reconsider the approach followed to carry out the simulations. While the validation procedure applied throughout the implementation had very specific goals and milestones since the very beginning, the method followed to gain insights on the overall NURBS-enhanced DE-FEM lost a bit of this structure. On the one hand, my review of the literature had been mostly focused on the formulation and implementation of the methods proposed, while overlooking, until further reading at a later stage, the reasons behind the choice of specific validation or illustration problems. This is, for example, the case of convergence studies. On the other hand, I realize that, in some circumstances, I

focused more on the details of a specific problem and on trying to improve the obtained results, instead of concentrating on the bigger picture and on the point I was trying to prove. In that sense, interrupting the work at some stages to sum up what had been done until that point could have been helpful to define the next steps in a more systematic way.

Another phase that, *a posteriori*, I would structure differently is the literature review and, in particular, the way it related to the implementation. When I started the thesis, my estimate was that I should have dedicated roughly 3 months to preliminary research and, towards the second half of this period, I should have started getting familiar with Hybrida as well. In practice, following this process was not always very effective: some aspects of the IGFEM and DE-FEM became more clear to me only once I faced their implementation and, in the literature study, I sometimes dedicated time to examine topics that were not strictly relevant to the project. In conclusion, it would have probably been more useful to conduct the review of the literature and of Hybrida's implementation in parallel, at least for what concerned the DE-FEM, with more interaction between the theory and its application.

A

NURBS-based geometries

A.1. Construction and properties of a spline

As illustrated in Chapter 1, B-spline basis functions are built recursively from the zeroth-order basis over a series of knot spans, according to the Cox-de Boor formula (equation 1.7). NURBS basis functions are then computed as rational functions of this B-spline basis, in which the contribution of each B-spline function is scaled by a weight. A NURBS curve in the physical space is constructed by means of control points, whose contribution is blended within the curve by means of the respective NURBS basis functions.

Overall, for the definition of a NURBS curve, a vector of knots, weights and control points are required, in a number that satisfies the desired order and continuity properties for the curve. By construction, according to equation 1.7, a basis function of order p spans $p+1$ knot intervals. Consequently, if m is the number of knots in the knot vector, a total of n basis functions with complete support can be constructed, with [23]:

$$n = m - p - 1 \tag{A.1}$$

n is also the number of weights and control points: from both sets, each weight and control point will be uniquely assigned to one specific basis function.

The choice of the parametric coordinates that define knot spans has fundamental effects on the continuity of the curve, in particular when knots are coincident with a certain multiplicity. This can be understood by considering a simple example. Let us assume to have a first-order basis, for which each function spans two knot intervals. If two knots are coincident, the interval between the two becomes null; therefore, none of the functions will be able to blend both before and after that specific coordinate and smoothness will be lost across that knot. In the general case, a NURBS curve is C^{p-k}–continuous at a point mapped from a knot with multiplicity k [23]: Figure A.1 reports an effective illustration of this property. In a spline, at points where C^0– or C^{-1}–continuity is verified, control points are interpolatory for the curve and the spline is tangent to the control polygon, being the polygon formed by joining the control points in sequence. Moreover, in an open knot vector, commonly used to construct open curves with the first and last control points as endpoints, the first and the last knots are taken with multiplicity $k = p+1$.

This whole procedure for the construction of NURBS may not seem very intuitive when trying to achieve specific shapes: in particular, because control points have non-interpolatory nature and their coordinates, together with the weights, need to be tuned to obtain the desired geometries. However, most of the software for computer-aided design allows building splines graphically, without manually defining the parameters for the curve, which are set automatically. As an example, Figure A.2 displays the possibility to extract the parametrization for a curve constructed with CAD. This illustration has been realized with Grasshopper, a plug-in for the CAD software Rhinoceros; more in general, NURBS parametric definitions can be obtained also from common CAD files.

A.1.1. Closed splines

The curves described in the previous section are generally utilized for representing open shapes or to model curves in such a way that the first and last point are constructed independently of each other. However, some applications may require modeling closed shapes with a certain degree of smoothness, also in the point where the two ends meet.

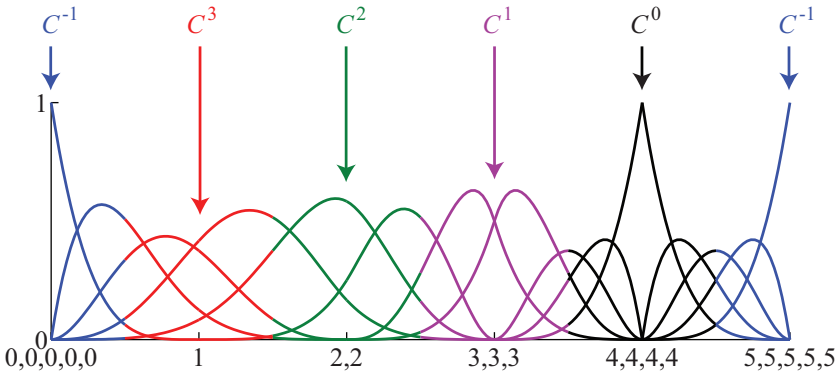


Figure A.1: Effect of knot multiplicity on the continuity of a fourth-order spline basis [8].

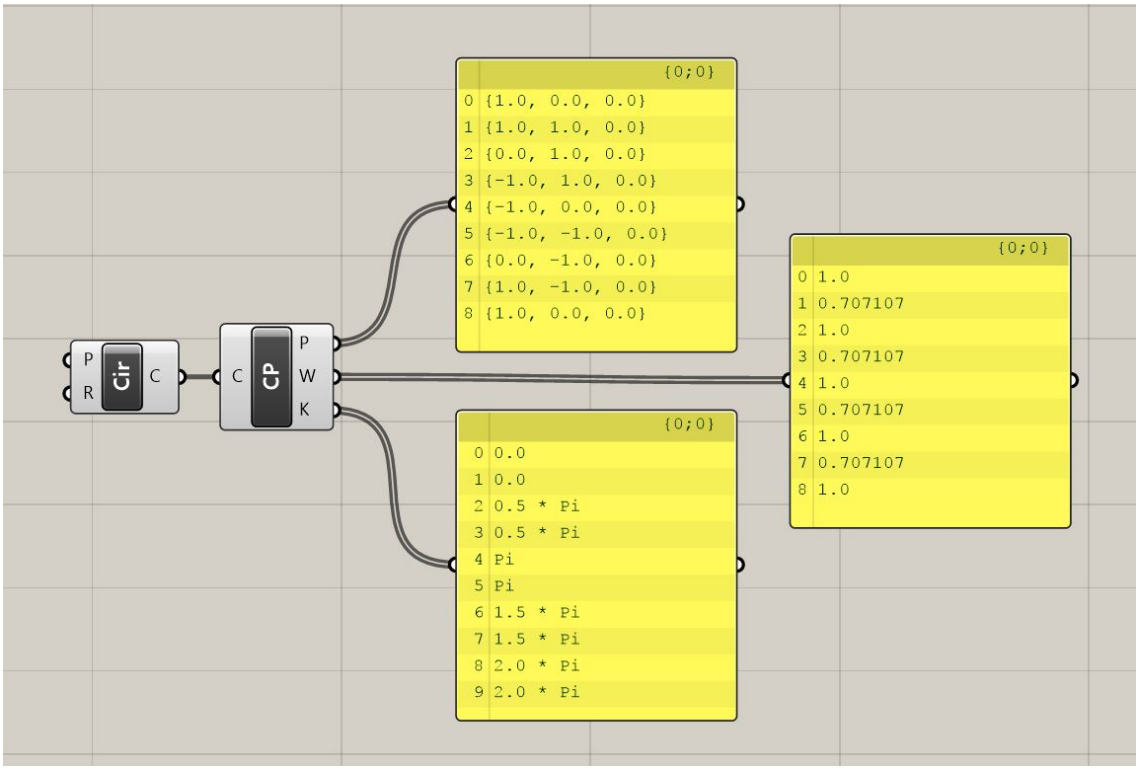


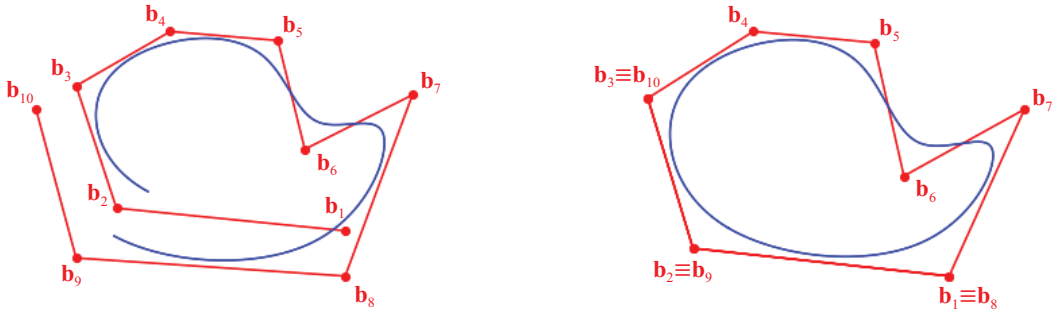
Figure A.2: Extraction of NURBS parametrization for a circle modeled in Grasshopper. By comparison with Figure 2.1, it can be observed that the construction is equivalent.

A technique to ensure the smoothness of a closed curve is illustrated by Lowther and Shene [17]. The last p control points wrap over the first p points, becoming coincident. Moreover, the knots at the ends of the knot vector are taken with single multiplicity instead of being repeated $p+1$ times and, to achieve the total number of knots to satisfy equation A.1, p equidistant knots are added at both sides of the knot vector. The physical points that compose the curve are still mapped from the original knot span only, so that the additional knots are actually located outside the curve's parametric domain.

Figure A.3, based on an example by Shene [29], illustrates the construction of a closed B-spline, by superposing its first and last control points, and the choice of the associated knots.

A.1.2. Convex hull

NURBS are intrinsically bounded and smooth curves. Unlike polynomials, their regularity improves with the order of the curve, due to the higher number of knot spans the basis functions can blend in. Moreover, they satisfy a *variation-diminishing property*, stating that a spline curve is smoother that its associated control polygon [23].



(a) (b)

Figure A.3: Construction of a closed B-spline (A.3b) from a generic open B-spline (A.3a). The third-order basis is built for an evenly-spaced knot vector $[-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10]$, while the curve is plotted for s in the range $[0,7]$.

This property results in an important consequence for curve: a NURBS is always contained inside the *convex hull* determined by its control points, as long as the weights are non-negative. In a broad sense, the convex hull can be taken as the convex polygon defined by the control points. Most commonly, though, the convex hull for a spline is defined as the union of the convex polygons constructed by taking $p+1$ successive control points [8], that results in a tighter, often globally non-convex hull. An illustration of this construction is given in Figure A.4.

Splines also satisfy an additional local property for convex hulls. In particular, for a parametric coordinate $s \in [s_i, s_{i+1})$, the corresponding physical point $\mathbf{c}(s)$ is strongly contained in the convex hull defined by the control points $\{\mathbf{b}_{i-p}, ..., \mathbf{b}_i\}$.

A.1.3. Knot insertion

Knot insertion is a form of refinement consisting in adding one or more knots without changing the shape of a spline [23]. When increasing the number of knots, satisfying equation A.1 requires either increasing the order of the curve or adding control points; the latter option, if performed according to the algorithm illustrated in this section, allows preserving the shape of the original curve.

The starting B-Spline curve will be considered to have order p, a knot vector $\Xi = [s_0, ..., s_{m-1}]$ of length m and a set of n control points $\mathbf{B} = \{\mathbf{b}_0, ..., \mathbf{b}_{n-1}\}$. A new knot $\bar{s}$ will be inserted, such that $\bar{s} \in [s_k, s_{k+1})$ for a certain k. According to the local convex hull property previously mentioned, the segment of the curve affected by the new knot vector lies within the hull defined by $\{\mathbf{b}_{k-p}, ..., \mathbf{b}_k\}$: outside this range, the control points will remain untouched.

Hence, the knot insertion algorithm to determine the modified control points reads [23]:

$$\bar{\mathbf{b}}_i = (1 - a_i)\mathbf{b}_{i-1} + a_i \mathbf{b}_i \qquad (i \in [k-p+1, k]) \tag{A.2}$$

with:

$$a_i = \frac{\bar{s} - s_i}{s_{i+p} - s_i} \tag{A.3}$$

The newly computed control points are substituted into the set of control points of the B-spline, which becomes: $\{\mathbf{b}_0, ..., \mathbf{b}_{k-p}, \bar{\mathbf{b}}_{k-p+1}, ..., \bar{\mathbf{b}}_k, \mathbf{b}_k, ..., \mathbf{b}_{n-1}\}$.

In case of a NURBS, the knot insertion operation can be conveniently performed on its associated B-spline: the latter shares the same knot vector and, if a general control point for the NURBS has coordinates $\mathbf{b}_i = (x_i, y_i, z_i)$ and weight w_i, the associated control point for the B-spline will be expressed as $\mathbf{b}_i = (x_i, y_i, z_i, w_i)$. The knot insertion algorithm for the B-spline will automatically allow computing the new control points and the associated weights for the NURBS. The example reported in Figure 2.2 illustrates the transformation of an arc's control points and parametrization after multiple knot insertion.

Knot insertion is a form of refinement for NURBS that turns particularly useful in isogeometric analysis, where knot spans define mesh elements. In NURBS-enhanced finite element methods, instead, the interest in knot insertion has different motivations. On one side, it leads to tighter convex hulls, but most of all it allows reducing the continuity in a curve's parametrization by placing multiple knots at a given location. To achieve C^{-1}–continuity in the parametrization (as in Figure 2.2) and be able to isolate a portion of the curve

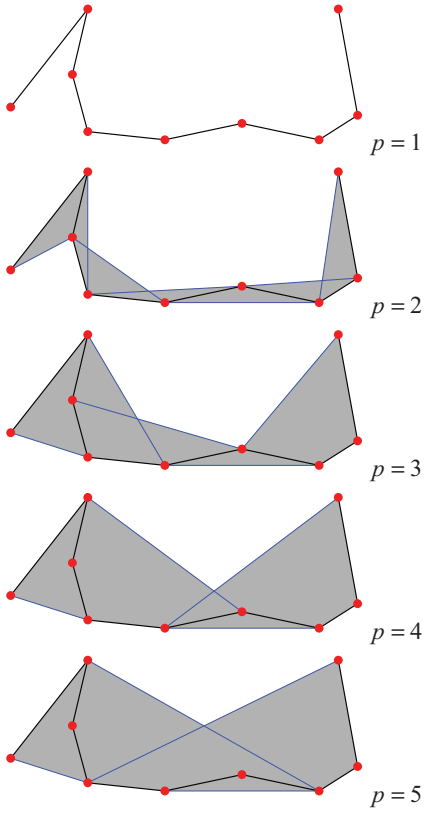


Figure A.4: Examples of convex hull for splines of different order, given the same control points [8].

and express it with only self-contained information, the end knots need to appear with $p+1$ multiplicity. Then, if the knot range for the segment lies between the i-th and j-th knot in the knot vector of the original curve, the weights will be extracted in the index range $[i, j-p-1]$, together with the corresponding control points.

A.1.4. Derivatives and direction vectors

The k-th derivative of a NURBS basis function can be computed analytically via the recursive formula:

$$\frac{d^k}{ds^k} R_i^p(s) = \frac{A_i^{(k)}(s) - \sum_{j=i}^{k} \binom{k}{j} W^{(j)}(s) \frac{d^{(k-j)}}{ds^{(k-j)}} R_i^p(s)}{W(s)} \tag{A.4}$$

where:

- $A_i^{(k)}(s) = w_i \frac{d^k}{ds^k} N_{i,p}(s)$,
- $W(s) = \sum_{i=1}^{n} N_{i,p}(s) w_i$,
- $W^{(k)}(s) = \frac{d^k}{ds^k} W(s) = \sum_{i=1}^{n} \left(\frac{d^k N_{i,p}(s)}{ds^k} w_i \right)$,
- $\binom{k}{j} = \frac{k!}{j!(k-j)!}$.

The derivatives of B-spline basis functions are equal to the derivatives of regular polynomials of the same order. From equation A.4, the derivative of a NURBS curve is computed trivially as:

$$\frac{d^k}{ds^k} \mathbf{c}(s) = \sum_{i=1}^{n} \frac{d^k}{ds^k} R_i^p(s) \mathbf{b}_i \tag{A.5}$$

This derivative corresponds to the tangent vector to the curve at the coordinate mapped from s, pointing in the direction of the increasing curvilinear abscissa, with length representing the speed of the coordinate variation. In the NURBS-enhanced implementation in this work, all tangent direction vectors are computed by normalizing expression A.5.

The normal vector at a given point, instead, can be calculated with different methods, including a finite differences approach based on the variation of the tangent direction at neighboring coordinates or by vector product between the tangent at that point and the normal to the plane.

A.2. Computation of the level-set function

In the DE-FEM, the evaluation of a point's location with respect to a discontinuity is performed via a level-set function. Specifically, the level-set function is here defined as a signed distance function, i.e.:

$$LSF(\mathbf{p}) = d(\mathbf{p}, \mathbf{c}(s)) \cdot sgn(\mathbf{p}), \tag{A.6}$$

where $d(\mathbf{p}, \mathbf{c}(s))$ is the distance between a given point $\mathbf{p}$ and the curve. This is determined through the solution of a minimization problem, for which the point on the NURBS is found with the minimum distance from the assigned point. The sign function $sgn(\mathbf{p})$ assumes three possible values, based on the relative position of the point with respect to a domain Ω associated with the discontinuity:

$$sgn(\mathbf{p}) = \begin{cases} +1 & \text{if } \mathbf{p} \notin \Omega \\ 0 & \text{if } \mathbf{p} \in \partial\Omega \\ -1 & \text{if } \mathbf{p} \in \Omega \end{cases} \tag{A.7}$$

Ω is often taken as the area or volume enclosed by an interface, thus meaning that the sign function is negative for all points located inside. However, for strong discontinuities or in general open curves, Ω is intended as the portion of the total material located on the left side of the path. This definition of the sign function employs a convenient property of NURBS, namely the fact of being oriented curves: the sign function is taken as the sign of the vector product between the distance vector and the tangent vector to the curve.

A.2.1. Computation of the distance

Finding the distance between one point in space and a NURBS curve is equivalent to solving the optimization problem:

$$\min_s f(s) = \min_s ||\mathbf{p} - \mathbf{c}(s)|| = \min_s \sqrt{\sum_{i \in \{x,y\}} (p_i - c_i(s))^2} \qquad \text{with } s \in [s_0, s_{m-1}] \tag{A.8}$$

For a NURBS curve, this results in a non-linear unconstrained optimization problem, with a bounded design variable s. Alternatively, by slightly manipulating the formulation, this could also be expressed as a constrained, unbounded optimization problem, in which the bounds have been turned into constraints:

$$g_1(s) = s_0 - s \leq 0 \qquad \text{and} \qquad g_2(s) = s - s_{m-1} \leq 0 \tag{A.9}$$

Solving this optimization problem allows finding the minimum value of the function f, being the point-curve distance, and the value of s at the point of the curve from which this distance is taken.

There exist several optimization algorithms, whose implementation in Python is also available, which do not require the computation of the Jacobian, or sensitivity, of the objective function with respect to the design variables. However, some numerical experiments allowed observing that these tend to be more sensitive to the initial guess for s and to become unstable in some cases.

Having obtained the NURBS function $\mathbf{c}(s)$ and its derivative analytically brings a relative ease in computing the Jacobian of the objective function, which is, in this case, a scalar. A simple derivation with respect to the design variable s yields:

$$\frac{\partial f(s)}{\partial s} = \frac{1}{f(s)} \cdot \sum_i \left(-\frac{\partial c_i(s)}{\partial s} (p_i - c_i(s)) \right) \qquad (i \in \{x, y\}) \tag{A.10}$$

The optimization has been performed with the library *scipy.optimize.minimize*, from which a method could be selected among sequential least squares programming (SLSQP), truncated Newton (TNC) and the limited-memory BFGS (L-BFGS-B). All these methods were able to carry out the required tasks successfully

when sensitivities were provided, while they were unstable in some cases when the Jacobian was approximated, with slightly better performance for the TNC algorithm. The truncated Newton method was generally applied to all the examples studied. The objective function has been provided as an input, together with its derivative, the bounds (s_0, s_{m-1}) for the design variable (or the corresponding constraints) and an initial guess for the parameter s. The initial guess is chosen as the closest point to $\mathbf{p}$ among a set of $3n$ evenly distanced samples taken along the curve, with n being the number of control points. This method has been proved to be robust even in cases of curves with multiple local maxima or minima, as illustrated in the validation example in Appendix B (Figure B.3) or in the lobated inclusion problem in Chapter 2.

A.3. Intersection with a line

For the purposes of finite element analysis, it is necessary to find intersection points between mesh lines and NURBS-based discontinuities. In the framework offered by Hybrida, mesh lines are not given explicitly, but are uniquely defined by the location of the nodes, accessed via connectivity information. Intersections are searched for when mesh lines are crossed (once) by a discontinuity, which is verified when the level-set function has opposite signs at the end nodes.

Two main strategies to find line-discontinuity intersections have been identified: either finding the point of the mesh for which the level-set function is null, or finding the coordinate on the NURBS at which the equation of the line is satisfied. While the first method is in use in the standard DE-FEM, where level-set functions are generally assigned explicitly, it is not a recommended procedure for a spline-based discontinuity: the computation of the level-set function is quite expensive and including it in an iterative solver (a bisection method in the case of the standard DE-FEM) should be avoided whenever possible.

The second option, then, appears to be significantly more convenient: the expression of mesh lines can be determined trivially from the coordinates of two nodes and, afterwards, the operation of finding intersections results in solving a simple nonlinear equation for the parameter s, where the nonlinear contribution comes from the computation of the physical coordinates $\mathbf{c}(s)$. The derivatives of the function can be computed analytically, which allows employing the Newton-Raphson method quite efficiently, with very few iterations.

An algorithm is implemented to provide the nonlinear solver with an initial guess. This first computes an estimate for the intersection point between mesh and discontinuity by linear interpolation of the coordinated of the two end points, weighted by their respective level-set function values: at the point found, the level-set function is null in linear approximation. The initial guess for the Newton solver will be then taken among a set of candidate parametric coordinates, as the one that approaches the point found more closely.

A.4. NURBS-enhanced mapping

The formulation for the NURBS-enhanced mapping is determined by the choice of the reference domain, on which the shape functions and the integration points are defined.

The choice of this reference domain has been based on the most recurrent type of NURBS-based element encountered in this sort of problems, *i.e.* triangles (but quadrilaterals could also be considered) with one spline edge. Mapping from a rectangular reference domain presents key advantages: while the direction along the curved edge is mapped non-linearly, the mapping in the opposite direction, towards the other vertex, is linear [25]. This yields less computational effort and a reduced complexity. Moreover, rectangular quadrature was demonstrated to be more efficient for polynomials with relatively low order, which is likely to be our case, whereas triangular quadrature is more accurate for high-order approximations. For our type of problems, transformations from a rectangular reference domain have been considered to be the most suitable choice.

Sevilla *et al.* [26] suggest the use of a rectangular domain $[s_a, s_b] \times [0,1]$, where the values in the abscissa are exactly the parametric coordinates of the NURBS segment. This choice of the reference domain is quite intuitive, but it presents several difficulties. If the spline segment appears with negative orientation within one element, the integration will result in a negative Jacobian unless one of the axes or the orientation of the element is inverted. Moreover, this mapping is hard to extend to the integration of elements with multiple NURBS edges, when the knot intervals facing each other have a different length.

Soghrati *et al.* [33] include elements with 2 or 3 curved edges in their method and choose a more general reference domain $[0,1] \times [0,1]$. From this, the points are first mapped to spline parametric coordinates and then, via the NURBS mapping, onto the physical domain. On top of this, it has been observed that a preliminary mapping is necessary to transform the coordinates of the Gauss points, conventionally defined in the square $[-1,1] \times [-1,1]$, to the reference coordinates chosen by Soghrati *et al.*.

After considering different options for the reference domain, this has been chosen as the square $[-1,1] \times [-1,1]$. The approach is similar to the one followed by Soghrati *et al.*, but it requires one less mapping: Gauss points are already given with respect to these coordinates. Moreover, Lagrangian shape functions for quadrilateral are also already built in the existing FE implementation on this reference domain. This construction results closer to the one adopted by Szabó *et al.* [37] for the blending function method. The additional mapping needed compared to the approach followed by Sevilla *et al.* [26] presents the advantages of having always a regular polygon, even with multiple NURBS edges with knot vectors of different sizes, and of having more control in handling the parametrization. This is particularly useful if an element presents one spline edge with opposite orientation with respect to the element boundary: then the mapping can be easily adjusted to reverse the knot span.

B

Validation for NURBS-enhanced operations

The implementation of NURBS-based geometric operations was supported by a systematic validation process, which was mostly excluded from the main body of this report, but still deserves some attention. After illustrating NURBS properties and geometric features in Appendix A, this chapter is dedicated to a more detailed description of the verification procedure, that allowed validating the implementation steps throughout the work, before the final validation of the resulting method presented in Chapter 2.

For the implementation of the NURBS-enhanced DE-FEM, the main tasks were initially identified as:

- the construction of B-splines and NURBS,
- the introduction of knot refinement,
- the computation of level-set functions,
- the implementation of the intersection algorithm,
- the extraction of NURBS-based discontinuity segments,
- the implementation of NURBS-based integration, and
- the creation of NURBS-enhanced integration elements.

While the validation of the last point was actually the validation of the NURBS-enhanced DE-FEM as a whole, the focus of this chapter will on the intermediate steps and, in particular, on the ones that required the verification of specific features and properties. The description of more trivial validation processes will be omitted.

B.1. B-spline basis functions and curves

The implementation of B-spline basis functions and curves has been validated in several ways. These consisted in:

- reconstructing a set of B-spline basis functions illustrated in the literature,
- verifying the partition of unity condition at every coordinate,
- validating computed B-Spline basis functions and their derivatives with their analytical expressions.

By following this process, a first qualitative verification was performed by comparing the obtained curves with examples from the literature, for which the defining parameters were known. One of the tests chosen consisted in reproducing the basis in Figure A.1, from a publication by Cottrell *et al.* [8]. The resulting curves are reported in Figure B.1. On the same basis, the partition of unity has been verified: at every point s, the sum of the basis functions is expected to be equal to 1. The maximum error for this case has been computed as $3.33 \cdot 10^{-16}$, in accordance with the machine precision.

Subsequently, a simpler example has been taken, for which the analytical expression of the basis functions and their derivatives were computed. The chosen example consists in a second-order basis, defined over the

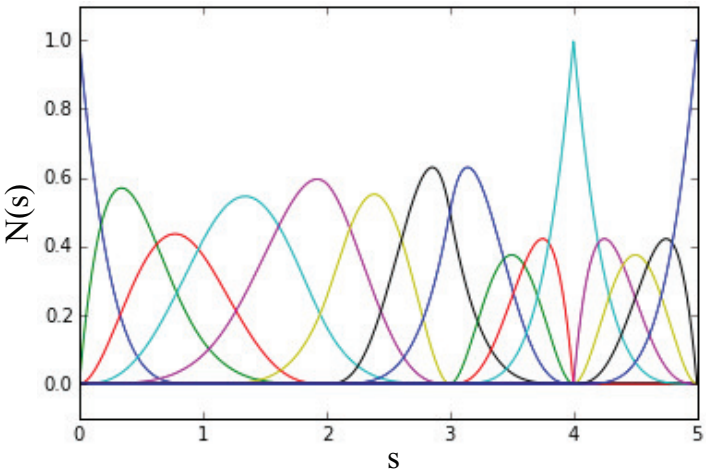


Figure B.1: Validation example for the construction of B-spline basis functions, based on the basis in Figure A.1 [8].

knot vector $[0,0,0,1,2,2,2]$. The B-spline basis functions are expressed as:

$$N_1 = \begin{cases} (1-s)^2 & s \in [0,1) \\ 0 & s \in [1,2) \end{cases} \tag{B.1}$$

$$N_2 = \begin{cases} s(1-s) + \frac{1}{2}s(2-s) & s \in [0,1) \\ \frac{1}{2}(2-s)^2 & s \in [1,2) \end{cases} \tag{B.2}$$

$$N_3 = \begin{cases} \frac{1}{2}s^2 & s \in [0,1) \\ \frac{1}{2}s(2-s) + (2-s)(s-1) & s \in [1,2) \end{cases} \tag{B.3}$$

$$N_4 = \begin{cases} 0 & s \in [0,1) \\ (s-1)^2 & s \in [1,2) \end{cases} \tag{B.4}$$

The computation of first- and second-order derivatives from these expressions is trivial. The basis functions and their derivatives for this examples were then constructed with the implemented algorithm (as in Figure B.2). The verification test on a sample of 1000 points resulted in identically 0 error for the basis and the second-order derivative and a discrepancy in the order of the machine epsilon ($4.44 \cdot 10^{-16}$) for the first-order derivative.

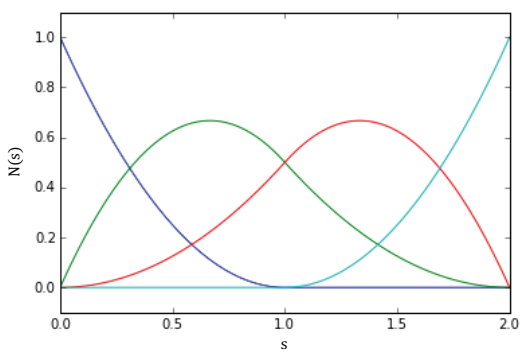
B.2. NURBS basis functions and curves

NURBS have been validated with an approach similar to the one adopted for B-splines, in particular based on the following observations:

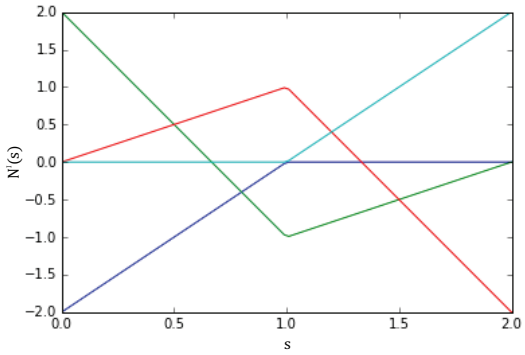
- when the weights are all equal, the particular case is verified for which NURBS are also B-splines;
- NURBS basis functions build a partition of unity at every point;
- NURBS can model conic sections exactly.

The verification based on the equivalence of NURBS and B-spline basis functions for all $w_i = 1$ has been performed based on the example proposed by Cottrell et al. [8] (Figure A.1), adopted also for the validation of B-splines. Based on the B-spline basis functions, the corresponding NURBS basis has been constructed and the error between the two has been evaluated for a set of points. The maximum error has been measured to be $2.22 \cdot 10^{-16}$, in accordance with the machine precision. Similarly, the derivatives of the basis functions have been compared, showing a maximum error of $8.88 \cdot 10^{-16}$ for the first-order derivative and $5.33 \cdot 10^{-15}$ for the second-order. The partition of unity property was also verified, with the same procedure described for B-splines.

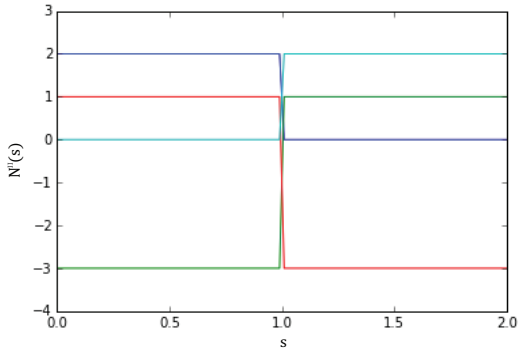
The capability to model conic sections exactly has been demonstrated by constructing a NURBS-based circle with unit radius, as illustrated in Figure 2.1. The validation consisted in measuring the radius at every point: on a set of 100 points, it was verified to be equal to 1 with a maximum error of $2.22 \cdot 10^{-16}$.



(a) Second-order B-spline basis.



(b) First derivative of the B-spline basis.



(c) Second derivative of the B-spline basis.

Figure B.2: B-spline basis functions and their derivatives used for validation with the analytical expressions.

B.3. Computation of the level-set function

The implementation of the level-set function was a delicate process, which required several attempts before yielding successful results. Designing suitable tests assumed particular importance in this case, as diverse geometries allowed highlighting different pitfalls in the initial implementations. The most relevant ones are here reported.

A first validation was performed for a circular interface. This test case, in appearance quite simple, revealed some complexity due to its spline-based construction: a NURBS-based circle is in fact an open curve wrapped around itself by a convenient choice of its control points; however, at the two coincident endpoints, the parametrization experiences a jump. This introduced several problems in the computation of the level-set function in correspondence of this neighborhood: if the initial guess falls on the wrong end of the curve, it is very hard for the algorithm to reach the opposite one to find the actual closest point. Noticing frequent errors in correspondence of endpoints allowed concluding that the candidate initial guesses for determining the closest point of the curve should not include the first and last coordinates, as the choice of one between the two would depend, in closed curves, only on numerical error.

Another validation example worth mentioning consisted in a closed interface with a concave and elongated shape, as illustrated in Figure B.3. This test case aimed at verifying that the algorithm could behave robustly also in case of multiple local maxima or minima; in particular, it allowed choosing a suitable number of initial guesses to ensure correct results even for problems with relatively complex geometries.

B.4. Numerical integration

The correct implementation and the accuracy of NURBS-based integration have been first verified by assessing the method's capability to calculate areas. Subsequent tests involved the recovery of constant states of stress, for which the procedure has been illustrated in Chapter 2.

Preliminary tests were required in order to choose a suitable number of integration points. In particular, being the number of points along ξ and η decoupled, the accuracy of the integration has been evaluated with respect to variations in the number of points in both coordinates.

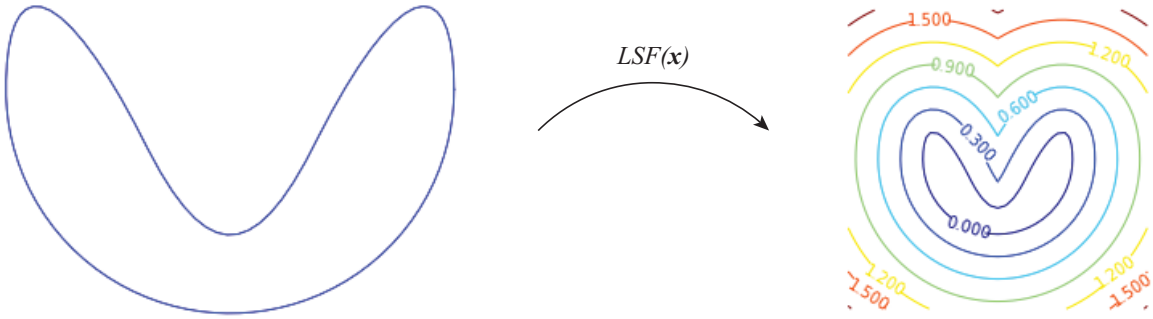


Figure B.3: Validation example for the computation of the level-set function, showing a concave interface and the contour plot for the level-set function.

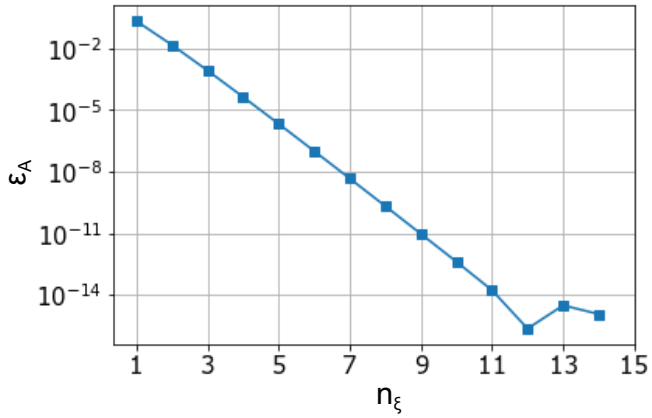


Figure B.4: Error in the area integration versus the number of Gauss points in the ξ-direction. Machine precision is reached for $n_\xi = 12$.

A first check has been run for the integration of areas, which are expected to be computed with only one integration point in case of simple rectangular elements. In our case, however, introducing NURBS to model geometries makes it impossible to achieve exact integration via Gauss quadrature. In this example, a quarter of a circle has been integrated, which is modeled as a triangle with one NURBS edge, mapped from the ξ-direction of the reference domain. As showed in Figure B.4, the number of quadrature points along ξ need to be increased up to 12 to achieve machine precision: a number which is much higher than the one needed to integrate a second-order curve like a circle. Increasing the number of points along η, instead, has been demonstrated to introduce no improvements, which was expected considering the linearity of the mapping in this direction.

For what concerns the integration of an element's contribution to the stiffness matrix, for which the integral $\int_e \mathbf{B}^T\mathbf{H}\mathbf{B}|\mathbf{J}_e|d\xi d\eta$ has to be computed, some preliminary considerations had to be made. In general, for a rectangular or parallelogram element, the determinant of the Jacobian is constant, while the rest of the integrand function yields a second-order polynomial [3]. Knowing that n Gauss points can integrate exactly functions of order $2n-1$ or lower, exact integration can performed in this case with two quadrature points in each direction. Indeed, bilinear quadrilateral elements, also in Hybrida, are integrated with a total of 2×2 Gauss points.

In the case of NURBS-based elements, the introduction of splines increases the order of the term to be integrated. The determinant of the Jacobian is generally not constant and, in particular, it includes the derivative of the NURBS function, that is also a rational polynomial. Conversely, the shape functions in reference coordinates present substantially the same definition as in standard quadrilateral elements. The overall function, due to the nonlinear rational contribution of the spline parametrization, cannot be integrated exactly by Gauss quadrature. However, it has been considered irrelevant to reach machine precision in the integration: in fact, the geometrical operations carried out before the analysis are also performed up to a certain tolerance, which has been considered as a reasonable limit to the expected accuracy of the integration.

Some numerical tests have been performed to determine how many Gauss points to set, based on the accuracy in representing the solution field and the areas: no significant improvement was noticed by increas-

ing the number of points above 6 in ξ and above 2 in η. Therefore, these values were chosen as the standard number of Gauss points for NURBS-enhanced elements.

Bibliography

[1] Alejandro M. Aragón and Angelo Simone. The Discontinuity-Enriched Finite Element Method. *International Journal for Numerical Methods in Engineering*, 2017. ISSN 1097-0207. URL `http://dx.doi.org/10.1002/nme.5570`.

[2] Ivo Babuška and Jens M. Melenk. The partition of unity finite element method. Technical report, University of Maryland – Institute for Physics Science and Technology, 1995.

[3] Klaus-Jürgen Bathe. *Finite element procedures*. K. J. Bathe, 2006.

[4] Yuri Bazilevs, Victor M. Calo, Yongjie Zhang, and Thomas J. R. Hughes. Isogeometric fluid–structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38(4-5):310–322, 2006.

[5] Kwok Wah Cheng and Thomas-Peter Fries. Higher-order XFEM for curved strong and weak discontinuities. *International Journal for Numerical Methods in Engineering*, 82(5):564–590, 2010.

[6] Brian Cotterell and James R. Rice. Slightly curved or kinked cracks. *International Journal of Fracture*, 16 (2):155–169, 1980.

[7] J. Austin Cottrell, Alessandro Reali, Yuri Bazilevs, and Thomas J. R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5257–5296, 2006.

[8] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[9] Christophe Daux, Nicolas Moës, John Dolbow, Natarajan Sukumar, and Ted Belytschko. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 48:1741–1760, 2000.

[10] Emmanuel De Luycker, David J. Benson, Ted Belytschko, Yuri Bazilevs, and Ming-Chen Hsu. X-FEM in isogeometric analysis for linear fracture mechanics. *International Journal for Numerical Methods in Engineering*, 87(6):541–565, 2011.

[11] C. Armando Duarte, Ivo Babuška, and J. Tinsley Oden. Generalized finite element methods for three-dimensional structural mechanics problems. *Computers & Structures*, 77(2):215–232, 2000.

[12] Thomas-Peter Fries and Ted Belytschko. The extended/generalized finite element method: an overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3): 253–304, 2010.

[13] Seyed Shahram Ghorashi, Navid Valizadeh, and Soheil Mohammadi. Extended isogeometric analysis for simulation of stationary and propagating cracks. *International Journal for Numerical Methods in Engineering*, 89(9):1069–1101, 2012.

[14] Thomas J. R. Hughes, John A. Cottrell, and Yuri Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135–4195, 2005.

[15] Thomas J. R. Hughes, Alessandro Reali, and Giancarlo Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199(5):301–313, 2010.

[16] Grégory Legrain. A NURBS enhanced extended finite element approach for unfitted CAD analysis. *Computational Mechanics*, 52(4):913–929, 2013.

[17] John Lowther and Ching-Kuang Shene. Teaching B-splines is not difficult! *ACM SIGCSE Bulletin*, 35(1): 381–385, 2003.

[18] Nicolas Mo es, John Dolbow, and Ted Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.

[19] Nicolas Moës, Mathieu Cloirec, Patrice Cartraud, and Jean-François Remacle. A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering*, 192(28):3163–3177, 2003.

[20] Nicolas Moës, Eric Béchet, and Matthieu Tourbier. Imposing Dirichlet boundary conditions in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 67(12): 1641–1669, 2006.

[21] Ahmad R. Najafi, Masoud Safdari, Daniel A. Tortorelli, and Philippe H. Geubelle. Shape optimization using a NURBS-based interface-enriched generalized FEM. *International Journal for Numerical Methods in Engineering*, pages n/a–n/a, 2017. ISSN 1097-0207. doi: 10.1002/nme.5482. URL `http://dx.doi.org/10.1002/nme.5482`. nme.5482.

[22] Aurelia Cuba Ramos, Alejandro M. Aragón, Soheil Soghrati, Philippe H. Geubelle, and Jean-François Molinari. A new formulation for imposing Dirichlet boundary conditions on non-matching meshes. *International Journal for Numerical Methods in Engineering*, 103(6):430–444, 2015.

[23] David F. Rogers. *An introduction to NURBS: with historical perspective*. Elsevier, 2000.

[24] Masoud Safdari, Ahmad R. Najafi, Nancy R. Sottos, and Philippe H. Geubelle. A NURBS-based interface-enriched generalized finite element method for problems with complex discontinuous gradient fields. *International Journal for Numerical Methods in Engineering*, 101(12):950–964, 2015.

[25] Ruben Sevilla and Sonia Fernández-Méndez. Numerical integration over 2D NURBS-shaped domains with applications to NURBS-enhanced FEM. *Finite Elements in Analysis and Design*, 47(10):1209–1220, 2011.

[26] Ruben Sevilla, Sonia Fernández-Méndez, and Antonio Huerta. NURBS-enhanced finite element method. In *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006.

[27] Ruben Sevilla, Sonia Fernández-Méndez, and Antonio Huerta. NURBS-enhanced finite element method (NEFEM). *International Journal for Numerical Methods in Engineering*, 76(1):56–83, 2008.

[28] Ruben Sevilla, Sonia Fernández-Méndez, and Antonio Huerta. Comparison of high-order curved finite elements. *International Journal for Numerical Methods in Engineering*, 87(8):719–734, 2011.

[29] Ching-Kuang Shene. B-spline curves: closed curves. `https://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/spline/B-spline/bspline-curve-closed.html`, 2002. Accessed: 2017-03-27.

[30] Choon Fong Shih and Robert J. Asaro. Elastic-plastic analysis of cracks on bimaterial interfaces: Part I – small scale yielding. *Journal of Applied Mechanics*, 55(2):299–316, 1988.

[31] George C. Sih, Paul C. Paris, and Fazil Erdogan. Crack-tip, stress-intensity factors for plane extension and plate bending problems. *Journal of Applied Mechanics*, 29(2):306–312, 1962.

[32] Soheil Soghrati. Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations. *Journal of Computational Physics*, 275:41–52, 2014.

[33] Soheil Soghrati and Ricardo Alvarez Merel. Nurbs enhanced hifem: A fully mesh-independent method with zero geometric discretization error. *Finite Elements in Analysis and Design*, 120:68–79, 2016.

[34] Soheil Soghrati, Alejandro M. Aragón, C. Armando Duarte, and Philippe H. Geubelle. An interface-enriched generalized fem for problems with discontinuous gradient fields. *International Journal for Numerical Methods in Engineering*, 89(8):991–1008, 2012.

[35] Natarajan Sukumar, David L. Chopp, Nicolas Moës, and Ted Belytschko. Modeling holes and inclusions by level sets in the extended finite-element method. *Computer Methods in Applied Mechanics and Engineering*, 190(46):6183–6200, 2001.

[36] Chin-Teh Sun and Zhihe Jin. *Fracture Mechanics*, chapter 3 - The Elastic Stress Field around a Crack Tip, pages 25–75. Elsevier – Academic Press, 12 2012. ISBN 9780123850010.

[37] Barna Szabó, Alexander Düster, and Ernst Rank. *Encyclopedia of Computational Mechanics*, volume 1 - Fundamentals, chapter 5 - The p-version of the Finite Element Method. John Wiley & Sons, 2004.

[38] Marcus H. Y. Tan, Masoud Safdari, Ahmad R. Najafi, and Philippe H. Geubelle. A NURBS-based interface-enriched generalized finite element scheme for the thermal analysis and design of microvascular composites. *Computer Methods in Applied Mechanics and Engineering*, 283:1382–1400, 2015.

[39] J. F. Yau, S. S. Wang, and H. T. Corten. A mixed-mode crack analysis of isotropic solids using conservation laws of elasticity. *Journal of Applied Mechanics*, 47(2):335–341, 1980.