NTNU – Trondheim
Norwegian University of
Science and Technology

Isogeometric analysis of thermoelastic problem

Siri Aakre

Master of Science in Physics and Mathematics
Submission date: August 2015
Supervisor: Trond Kvamsdal, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

NTNU – Trondheim
Norwegian University of
Science and Technology

Isogeometric Analysis of Thermoelasticity

Siri Aakre

Submission date: August 2015
Supervisor: Trond Kvamsdal

Norwegian University of Science and Technology
Department of Mathematical Sciences

Title: Isogeometric Analysis of Thermoelasticity

Student: Siri Aakre

Problem description: Investigate the use of isogeometric methods for addressing thermoelastic problems with varying material parameters. In particular we will study the robustness and efficiency of an isogeometric approach regarding the choice of function spaces for displacement and temperature.

Supervisor: Trond Kvamsdal

Abstract

In this thesis there is given an introduction to isogeometric finite element method for uncoupled thermo elastic problems using non uniform rational B-splines (NURBS) as basis functions. The numerical analysis of the uncoupled thermo elastic problem is done with the new finite element solver IFEM, which is a isogeometric solver under development at SINTEF. The performance of IFEM is tested on a problem with a smooth solution, as well as a singular problem. It is shown that while IFEM, at the current state, achieves an optimal convergence rate for the smooth model; this is not the case for the model containing the singularity.

Sammendrag

I denne oppgaven blir det gitt en introduksjon av isogeometrisk finite element method, og hvordan man løser ukoblede termoelastiske problemer isogeometrisk ved å benytte NURBS som basisfunksjoner. Det vil bli gjort en numerisk analyse av det ukoblede termoelastiske problemet med den nye isogeometriske løseren IFEM som for tiden er under utvikling hos SINTEF. IFEM løseren blir testet på et problem med glatt løsning og et singulært problem. Det viser seg at selv om IFEM oppnår god konvergensrate for det glatte problemet gjør den ikke det for det singulære.

Preface

This Master thesis is my final work for the Master of Science program called Applied Physics and Mathematics, with a specialization in Industrial Mathematics at the Norwegian University of Science and Technology (NTNU). The thesis was completed during the summer of 2015. The reader is expected to have prior knowledge and experience with the finite element method and numerical analysis, but some key concepts are still reviewed.

I had the privilege to test and work with the new isogeometric finite element solver IFEM currently under development at SINTEF in my thesis work, which was both an interesting and challenging task. I would like to thank Yared Worku Bekele, Knut Morten Okstad and Arne Morten Kvarving at SINTEF for showing me the ropes and helping me understand IFEM. I am also grateful to my supervisor Trond Kvamsdal for taking time out of his busy schedule to share of his wast knowledge and help me with my work. I have greatly enjoyed working with my thesis.

Lastly I would like to thank Olav Møyner for being my ”spirit guide” during my years at the university.

Contents

1 Introduction **1**
- 1.1 Computer-aided engineering . . . 1
- 1.2 Computer-aided design . . . 2
- 1.3 New demands and isogeometry . . . 2
- 1.4 Thermoelasticity . . . 4
- 1.5 This thesis . . . 4

2 Finite element theory **7**
- 2.1 Sobolev and Hilbert spaces . . . 7
- 2.2 Finding the variational formulation . . . 9
- 2.3 Galerkin method . . . 10

3 B-splines and NURBS **13**
- 3.1 Knot vectors . . . 13
- 3.2 Basis functions . . . 14
  - 3.2.1 Derivative of basis functions . . . 14
- 3.3 B-splines geometry . . . 14
  - 3.3.1 B-spline curves . . . 15
  - 3.3.2 B-spline surface . . . 15
- 3.4 NURBS . . . 15
  - 3.4.1 Geometric perspective . . . 16
  - 3.4.2 Algebraic perspective . . . 16
  - 3.4.3 Derivatives of NURBS basis functions . . . 17
- 3.5 Knot Insertion . . . 18
- 3.6 Refinement methods . . . 18
  - 3.6.1 h-refinement: knot insertion . . . 18
  - 3.6.2 p-refinement: order elevation . . . 19
  - 3.6.3 k-refinement: higher order and higher continuity . . . 20
- 3.7 An example . . . 20

4 Isogeometric analysis **23**
- 4.1 Isogeometric concept . . . 23

4.2 Spaces and mapping 24
4.3 Isogeometric finite element analysis 26

5 Thermoelasticity **29**
5.1 Heat transfer problem 29
5.1.1 Basic equations of thermal heat transfer 30
5.1.2 The variational formulation 30
5.2 Thermo elasticity problem 33
5.2.1 Basic equations of thermoelasticity 33
5.2.2 The variational formulation 34
5.3 Error estimation 35

6 Numerical examples **37**
6.1 IFEM solver 37
6.2 Model problems 38
6.3 The Annulus: a circular pipe 39
6.3.1 Analytical solution for the annulus 40
6.3.2 Results for the annulus 41
6.4 L-shape: the square pipe 43
6.4.1 Results for the L-shape 44

7 Conclusion **61**
7.1 Future work 61

Bibliography **63**

Chapter 1
Introduction

In the modern world of computer-aided engineering (CAE); particularly finite element analysis (FEA) and computer-aided design (CAD) are essential tools for solving a variety of problems. They both play an important part in developing new technologies and problem simulations. Isogeometric analysis is a computational method that strives to combine both approaches by utilizing the same data set in both design and analysis. The process of integrating these two approaches has, however, not been uncomplicated. The main reason behind the difficulties stems from the fact that CAE and CAD has evolved in two different communities, each with their own background and focus. In this chapter there will be given a brief overview of the development of the two technologies, the difficulties in combining these and the introduction of the isogeometric method.

1.1 Computer-aided engineering

Computer-aided engineering is being applied everywhere engineers are at work. It uses, amongst other things, computer software to perform analysis and simulations. CAE is being used for mathematical computations and for solving a variety of engineering problems. The finite element method (FEM) is very popular in this regard, and it is applied to solve and analyze partial differential equations by utilizing piecewise polynomials. When the finite element method was born in the 1950s [20] calculations were still being done by hand, and because of this the method was limited to solving small and easy systems. The dawn of computers opened doors for a lager variety of problems as it was no longer limited by manpower. This made it possible to analyze increasingly complex models. Today FEM is one of the most popular methods and so well-established that you can find it in almost every aspect of the industry.

1.2 Computer-aided design

Computer-aided design originates back to 1966 [20] and made it possible to utilize computers in design and drawing. It is commonly agreed that it originates back to the work of two French engineers, Pierre Bèzier and Paul de Faget de Casteljau, working for Renault and Citroën respectively. CAD is now a very important tool in industrial and architectural design, as well as other aspects of drawing and design [31]. CAD has evolved much since its beginning, and experienced "a bloom" in the 1970's when it became possible to make computers with a graphical interface. Linear combinations of B-splines started being used to represent curves in 1972 [30], and non-uniform rational B-splines (NURBS) have been popular since 1975 [34]. Today NURBS is one of the most commonly used basis to represent geometries [15, 16, 10]. NURBS has the advantage of being able to represent conic sections like circles and cylinders perfectly without the usual approximation error. This precise representation makes the use of NURBS popular in CAD. While in the beginning NURBS was only being used in packages developed for car companies, CAD tools today are used in all standard CAD packages [31].

1.3 New demands and isogeometry

Since these two technologies have evolved separately in two different communities, with different focus and goals in mind, combining them turned out to be no easy task. The fact that a model has been created and meshed from CAD based geometry does not automatically mean that it is suitable for analysis. As the technologies have matured, the gap between the technologies have only widened. Designers have kept their focus on systems that are easy to manipulate, visualize and construct. Analysts on the other hand, have concentrated on developing systems which are accurate, fast and easy to interpret. Both fields have relied on completely different geometric constructs. However, it eventually became apparent that engineering design and analysis are not separate endeavors. Design needs to be able to communicate with the analysis and vice versa.

As we need to be able to handle increasingly complex constructions, our methods for both design and analysis have to be more efficient and accurate. After designers have generated CAD files for geometry, they need to be made suitable for analysis; meaning that the designs need to be meshed properly in order to make it possible to conduct a large-scale finite element analysis. Creating a model and converting a geometric model from design to analysis is not an easy task. We require a geometry that is suitable for analysis, as well as it sufficiently preserves the physical geometry of the problem. The more complex the model is, the more of the time spent on overall analysis must be devoted to converting the CAD geometry to a mesh that can be used in the analysis. For complex engineering designs this is estimated to take as

much as 80% of the overall time spent on the task. It has been found that just the creation of an analysis suitable geometry uses about 60% of the total time, and mesh construction takes about 20% of time. This leaves only 20% for the actual analysis. In the process of integrating CAD and CAE, many of the difficulties stem from lack of an effective way to create a geometry specific to simulation as well as automated mesh generations. The 80/20 ratio of modeling/analysis is a common experience in the industry according to [20] Thus the motivation for finding a way to combine CAD and CAE has only increased with modern demands. There has been a lot of effort put into combining these, however it has proven to be a large gap to bridge.

While previous attempts at combining these had focused on converting from one system to another, Hughes at the University of Austin had the idea that the problem might lie elsewhere. It became apparent to him that the best way to break down the the barriers between CAD and CAE was not only to reconstitute the process, but also to maintain the compatibility with existing practices. He meant it would be a better idea to focus on the system itself, create and focus on one geometric model. Instead of struggling with bad convergence rates and low precision due to the lack of an accurate mesh when trying to connect the two systems, he thought the focus should be on using the same system for both design and analysis. He believed that the whole basis for the analysis should be the same as for the basis used in the design process, and that the whole basis for analysis should be exchanged for the basis from design. This is what we now refer to as isogeometric design and analysis. It was first introduced in 2005 by Hughes [21]. Since then a number of papers have been written on the isogeometric approach in a variety of fields such as analysis of blood flow [3] and wind-turbines [5], turbulence modeling of incompressible flow [36] and structural vibrations [12].

As previously mentioned, engineering design is dominated by NURBS. They are often more suited to represent the geometry than other polynomials. NURBS are convenient for surface modeling because of their ability to represent conic sections. NURBS represent billions of dollars in development investments and their superiority is undeniable. Working with NURBS as basis functions means that we no longer have to make finite element meshes explicitly, but can use the meshes already made to represent the geometry in the design. By using the same basis and mesh for both the geometry and analysis we get the luxury of working on a precise geometric model. This gives us both faster and more accurate results since we have eliminated the approximations normal in the model, making the results much more precise. Hughes also claimed that using the same system would give more accurate results.

1.4 Thermoelasticity

The study of a solid body under temperature changes as been thoroughly studied in many books and publications for a variety of different media and shapes, both regarding heat conduction and thermo elasticity. Developments in both these fields have resulted in a variety of techniques and solutions [24]. Particularly numerical analysis has been studied and various approaches based on variational methods have been developed for the thermoelastic problem and the methods have been utilized in a variety of applications [25]. In this thesis we will focus on solving the thermoelastic equation with a isogeometric finite element method.

The thermoelastic equation is among other things an important part of the extrusion process of aluminum [26, 9]. Aluminum alloys are increasingly more popular and have a variety of new application areas. It plays amongst other things an important role in aerospace manufacturing, especially since the introduction of metal skinned aircrafts. The extrusion of aluminum is thus an important problem to be solved. Quantitative knowledge about this process is hard to come by (the extrusion process being complicated), making accurate simulations especially important. Since the model of the problem is complex; the process is often costly and cumbersome, making the need for a good numerical simulator important. It both saves time and money, and may give us more information than could be hoped for through physical experiments.

Thermomechanical problems however do not only appear in relation to aluminum, but in a variety of other engineering situations as well.

1.5 This thesis

In this thesis there will be given an introduction to isogeometric finite element method for the thermoelastic problems. It is assumed that the reader is somewhat familiar with classical numerical techniques and finite element analysis, a very short introduction to some key concepts is given in chapter two. Chapter three contains a more detailed introduction to the theory behind B-splines and NURBS, which will form the basis for the isogeometric analysis.

In chapter four the core concepts of isogeometric analysis is introduced and compared it to classical finite element methods. The major differences between the two approaches will be highlighted here. We go on to present the uncoupled thermoelastic problem and its basic equations. A numerical analysis of the uncoupled thermoelastic problem with the new isogeometric finite element solver IFEM, currently under development at SINTEF, will be done in chapter six. We wish to know more about the robustness of the solver and its performance on thermoelastic problems.

In chapter six we therefore define the model problems on which we will test the numerical attributes of IFEM. We also present the results from our testing here. The different model problems are tested with different regularities and polynomial orders. We are interested to see if function spaces affect the performance of the solver, and if the accuracy of the solution is sensitive to the multiplicities of knot lines in the model.

We continue on to a conclusion and finish with a short discussion and suggestions for future work.

Chapter 2
Finite element theory

In this chapter there is given a short summary of some key concepts in the theory behind the finite element method. We assume that the reader is already somewhat familiar with finite element analysis, if not, there is a variety of books that offer deep insight into the method and a more thorough dissertation of the method, for example[6, 7, 28].

2.1 Sobolev and Hilbert spaces

In this chapter there will be given a short introduction to Sobolev and Hilbert spaces, which is the functional spaces that the trial and weighting basis functions, used to find the variational formulations, belong to. A more detailed introduction can be found in chapter 2 in [28].

First we need to remember some concepts from linear algebra.

Letting V be a linear space, then if L is linear and $L : V \to \mathbb{R}$ it is a linear functional on V such that

$$L(\alpha v + \beta u) = \alpha L(v) + \beta L(u) \quad \forall \alpha, \beta \in \mathbb{R}$$

A bilinear functional, $a(\cdot, \cdot)$ on $V \times V$, is $a : V \times V \to \mathbb{R}$ linear in both arguments. A bilinear functional which is symmetric, i.e

$$a(u, v) = a(v, u) \quad \forall u, v \in V$$

and has a bilinear symmetric form defines an inner product on V if

$$a(v, v) > 0 \quad \forall v \in V, v \neq 0$$

The norm associated with an inner product is given by

$$\|v\| = (a(v, v))^{\frac{1}{2}} \quad v \in V$$

If the Cauchy- Schwartz Inequality holds

$$|a(u,v)| \leq \|u\|\|v\| \quad u,\, v \in V$$

then the space, V, is complete with respect to the norm if all Cauchy sequences in V converges with respect to that norm.

If a normed linear space V, is also complete it is called a Hilbert space. The Hilbert space H is a linear space with the inner product $(\cdot,\cdot)_H$ and with the norm $\|f\|_H \equiv \sqrt{(f,f)}$.

Considering the space of square-integrable functions

$L^2(\Omega) = \{f : \Omega \to \mathbb{R} \text{ s.t } \int_\Omega f^2 \, d\Omega < \infty\}$

The space L^2 is a Hilbert space whose inner product is defined as

$$(f,g)_{L^2(\Omega)} = \int_\Omega fg \, d\Omega$$

with the norm in L^2 is induced by this inner product

$$\|f\|_{L^2} = \left((f,f)_{L^2(\Omega)}\right)^{\frac{1}{2}} = \left(\int_\Omega f^2 \, d\Omega\right)^{\frac{1}{2}}$$

These results can be generalized for an arbitrary p by defining $L^p(\Omega) = \{f : \Omega \to \mathbb{R} \text{ s.t } \int_\Omega |f|^p \, d\Omega < \infty\}$ with the induced norm

$$\|f\|_{L^p} = \left(\int_\Omega |f|^p \, d\Omega\right)^{\frac{1}{p}}$$

In an $L^2(\Omega)$ space we are not guaranteed that the derivatives (in the sense of distribution) are still in $L^2(\Omega)$. In a Sobolev space however, we have more knowledge about the behavior of the derivatives. Introducing a multi-index notation for the derivatives by defining the vector $\alpha = (\alpha_1 ... \alpha_n)$. It is an n-tuple of non-negative integers and $f : \Omega \to \mathbb{R}$ be a function defined in $\Omega \subset \mathbb{R}$. We can write derivatives as

$$D^\alpha f(\mathbf{x}) = \frac{\partial^{|\alpha|} f(\mathbf{x})}{\partial x_1^{\alpha_1} ... \partial x_n^{\alpha_n}}$$

Letting Ω be an open set of $\mathbb{R}$, k a positive integer and $1 \leq p \leq \infty$. We introduce Sobolev space of order k on Ω on the space of functions $v \in L^p(\Omega)$ such that all their (distributional) derivatives of v up to order k belong to $L^p(\Omega)$.

$$W^{k,p}(\Omega) = \{v \in L^p(\Omega) : D^\alpha v \in L^p(\Omega) \quad \forall \alpha : |\alpha| \leq k\}$$

For $1 \leq p \leq \infty$ this is a Banach space with norm

$$(u, v)_{W^{k,p}} = \sum_{|\alpha| \leq k} \int_\Omega (D^\alpha u)(D^\alpha v) \, d\Omega$$

and the norm

$$\|u\|_{W^{k,p}(\Omega)} = \left(\sum_{|\alpha| \leq k} \int_\Omega |(D^\alpha u)|^p \, d\Omega \right)^{\frac{1}{p}}$$

Note that for $k = 0$, $W^{k,p}(\Omega) = L^p(\Omega)$ and that for $p = 2$, $W^{k,2}(\Omega) = H^k(\Omega)$

2.2 Finding the variational formulation

In this section we give a short example of the finite element method applied on the classic model problem of the Poisson equation with mixed boundary conditions, in the example we find the variational formulation of a strong problem. A more detailed example can be found in chapter 3 in [28]. Here Γ_D is Dirichlet boundary condition and Γ_N are the Neumann boundary condition. The Dirichlet conditions are what is called essential boundary conditions, they are imposed explicitly in the same functional space as the problem. Neumann boundary conditions are called natural and they are satisfied implicitly by the solution. Together they yield a partition of $\partial\Omega$.

Given a domain $\Omega \subset \mathbb{R}^2$ and continuous function $f : \Omega \to \mathbb{R}$, we want to find a function $u : \Omega \to \mathbb{R}$ such that satisfies equations below.

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \Gamma_D \\ \frac{\partial u}{\partial n} &= \phi \quad \text{on } \Gamma_N \end{aligned}$$

This problem is known as a strong formulation on the Poisson problem. When rewriting the problem to its weak formulation, or variational formulation, we start by multiplying the equation with a test function $v \in \mathcal{V}_0$. Denoting the space $\mathcal{V}_0$ as $\mathcal{V} = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$ and then integrate over the domain

$$\int_\Omega v \cdot \nabla^2 u \, d\Omega + \int_\Omega vf \, d\Omega = 0$$

by utilize Green's formula on the first integral, we end up with

$$\int_\Omega \nabla u \cdot \nabla v \, d\Omega - \int_{\partial\Omega} v \cdot \nabla u \, d\gamma = \int_\Omega vf \, d\Omega$$

We recall that $\frac{\partial u}{\partial n} = \phi$ on Γ_N, and by exploiting the additivity of integrals we obtain

$$\int_\Omega \nabla u \cdot \nabla v \; d\Omega - \int_{\Gamma_D} v \cdot \nabla u \; d\gamma - \int_{\Gamma_N} \phi v \; d\gamma = \int_\Omega v f \; d\Omega$$

By imposing the fact that the test function we have selected vanishes on Γ_D. The weak formulation of the Poisson problem with mixed boundary conditions consist of finding $u \in \mathcal{V}_g$

$$\int_\Omega \nabla u \cdot \nabla v \; d\Omega = \int_\Omega v f \; d\Omega + \int_{\Gamma_N} \phi v \; d\gamma \quad \forall v \in \mathcal{V}$$

where $\mathcal{V}_g$ is a set $\mathcal{V}_g = \{u \in H^1(\Omega) : u|_{\Gamma_D} = g\}$

However, in order to get a symmetric problem we need to introduce a lifting of the boundary data so that the (new) unknown solution space coincides with the test function. We suppose to know a continuous function R_g, such that $R_g \in H^1(\Omega)$, $R_g|_{\Gamma_D} = g$. We define $\mathring{u} = u - R_g$ and $\nabla u = \nabla \mathring{u} + \nabla R_g$. The weak formulation for the problem then consists of finding $\mathring{u} \in \mathcal{V}_0$ such that

$$\int_\Omega \nabla \mathring{u} \cdot \nabla v \; d\Omega = \int_\Omega v f \; d\Omega + \int_{\Gamma_N} \phi v \; d\gamma - \int_\Omega \nabla R_g \cdot \nabla v \; d\Omega \quad \forall v \in \mathcal{V}$$

Remember that we have denoted the space $\mathcal{V}_0 = H^1_{\Gamma_D}(\Omega)$. The weak formulation can be rewritten into a more compact way by introducing the following bilinear form

$$a(\mathring{u}, u) = \int_\Omega \nabla \mathring{u} \cdot \nabla v \; d\Omega$$

and the following linear functional

$$F(v) = \int_\Omega v f d\Omega + \int_{\Gamma_N} \phi v d\gamma - \int_\Omega \nabla R_g \cdot \nabla v \; d\Omega$$

Giving us the variational formulation of the Poisson problem

$$\text{find } \mathring{u} \in \mathcal{V} : \quad a(\mathring{u}, u) = F(v) \quad \forall v \in \mathcal{V}$$

2.3 Galerkin method

To illustrate the Galerkin method we start with a somewhat simpler version of the Poisson problem with Dirichlet boundary conditions, with the weak form is defined as

$$\text{find } u \in \mathcal{V} : \quad a(u, v) = F(v) \quad \forall v \in \mathcal{V}$$

where $\mathcal{V}$ is an Hilbert space, subspace of $H^1(\Omega)$

When preforming an approximation via Galerkin method we start by defining a proper finite-dimensional subspace

$$\mathcal{V}_h \subset V, \quad \dim \mathcal{V}_h < \infty \quad \forall h > 0$$

The Galerkin problem takes the form

$$\text{find } u_h \in \mathcal{V}_h : \quad a(u_h, v_h) = F(v_h) \quad \forall v_h \in \mathcal{V}_h$$

Letting the functions $\{\mathcal{N}_1 \ ... \ \mathcal{N}_{n_h}\}$ form the basis of $\mathcal{V}$, meaning that all the functions in the space $\mathcal{V}_h$ are a linear combinations of $\mathcal{N}$. It is required that

$$a(u_h, \mathcal{N}_i) = F(\mathcal{N}_i) \quad i = 1, 2, \dots n_h$$

and since $u_h \in \mathcal{V}_h$

$$u_h(\mathbf{x}) = \sum_{j=1}^{n_h} u_j \mathcal{N}_j(\mathbf{x}), \text{ where } u_j, j = 1, \dots, n_h$$

giving us the equations

$$\sum_{j=1}^{n_h} u_j a(\mathcal{N}_j, \mathcal{N}_i) = F(\mathcal{N}_i), \quad i = 1, \dots, n_h$$

Defining a stiffness matrix, $\mathbf{A}$, with the elements $a_{ij} = a(\mathcal{N}_j, \mathcal{N}_i)$. A vector $\mathbf{f}$ with components $f_i = F(\mathcal{N}_i)$. By letting $\mathbf{u}$ be a vector containing the unknown coefficients u_j. Combining these we get the linear system

$$\mathbf{Au} = \mathbf{f}$$

Not only does the Galerkin method guarantee a stable discrete solution u_h, it can also be proved that the solution of the Galerkin problem exists and is unique. The weak solution of the problem, u_h, also converges to the solution of u when $h \to 0$.

Chapter 3
B-splines and NURBS

B-splines are built from a set of polynomial functions defined on non-overlapping connected intervals. They consist of n piecewise polynomial basis functions of degree p. B-splines are smooth, differentiable and continuous functions within each of these subintervals, although they have limited continuity and differentiability at the interval boundaries. The information about how these subintervals are connected is given in a knot vector. This chapter provide an introduction to B-splines and their non uniform national form, NURBS.

3.1 Knot vectors

Each dimension has a knot vector which is a non-decreasing set of coordinates in the parameter space defined by a set of knots (coordinates) that gives the information of the location of the interval boundaries, written as $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$. Here $\xi_i \in \mathbf{R}$ is the ith knot, p is the polynomial order, and n is the number of B-spline basis functions constructed from the knot vector. Note that it is the relative difference between the knots that determines the B-spline functions, not the location of the knots themselves.

Knot values may be repeated, and a knot vector where the first and last knot appear $p + 1$ times is called an open knot vector. In an open knot vector the basis functions interpolate the knots at the beginning and the end.

As mentioned above, the knots partition the parameter space into elements. The knot spans will be the element domains and the basis functions will be C^∞ within an element. The basis functions have C^{p-m} continuity, where p is the degree of the polynomial and m is the knot's multiplicity, over the element domains. From this we see that repeated knots reduce the continuity of the basis functions across than particular knot. This can be seen in Figure 3.3

3.2 Basis functions

The basis functions are created recursively with of Cox-de Boor formula, [20] , using the knot spans from the knot vector.

For $p = 0$ we define the basis function has a

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

While for $p = 1, 2, ...$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1} + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{3.2}$$

If we have repeated knots in the knot vector the denominator of these coefficients will be zero. This problem is solved by defining the whole term as the value zero when the situation occurs.

Two important features of the B-spline basis functions are that they are non-negative and sums to one everywhere, i.e

$$N_{i,p}(\xi) \geq 0$$
$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1$$

and together they form a basis, which means they are linearly independent. It is also important to note that B-splines have a local support and that this support is always limited to $p + 1$ knot spans for B-spline functions of order p. Thus B-spline $N_{i,p}$ only depends on the knots (ξ_i, ξ_{i+p+1}).

The notation $N_{i,p}$ refers to the ith basis function of order p, with $i \in [1, n]$.

3.2.1 Derivative of basis functions

In the analysis we also need the derivative of the basis function. For the polynomial of order p the derivative of the ith basis function is given by

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1} - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \tag{3.3}$$

3.3 B-splines geometry

In order to move the B-splines from the parameter space, where they are defined, into the physical space we build B-spline curves or surfaces. These are made by linear

combinations of the B-spline basis functions and the control points. The control points themselves represent the geometry, and are points in the physical space. The curve or surface will be the mapping from the parameter space to the physical space. The notion of control points and spaces will be explained more thoroughly in the next chapter.

3.3.1 B-spline curves

Given n basis functions, $N_{i,p}$, and corresponding control points $\mathbf{B}_i \in \mathbb{R}^d$ with $i = 1, 2, ..., n$, a piecewise-polynomial B-spline curve is given by

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{B}_i \tag{3.4}$$

The B-spline curve is not required to interpolate at the control points, but it may be forced to by utilizing the fact that we can control the continuity of the basis function (and by extension the B-spline curve) across a knot. Since the continuity across the knot is determined by C^{p-m}, using a knot vector with the right multiplicity ($m = p$) can ensure C^0 continuity across a particular knot, which will make the curve interpolate.

3.3.2 B-spline surface

The curve can be extended to a surface. With a given control net $B_{i,j}$, $i = 1, 2, ..., n$, $j = 1, 2, ..., m$, polynomial orders p and q and two knot vectors $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$ and $\mathcal{H} = \{\eta_1, \eta_2, ..., \eta_{m+q+1}\}$, we can create a tensor product B-spline surface with a formula

$$\mathbf{S}(\xi, \eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M(\eta)_{j,q}\mathbf{B}_{i,j} \tag{3.5}$$

where $N_{i,p}(\xi)$ $M_{j,q}(\eta)$ are univariate B-spline basis functions of order p and q, that corresponds to knot vectors Ξ and $\mathcal{H}$ respectively.

This tensor product is non-negative and forms a partition of unity as $\forall(\xi, \eta) \in [\xi_1, \xi_{n+p+1}] \times [\eta_1, \eta_{m+q+1}]$

$$\sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta) = \Big(\sum_{i=1}^{n} N_{i,p}(\xi)\Big)\Big(\sum_{j=1}^{m} M_{j,q}(\eta)\Big) = 1$$

3.4 NURBS

Non uniform rational B-splines, NURBS for short, are built from B-splines. They are piecewise rational polynomials with non uniform knot vectors. They are rational as a result of the basis functions being multiplied by a weighting factor and divided

by the sum of the B-splines basis functions multiplied by the same weights. NURBS retain all the properties of a B-spline. By using NURBS as basis we gain control of the continuity across the element edges that we usually lack in classical FEA.

3.4.1 Geometric perspective

In a geometrical sense NURBS are regular B-spline curves projected to a space of one order lower. If we assume that we have the B-spline curve $\mathbf{C}^w(\xi)$, the projective curve, with the associated projective control points $\mathbf{B}_i^w \in \mathbf{R^{d+1}}$, then the control points for the NURBS curve are given by

$$(\mathbf{B}_i)_j = \frac{(\mathbf{B}_i^w)_j}{w_i} \quad j = 1,...,d$$
$$w_i = (\mathbf{B}_i)_{d+1})$$

Dividing the projective NURBS control points by the weights is the same as applying a projective transformation on them. The transformation should be done to every point in the curve. This is accomplished by defining a weighting function

$$W(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) w_i$$

here $N_{i,p}(\xi)$ is the B-spline basis functions. We can now define the NURBS curve as

$$(\mathbf{C}(\xi)_i)_j = \frac{(\mathbf{C}^w(\xi))_j}{W(\xi)} \quad j = 1,...,d$$

where $\mathbf{C}(\xi)^w = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{B}_i^w = \sum_{i=1}^{n} N_{i,p}(\xi)\mathbf{B}_i w_i$

3.4.2 Algebraic perspective

In order to ensure that the properties of the B-spline functions will be retained in the NURBS the basis for the NURBS space is constructed from the knot vectors, and the curves or surfaces are built from linear combinations of basis functions and control points.

With this in mind we can define the NURBS basis functions as

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) w_i}{W(\xi)}$$

where $W(\xi)$ is the weighting function defined in the previous section. NURBS curves are given by

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\mathbf{B}_i$$

In 2D the NURBS basis functions will be given as a tensor product

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{W(\xi,\eta)}$$

where the weighting function is defined as

$$W(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}$$

and the 2D surfaces will in turn be given by

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\mathbf{B}_{i,j}$$

3.4.3 Derivatives of NURBS basis functions

In our finite element analysis we need the derivatives of the NURBS basis functions. The derivatives are found by differentiating the basis functions using the quotient rule.

$$\frac{d}{d\xi}R_i^p(\xi) = w_i\frac{\Big(\frac{d}{d\xi}N_{i,p}(\xi)\Big)W(\xi) - N_{i,p}(\xi)\Big(\frac{d}{d\xi}W(\xi)\Big)}{(W(\xi)^2)}$$

where the derivative of the weighting function is

$$\frac{d}{d\xi}W(\xi) = \sum_{i=1}^{n}\Big(\frac{d}{d\xi}N_{i,p}(\xi)\Big)w_i$$

In 2D we need to differentiate with respect to ξ and η respectively

$$\frac{d}{d\xi}R_{i,j}^{p,q}(\xi,\eta) = w_i\frac{\Big(\frac{d}{d\xi}N_{i,p}(\xi)\Big)M_{j,q}(\eta)W(\xi,\eta) - N_{i,p}(\xi)M_{j,q}(\eta)\Big(\frac{d}{d\xi}W(\xi,\eta)\Big)}{(W(\xi,\eta)^2)}$$

$$\frac{d}{d\eta}R_{i,j}^{p,q}(\xi,\eta) = w_i\frac{N_{i,p}(\xi)\Big(\frac{d}{d\eta}M_{j,q}(\eta)\Big)W(\xi,\eta) - N_{i,p}(\xi)M_{j,q}(\eta)\Big(\frac{d}{d\eta}W(\xi,\eta)\Big)}{(W(\xi,\eta)^2)}$$

with the derivatives of the weighting functions given as

$$\frac{d}{d\xi}W(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\Big(\frac{d}{d\xi}N_{i,p}(\xi)\Big)M_{j,q}(\eta)w_{i,j}$$

$$\frac{d}{d\eta}W(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m}N_{i,p}(\xi)\Big(\frac{d}{d\eta}M_{j,q}(\eta)\Big)w_{i,j}$$

3.5 Knot Insertion

A geometry based on NURBS can be refined while the geometry remains intact. This is useful when faced with problems that require increased precision in certain points, for example when working with singularities. Knot insertion is a way of enriching the basis without changing or compromising the geometry. Additional knots are inserted into the knot vector, thus making the knot spans smaller and shrinking the elements without changing the curve geometrically or parametrically. With the original knot vector as previously described as $\Xi = \xi_1, ..., \xi_{n+p+1}$ the goal is to create an extended knot vector $\overline{\Xi} = \xi_1, ..., \xi_{n+m+p+1=\xi_{n+p+1}}$ and use the extended knot vector to form new $n + m$ basis functions. We also get $n + m$ new matching control points, $\overline{\mathcal{B}}$, formed from linear combinations of original control points $\mathcal{B}$.

$$\overline{\mathcal{B}} = \mathbf{T}^p \mathcal{B} \tag{3.6}$$

where

$$T_{i,j} = \begin{cases} 1 & \overline{\xi_i} \in [\xi_j, \xi_{i+1}) \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

and

$$T_{ij}^{q+1} = \frac{\overline{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} T_{i,j}^q + \frac{\xi_{j+q+1} - \overline{\xi}_{j+q}}{\xi_{j+q+1} - \xi_{j+1}} T_{ij+1} \quad q = 0, 1, 2, ..., p-1 \tag{3.8}$$

3.6 Refinement methods

As previously mentioned, the isometric mesh does not only represent the exact geometry, but it also allows us to refine it to any level without ever altering the geometry in any way [11]. The main strategies of refinement are presented below. Both h-refinement and p-refinement may be familiar as it exists in classic FEA as well, but k-refinement is a more efficient and higher order concept new for Isogeometric analysis. In [21] k-refinement is described as a (potentially) superior approach to high precision analysis.

3.6.1 h-refinement: knot insertion

h-refinement is a refinement method based on knot insertion. As mentioned in chapter 3.5 knots can be inserted without changing the curve (not geometrically nor parametrically). Inserting new knot values splits the existing elements into smaller ones, while repeating knots reduces the continuity of the basis. One can retain the continuity of the curve by selecting control points as described in equations 3.6, 3.7, 3.8. Note that if an (internal) knot appears more that p times it will cause the curve to become discontinuous.

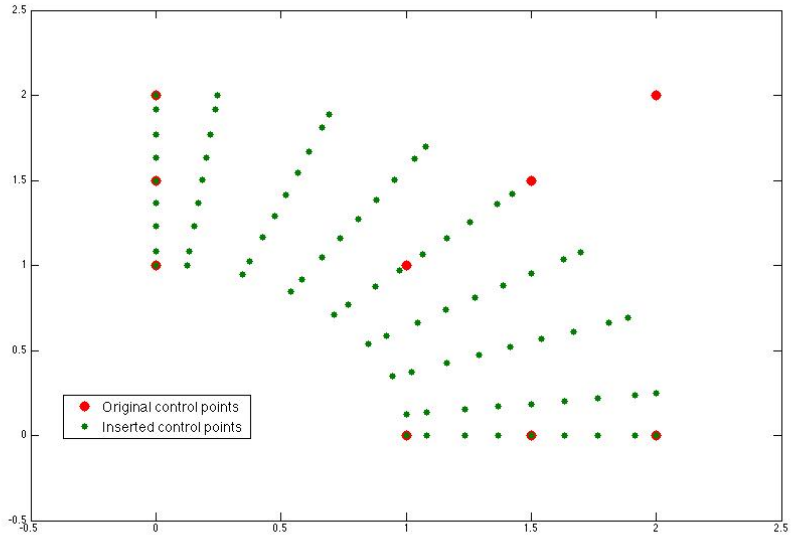

Figure 3.1: This figure shows the an example of knot insertion. Here red knots are the original knots, and the green knots represents the inserted knots.

The h-refinement strategy of inserting knots is a way to enrich the solution space by adding additional basis functions of the same order, at the same time as leaving the curve unchanged.

3.6.2 p-refinement: order elevation

The polynomial order, p, of the basis functions can also be raised without changing the geometry or parameterization. The p-refinement method is based on order elevation (in CAD community often called degree elevation). It entails raising the polynomial order of the basis, and thus the solution space. As previously mentioned the basis functions are $p - m$ continuous across the element boundaries. It follows that when the order, p, is increased we also need to change the multiplicity m in order to retain the original continuity. When elevating the order the multiplicity of each knot value is increased by one, no new knots are added. The multiplicities of the knots is increased but no new elements are introduced.

The p-refinement process starts with replicating the knots until their multiplicity is equal to the polynomial order, practically subdividing the curve. This process effectively divides the curve into many Bézier curves by knot insertion (for more

information of Bèzier curves see [31]). Then the order of the polynomial is elevated on each of these individual segments and lastly we remove the unnecessary knots in order to combine the segments into one. The result is one order elevated B-spline curve. There has been done extensive research on different p-refinement methods and there exists several different algorithms for this purpose. A deeper and more thorough treatment of the subject can be found in [27].

3.6.3 k-refinement: higher order and higher continuity

With NURBS as basis functions we can define new elements by inserting new knot values, for which the continuity across the boundary will be C^{p-m}. We can lower the continuity of the basis across the element boundary by repeating existing knot values.

Consider a unique knot value, $\bar{\xi}$, inserted into a knot vector. It is inserted between to knot values in a curve with order p, this means that the number of continuous derivatives at the new knot is $p-1$. Then the order is elevated to a higher order, q, and the multiplicity of every knot is incremented $(q-p)$ times which means that the discontinuities are preserved in the pth derivative of the basis. The basis will still have $p-1$ continuous. In other words, even though the order of $\bar{\xi}$ is q, the basis only has $p-1$ continuous derivatives at the knot value. If the procedure is done the other way around, i.e we started by elevating the order of the original knot vector to q and then inserted the new knot $\bar{\xi}$, the basis would have $q-1$ continuous derivatives at this knot. This last way of elevating the order and achieving higher continuity is referred to as k-refinement. As we need quadratic level NURBS to represent conic structures perfectly, one of the great strengths of the polynomial and key feature of the isogeometric method, k-refinement is an important concept.

3.7 An example

Figure 3.2 depicts the B-splines created from the open knot vector $\Xi = [0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1]$ of polynomial order 2. Since the knot vector contains 9 knots, we end up with $n = 9 - 2 - 1 = 6$ basis functions to construct the B-spline curve. There are no repeated interior knots so the basis functions are C^1-continuous across these knot spans.

This is not the case however in Figure 3.3 which depicts the basis functions created from knot vector $\Xi = [0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1]$, where knot $\frac{1}{4}$ is repeated. In this case the basis functions will be C^0-continuous across the interior knot span, since $p - m = 2 - 2 = 0$. In both figures the basis is an interpolation at the boundary of the domains this is a result of the knot vectors being open. With the basis functions and knot vector as in figure 3.2 , control points $\mathbf{B_x} = [0, 6, 4, 4, 2, 1]$ and

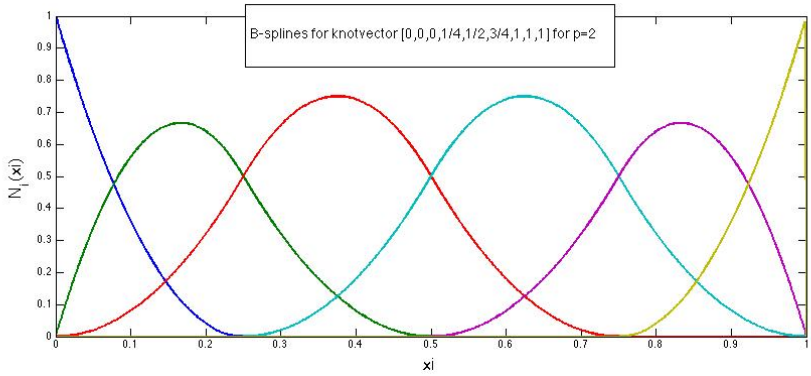

Figure 3.2: These are the B-splines created from the open uniform knot vector $\Xi = [0, 0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1]$ with p=2

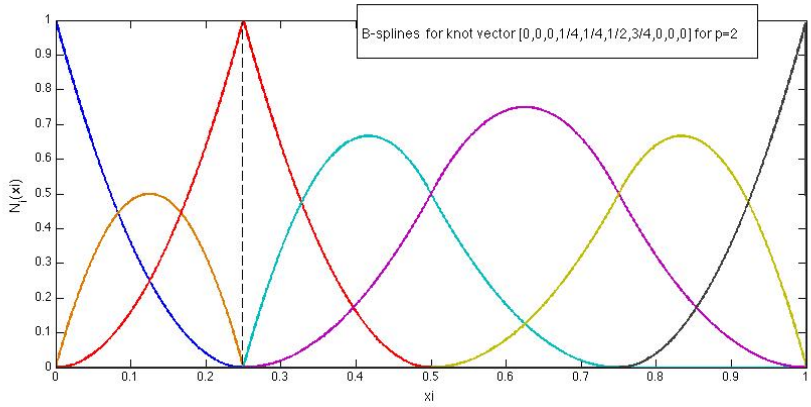

Figure 3.3: These are the B-splines created from the open non-uniform knot vector $\Xi = [0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1, 1]$ with p=2. Note the reduced continuity of the basis functions across the repeated knot(dotted line)

$\mathbf{B_y} = [1, 2, 5, 3, 0]$ and weights equal to one we can create the first NURBS curve in Figure 3.4, since the weights are one this is also a B-spline curve.

It can be seen how the choice of control points effects this curve in the second plot in Figure 3.4, which shows the NURBS curve of the same knot vector as in Figure 3.2 but with different control points. Different control points means a different geometry.

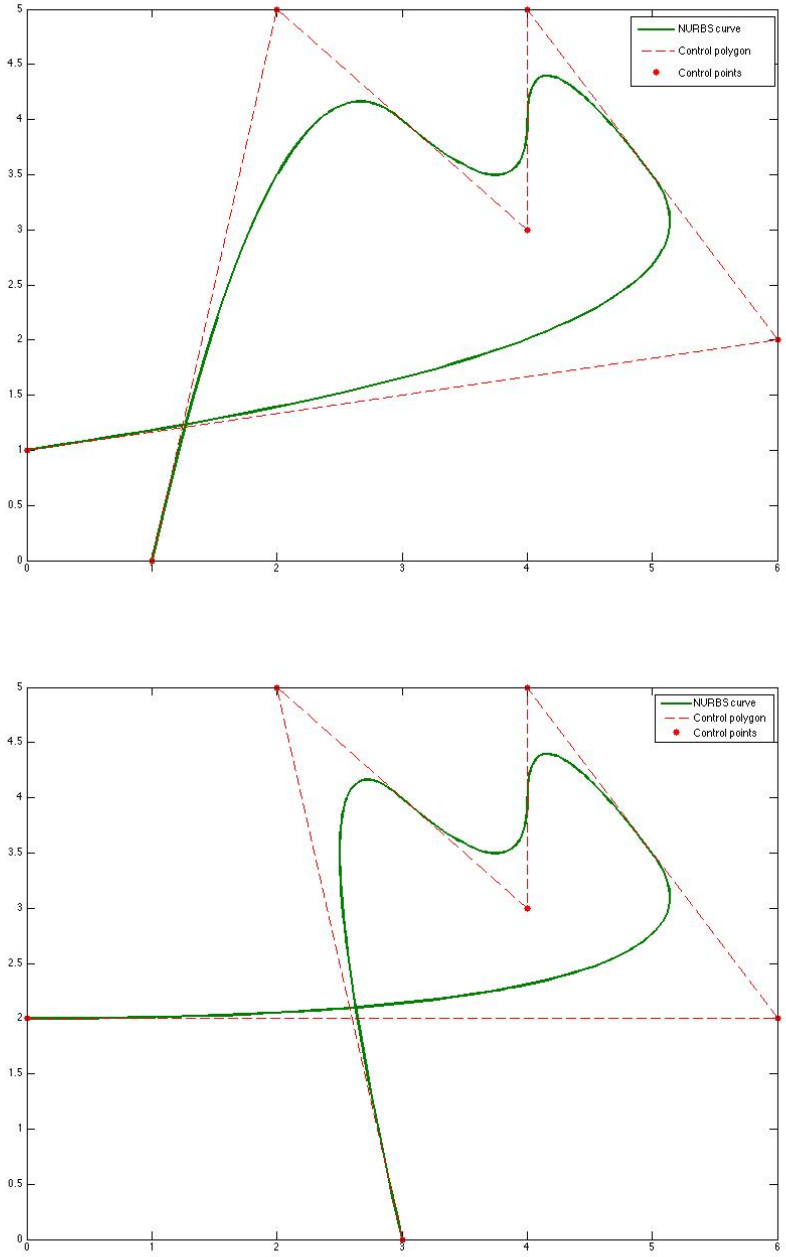

Figure 3.4: The NURBS curves for the same knot vectors, but different control points.

Chapter 4
Isogeometric analysis

In this chapter we will present the isoparametric concept and its differences from the classic finite element method.

While in classical finite element analysis the parameter space is local to individual elements, the parameter space of the B-splines is defined to be local to a patch rather than a single element. A patch can be viewed as a sub domain comprised of multiple uniform elements, and the patch is partitioned into elements by a set of internal knots. The elements will thus be equal to the spans of the knots in the parameter space. A B-spline mapping takes the patch from the parameter space into the physical space. In [20] each element in the physical space is described as the image of a corresponding element in the parameter space, but that the mapping itself is global to the whole patch, rather than the element themselves. These different spaces and mappings will also be discussed later in this chapter.

4.1 Isogeometric concept

The isoparametric concept is well known in classic FEA. It consists of using the same basis both to model the geometry as well as span out the solution space. The basis is chosen to approximate the unknown solution field, and then this basis is used to approximate the already known geometry. In isogeometric analysis however, the focus lies on selecting a basis which is able to exactly represent the known geometry and use this as a basis for our finite element analysis. Such that we can go directly from design to analysis without compromising (the integrity of) the geometry. In [20] Hughes describes it as reversing the isoparametric arrow. Classical FEA imposes a chosen solution space onto the description of the geometry, while in isogeometric analysis this is reversed. Here we begin with a basis chosen for its capability to represent the exact geometry, and impose it on the solution fields.

Since spline basis functions are numerically superior when it comes to solving problems on difficult geometries such as cones and circles, B-splines and NURBS is

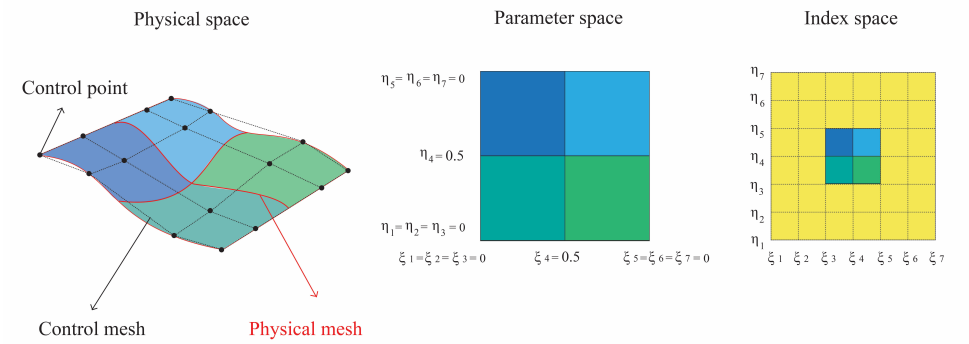

Figure 4.1: The figure illustrates the different spaces used in a NURBS single-patch geometry for the open knot vectors $\Xi = [0, 0, 0, \frac{1}{2}, 1, 1, 1]$ and $\mathcal{H} = [0, 0, 0, \frac{1}{2}, 1, 1, 1]$. In the physical space the physical mesh is shown with the red lines, the control mesh is the dotted line and the control points are shown as black dots. The quadratic basis functions are defined in the parameter space. In the index space the zeros measure knot spans are colored yellow, these are only visible in this space.

often used to express the geometry exactly and to span the solution space in the finite method.

In this report NURBS are used as basis functions to represent the geometry as well as the finite element analysis such that we can get a seamless transfer between design and analysis.

4.2 Spaces and mapping

One of the differences between classical FEM and isogeometric analysis is the meshes and spaces used. This chapter will try to make the distinction between the different meshes, spaces and the mapping between them clear. There are two different meshes, physical mesh and control mesh, where the geometry is determined. Even though the basis functions have support over multiple elements and the solution vector u contains contributions from different basis functions, this support is viewed as local.

As can be seen in figure 4.1 we are working in three different spaces. The physical space, the parameter space (where the basis functions) are defined and the elements are given and the index space of the patch (where each knot is uniquely defined).

During the calculation of the stiffness matrix the integrals are evaluated utilizing Gaussian quadrature in the parent element. In order to do this we need to move from the physical space to the unit square where the integration is preformed. This

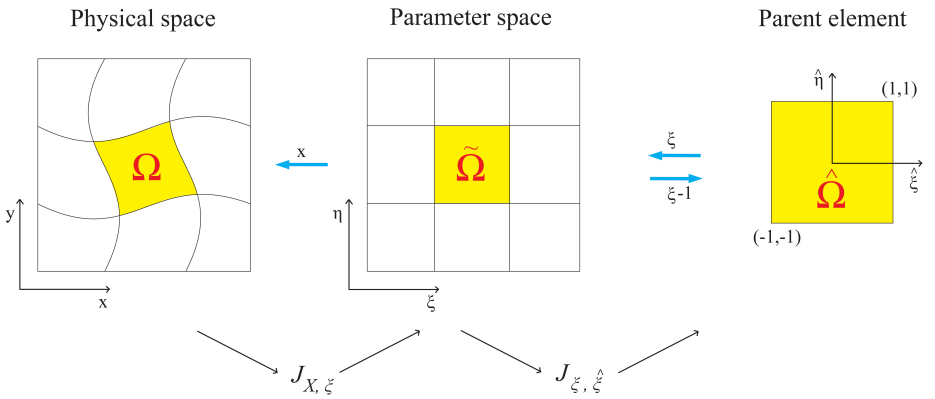

Figure 4.2: The figure illustrates the movement between the spaces.

Physical mesh	A decomposition of the geometry, the mesh is divided into non overlapping physical elements. While in classic FEA the domain is divided into elements by nodal points, when we use NURBS the domain is divided into elements by the knot spans.
Control mesh	The control mesh is defined and interpolated by the control points.

is done by "mapping through" the parameter space. See figure 4.2.

With the knot vectors $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$ and $\mathcal{H} = \{\eta_1, \eta_2, ..., \eta_{m+q+1}\}$, the mapping from the parent element to parameter space is given by

$$\xi = \begin{bmatrix} \xi(\hat{\xi}) \\ \eta(\hat{\eta}) \end{bmatrix} = \begin{bmatrix} \frac{(\xi_{i+1}-\xi_i)\hat{\xi}+(\xi_{i+1}+\xi_i)}{2} \\ \frac{(\eta_{i+1}-\eta_i)\hat{\eta}+(\eta_{i+1}+\eta_i)}{2} \end{bmatrix} \tag{4.1}$$

and the Jacobian matrix for this mapping is

$$\mathbf{J}_{\xi,\hat{\xi}}(\hat{\xi},\hat{\eta}) = \begin{bmatrix} \frac{\partial\xi}{\partial\hat{\xi}} & \frac{\partial\xi}{\partial\hat{\eta}} \\ \frac{\partial\eta}{\partial\hat{\xi}} & \frac{\partial\eta}{\partial\hat{\eta}} \end{bmatrix} = \begin{bmatrix} \frac{(\xi_{i+1}-\xi_i)}{2} & 0 \\ 0 & \frac{(\eta_{i+1}-\eta_i)}{2} \end{bmatrix} \tag{4.2}$$

The next step is to calculate the Jacobian to map from a point in the physical

space to the parameter space

$$\mathbf{J}_{x,\xi} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{4.3}$$

To map from Ω to $\hat{\Omega}$ two Jacobians is combined

$$\mathbf{J}_{x,\hat{\xi}} = \begin{bmatrix} \frac{\partial x}{\partial \hat{\xi}} & \frac{\partial x}{\partial \hat{\eta}} \\ \frac{\partial y}{\partial \hat{\xi}} & \frac{\partial y}{\partial \hat{\eta}} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial \xi}{\partial \hat{\xi}} & \frac{\partial \xi}{\partial \hat{\eta}} \\ \frac{\partial \eta}{\partial \hat{\xi}} & \frac{\partial \eta}{\partial \hat{\eta}} \end{bmatrix} = \mathbf{J}_{x,\xi}\mathbf{J}_{\xi,\hat{\xi}} \tag{4.4}$$

It is a bit cumbersome to find the derivatives with regards to x and y since the basis functions are defined in the parameter space. To find the expression for $\nabla N_i(x, y)$ start with the inverse and transpose of Jacobian of the mapping from the physical space to the parameter space.

$$(\mathbf{J}_{x,\xi})^{-T} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \tag{4.5}$$

and we let

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \quad \hat{\nabla} = \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{bmatrix}$$

combining this we get

$$\nabla = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \end{bmatrix} = (\mathbf{J}_{\mathbf{x},\xi})^{-T}\hat{\nabla}$$

When calculating the derivatives of the basis functions with regards to the x and y, $\mathbf{G} = (\mathbf{J}_{\mathbf{x},\xi})^{-T}$ is used and gives the derivatives of the basis functions in the physical space.

$$\nabla N_i(x, y) = \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i(\xi,\eta)}{\partial \xi} \\ \frac{\partial N_i(\xi,\eta)}{\partial \xi} \end{bmatrix} = \mathbf{G}\hat{\nabla}N_i(\xi, \eta) \tag{4.6}$$

4.3 Isogeometric finite element analysis

In this paper we focus on an isogeometric approach to the finite element method and in this chapter we will try to highlight the differences from the classical approach.

The finite elements are usually built by piecewise polynomials, as the reader might know in classical FEM they are often Lagrange or Hermite polynomials [19].

Lagrange elements, made from tensor products of Lagrange interpolation polynomials, are very popular in FEA. However, when preforming a isogeometric analysis the elements are usually built by splines, in our case NURBS. It is good to note that when B-splines (more precisely Bernstein polynomials) form the basis for an element with C^0 continuity it spans the same space as Lagrange elements, even though the basis functions are different. These elements are called Bézier elements and they give the exact same results as classic Lagrange elements while still processing all the properties of B-splines. It means that anything that can be done with Lagrange elements in classic finite element analysis is also possible to do with B-splines and isogeometry.

An analysis based on NURBS contains much of the same features as a classic finite element approach.There is however some items and features that makes the framework for a NURBS based analysis that differs somewhat from the traditional approach. The biggest differences between the methods are summarized in table 4.3. As previously mentioned the mesh for a NURBS patch is defined by the product of knot vectors and the domain is subdivided by knot spans into elements. The control points defines the geometry. Note also that the definition of a finite element in an isogeometric setting the element domain refers to a patch and when it comes to B-splines and NURBS an element is defined as each tensor product knot span.

There is also a practical difference in the architecture of the code when we use a different basis for the analysis (we will not go through the differences in the code architectures, but a detailed summary is presented in [20]).

The fact that the isogeometric analysis uses an exact geometry at all levels of the discretization makes a big difference. It affects the accuracy of the analysis as well as the process as a whole, because it means that refinement does not require any external description of the geometry.

The finite element solution, regardless of the approach, of the variational formulation will be a linear combination of the basis functions. While in FEA the coefficients are the nodal variables, in isogeometric analysis they will be the same as control variables. The nodal points and variables in classic FEM are usually interpolated, this is however not the case with the isogeometric method. There are also many similarities between the two methods. Nothing about the mathematical structure differs, they are both isoparametric implementations of the Galerkin method, both utilize bases that obey the partition of unity as well as affine transformations achieved by applying them to the coefficients which define the geometry directly.

When preforming a Galerkin projection the exact solution is projected into a finite subspace. This subspace is built up by the chosen basis functions, in the isogeometric case the subspace will be built by NURBS mapping the surface to the

Isogeometric analysis	Classic finite element analysis
NURBS basis	Polynomial basis
Exact geometry	Approximate geometry
Knots	Mesh
Control points	Nodal points
Control variables	Nodal variables
No interpolation of control points and variables in the basis	Interpolation of the nodal points and variables in the basis
C^{p-m} continuity across element edges	C^0 continuity over element edges
Subdomains	Patches

Table 4.1: This table shows main differences between an isogeometric and classic finite element approach.

physical domain Ω. The space of possible solutions, $\mathcal{S}$, is limited to a subspace $\mathcal{S}_h$ (still a Sobolev space as discussed in chapter 2.1) spanned by a finite number of NURBS basis functions.

$$\mathcal{S} \subset \mathcal{S}_h$$

For the one dimensional case and two dimensional case respectively, the subspaces are defined as

$$\begin{aligned} \mathcal{S}_h &= \text{span}\left\{ R_i(\xi) \in \mathcal{S} \middle| \mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\mathbf{B}_i = \Omega \right\} \\ \mathcal{S}_h &= \text{span}\left\{ R_{i,j}^{p,q}(\xi,\eta) \in \mathcal{S} \middle| \mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\mathbf{B}_{i,j} = \Omega \right\} \end{aligned} \tag{4.7}$$

A much more detailed discussion regarding these isogeometric spaces can be found in [14].

Chapter 5
Thermoelasticity

In this chapter we will present the basic equations for the heat transfer problem and thermoelasticity problem, and each of their variational formulations. We assume that the deformations are limited, i.e that they are so small that they in turn do not generate heat. This means that the thermoelasticity equation can be solved in an uncoupled way. In practical terms this means that we can calculate the temperature field independently from the elasticity problem. The temperature is solved separately, as a thermal problem, and then the solution temperature is used to calculate thermal stress and find the stresses and strains.

In both cases the problem is solved with a traditional Galerkin formulation, the NURBS based elements are used to construct the finite element spaces and then utilized in the approximation.

5.1 Heat transfer problem

In this chapter we look at the heat transfer problem and how we find the variational formulation of the problem and estimate the temperature field inside a body.

Heat transfer is the transport of thermal energy due to a spatial temperature difference. Heat transfer in static materials depends only on the spatial gradient and material properties. When the spatial gradient is present in a body or between systems in thermal contact heat transfer will occur. Common experience gives that heat will flow from a region of high temperature to one with low temperature. This flow is referred to as heat flux. One of the basic principles behind the equations governing heat transfer is balance, for the system to satisfy conservation of energy. This means that the heat inflow and heat generated must be equal to heat outflow plus change in internal energy.

5.1.1 Basic equations of thermal heat transfer

We are considering a solid through which heat is flowing, but no heat is generated, the governing equation for conduction for heat transfer is given in [35] as:

$$\rho c \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + f \text{ in } \Omega \times (0, T_f] \tag{5.1}$$

which is applied and solved on the domain $\Omega \times (0, T_f]$ for $\Omega \subset \mathbb{R}$ with specified boundary conditions.

In equation 5.1 the thermal properties of ρ, c and k are constants. They are independent of both position and temperature. They represent density, specific heat and thermal conductivity. T is the unknown temperature field and f represents the body sources of heat, in our case there will be no heat generation or change of internal energy inside the body.

We can apply a number of boundary conditions on different portions of the boundary. Boundary conditions are used to express, for instance, either a given temperature, $\overline{T}$, an applied heat flux, $\overline{Q}$, or convection with the environment. The different kinds of boundary conditions are defined in equations 5.2, 5.3 and 5.4 below. They represent Dirichlet, Neumann and Robin boundary conditions respectively.

$$T = \overline{T} \text{ on } \Gamma_d \tag{5.2}$$

$$-k \frac{\partial T}{\partial \hat{\mathbf{n}}} = \overline{Q} \text{ on } \Gamma_q \tag{5.3}$$

$$-k \frac{\partial T}{\partial \hat{\mathbf{n}}} = h(T - \overline{T}_e) \text{ on } \Gamma_c \tag{5.4}$$

The over line means that the values are known for every time step. In the cases of heat flux and convection $\hat{\mathbf{n}}$ refers to the unit normal vector. In equation 5.4 h refers to the convection coefficient and T_e.

5.1.2 The variational formulation

For the transient heat transfer problem given in the previous section we apply Rothe's method. The time semi-discretization is done with backward-difference first order (BDF1) method for the first two initial time conditions and backward-difference second order (BDF2) method for the later time steps. Rothe's method was chosen instead of the more common Method of line because it has various advantages over the Method of line, especially with regards to adaptivity. With Rothe's method we first discretization in time direction, and then the time semi-discretized equations are discretized in a spatial direction. Another advantage of the Rothe's method approach

with BDF1 and BDF2 discretization is that the scheme is unconditionally stable. An unconditionally stable scheme means that there is no need to satisfy a restriction on the time stepping.

We utilize the following time semi-discretizations for the time domain $0 = t_0 < t_1 < ... < t_m = T_f$, with time step $\Delta t = t_n - t_{n-1}$

$$\rho c \frac{T^n - T^{n-1}}{\Delta t} = \nabla \cdot (k \nabla T^n) + f^n \text{ for } n = 1, 2, ..., m \tag{5.5}$$

$$\rho c \frac{\frac{2}{3}T^{n+1} - 2T^n + \frac{1}{2}T^{n-1}}{\Delta t} = \nabla \cdot (k \nabla T^{n+1}) + f^n \text{ for } n = 2, ..., m-1 \tag{5.6}$$

Equation 5.5 represent BDF1 method and equation 5.6 BDF2 method. Here superscript n indicates the time step number.

After the semi-discretization in time the spatial discretization is done with the Galerkin formulation. We multiply both sides with a test function v and integrate over the domain. This test function will be (as previously mentioned) a NURBS function that lives in the space $\mathcal{V}_{h,0} \subset \mathcal{V}_0$ where the space $\mathcal{V}_0$ denoted $\mathcal{V}_0 = \{v \subset H^1(\Omega) : v|_{\Gamma_d} = 0\}$

The finite dimensional function space $\mathcal{V}_{h,d} \subset \mathcal{V}_d$ contains functions that satisfy the Dirichlet boundary conditions a specified temprature on the edge, $T = d$ on Γ_d. Note that $\mathcal{V}_d = \{v \subset H^1(\Omega) : v|_{\Gamma_d} = d\}$. The goal is to find the temperature, $T^n \in \mathcal{V}_{h,d}$ such that

$$\int_\Omega \left(\rho c \frac{T^n - T^{n-1}}{\Delta t} \cdot v \right) \, d\Omega = \int_\Omega \left((\nabla \cdot (k \nabla T^n) + f^n) \cdot v \right) \, d\Omega \quad \forall v \in \mathcal{V}_{h,0} \tag{5.7}$$

Utilizing both the boundary condition 5.3 and the divergence theorem together with partial integration

$$\begin{aligned} \int_\Omega \left(\rho c \frac{T^n - T^{n-1}}{\Delta t} \cdot v \right) \, d\Omega + \int_\Omega k \nabla T^n \cdot \nabla v \, d\Omega &= \int_\Omega f^n v \, d\Omega + \int_{\Gamma_q} v \left(k \frac{\partial T^n}{\partial n} \right) d\Gamma_q \\ \int_\Omega \frac{\rho c}{\Delta t} T^n v \, d\Omega + \int_\Omega k \nabla T^n \cdot \nabla v \, d\Omega - \int_\Omega \frac{\rho c}{\Delta t} T^{n-1} v \, d\Omega &= \int_\Omega f^n v \, d\Omega + \int_{\Gamma_q} \overline{Q} v \, d\Gamma_q \end{aligned} \tag{5.8}$$

By letting the approximated solution at time step n, T_h^n be expressed in terms of the basis functions, ϕ, $T_h^n = \sum_{i=1}^n T_{h,i}^n \phi_i^n$. With the matrix coefficient for $i, j = 1, .., n_{dof}$,

the following is defined

$$\begin{aligned} M_{ij}^{n} &= \int_{\Omega} \phi_i^n \phi_j^n \; d\Omega \\ S_{ij}^{n} &= \int_{\Omega} k \left(\nabla \phi_i^n \cdot \nabla \phi_j^n \right) \; d\Omega \\ M_{ij}^{n,n-1} &= \int_{\Omega} \phi_i^n \phi_j^{n-1} \; d\Omega \\ F_i^n &= \int_{\Omega} f^n \phi_i^n \; d\Omega + \int_{\Gamma_q} \overline{Q} \phi_i^n \; d\Gamma_q \end{aligned}$$

which means that we can write equation 5.8 as

$$\left(\frac{\rho c}{\Delta t} M^n + S^n \right) T_h^n = \frac{\rho c}{\Delta t} M^{n,n-1} T_h^{n-1} + F^n \tag{5.9}$$

We do the same for for equation 5.6

$$\int_{\Omega} \left(\rho c \frac{\frac{2}{3} T^{n+1} - 2T^n + \frac{1}{2} T^{n-1}}{\Delta t} \cdot v \right) d\Omega = \int_{\Omega} \left(\left(\nabla \cdot (k \nabla T^{n+1}) + f^n \right) \cdot v \right) d\Omega \quad \forall v \in \mathcal{V}_{h,0} \tag{5.10}$$

and by following the same steps as before we get

$$\begin{aligned} &\int_{\Omega} \frac{2\rho c}{3\Delta t} T^{n+1} v \; d\Omega + \int_{\Omega} k \nabla T^{n+1} \cdot \nabla v \; d\Omega - \int_{\Omega} \frac{2\rho c}{\Delta t} T^n v \; d\Omega + \int_{\Omega} \frac{\rho c}{2\Delta t} T^{n-1} v \; d\Omega \\ &= \\ &\int_{\Omega} f^n v \; d\Omega + \int_{\Gamma_q} \overline{Q} v \; d\Gamma_q \end{aligned} \tag{5.11}$$

$$\begin{aligned} M_{ij}^{n,n+1} &= \int_{\Omega} \phi_i^n \phi_j^{n+1} \; d\Omega \\ M_{ij}^{n} &= \int_{\Omega} \phi_i^n \phi_j^n \; d\Omega \\ S_{ij}^{n,n+1} &= \int_{\Omega} k \left(\nabla \phi_i^n \cdot \nabla \phi_j^{n+1} \right) \; d\Omega \\ M_{ij}^{n,n-1} &= \int_{\Omega} \phi_i^n \phi_j^{n-1} \; d\Omega \\ F_i^n &= \int_{\Omega} f^n \phi_i^n \; d\Omega + \int_{\Gamma_q} \overline{Q} \phi_i^n \; d\Gamma_q \end{aligned}$$

which means that we can write equation 5.11 as

$$\left(\frac{2\rho c}{3\Delta t} M^{n,n+1} + S^{n,n+1} \right) T_h^{n+1} = \frac{\rho c}{2\Delta t} M^{n,n} T_h^n - \frac{\rho c}{2\Delta t} M^{n,n-1} T_h^{n-1} + F^n \tag{5.12}$$

5.2 Thermo elasticity problem

It is a known fact that when a body experiences a change in temperature it may cause the body to deform, and any body that undergoes a deformation may experience strain. A thermoelastic problem essentially consists of two components, one relating to the mechanical deformations and the other to the thermal expansion. If the temperature variations due to mechanical deformations are negligibly small the thermal strain is a linear function of the temperature change. When the deformations are so small that they do cause heat production in the body, the problem can be solved sequentially. This is referred to as an uncoupled thermoelastic problem.

5.2.1 Basic equations of thermoelasticity

The strain due to thermal expansion, called the thermal strain ε^{th}, is equal to

$$\varepsilon_{ij}^{th} = \alpha(T)\theta\delta_{ij} \tag{5.13}$$

Here $\theta = T - T_0$ is the temperature change, T_0 is the initial temperature and T is the applied temperature, α is the thermal expansion coefficient (which may be dependent on temperature) and δ_{ij} is the dirac-delta function.

Note that in this it is assumed that the initial temperature is equal to the reference temperature where the body does not experience deformation. If this is not true, and the reference temperature, T_{ref}, is different from the initial temperature, T_0, the equation can be written as

$$\varepsilon_{ij}^{th} = \alpha(T)(T - T_{ref}) - \alpha(T_0)(T_0 - T_{ref})\delta_{ij} \tag{5.14}$$

The mechanical strain due to the elastic deformations is equal to the difference between total and thermal strain.

$$\boldsymbol{\varepsilon}^{el} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}^{th} \tag{5.15}$$

Total deformations is given as

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\mathbf{u} + \nabla^t\mathbf{u}) \tag{5.16}$$

where $\mathbf{u}$ is the displacement vector field.

When working with an uncoupled problem the thermoelasticity equation can be solved sequentially, i.e the temperature can be evaluated independently from the strain. If the temperature field is known at each time step, it is possible to find the stress state from the induced deformations. Utilizing the linear relation between stress, strain and elasticity modulus we can express the stress in the body as

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon}^{el} \tag{5.17}$$

Known as the generalized Hooke's law, where the matrix D contains Young's modulus E and ν, which both are constant material properties.

Combining this fact with equations for strain, 5.13,5.15, 5.16, we can express the stress in terms of the displacement vector field $\mathbf{u}$ as

$$\boldsymbol{\sigma} = \frac{\mathbf{D}}{2}(\nabla \mathbf{u} + \nabla^t \mathbf{u}) - \mathbf{D}\alpha(T)\theta\delta_{ij} \tag{5.18}$$

If we also imagine no dynamic effects from the loads on the body and neglect body force contributions the equilibrium equations is reduced to

$$\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} = \mathbf{0} \tag{5.19}$$

Inserting the stress for thermoelasticity found in 5.18 we end up with the equation

$$\boldsymbol{\nabla} \cdot \left(\frac{\mathbf{D}}{2}(\nabla \mathbf{u} + \nabla^t \mathbf{u}) - \mathbf{D}\alpha(T)\theta\delta_{ij} \right) = 0 \tag{5.20}$$

Equation 5.20 can be solved with Dirichlet and/or Neumann boundary conditions shown in 5.21 respectively.

$$\begin{aligned} \mathbf{u} &= \overline{\boldsymbol{g}} \text{ on } \Gamma_d \\ \boldsymbol{\sigma} \cdot n &= \overline{\mathbf{h}} \text{ on } \Gamma_n \end{aligned} \tag{5.21}$$

5.2.2 The variational formulation

We find the variational form of the problem stated above by multiplying each side by a test function $\mathbf{v} \in \mathcal{V}_0$ where $\mathcal{V}_0 = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$ and integrate over the domain. Obviously the test function $\mathbf{v}$ we have chosen is a NURBS polynomial.

A Galerkin projection is used, we search for the approximated solution in the finite dimensional subspace, and $\mathbf{v} \in \mathcal{V}_{h,0} \subset \mathcal{V}_0$

$$\int_\Omega \left(\boldsymbol{\nabla} \cdot \frac{\mathbf{D}}{2}(\nabla \mathbf{u} + \nabla^t \mathbf{u}) \right) \mathbf{v} \, d\Omega = \int_\Omega (\boldsymbol{\nabla} \mathbf{D}\alpha(T)\theta\delta_{ij}) \, \mathbf{v} \, d\Omega \quad \forall \mathbf{v} \in \mathcal{V}_{h,0} \tag{5.22}$$

Greens formula together with the boundary conditions are used to obtain the variational formulation. It consists of finding the displacements, $\mathbf{u} \in \mathcal{V}_{h,d} \subset \mathcal{V}_g$. Here $\mathcal{V} = \{u \in H^1(\Omega) : u_{\Gamma_D} = g\}$, which contains the functions which satisfies the Dirichlet boundary conditions on the boundary of Γ_d such that

$$\int_\Omega \frac{\mathbf{D}}{2} \nabla \mathbf{u} \cdot \nabla \mathbf{v} d\Omega + \int_\Omega \mathbf{D}\alpha(T)\theta\delta_{ij} \cdot \nabla \mathbf{v} d\Omega = \int_{\Gamma_n} \overline{\mathbf{h}} \cdot \mathbf{v} d\Gamma_n d\Omega \quad \forall \mathbf{v} \in \mathcal{V}_{h,0} \tag{5.23}$$

If we know the temperature, T^n , at a time, t^n, the formulation for the displacement at this moment in time, $\mathbf{u}^n \in \mathcal{V}_{h,g}$, is given by

$$\int_\Omega \frac{\mathbf{D}}{2} \nabla \mathbf{u}^n \cdot \nabla \mathbf{v}^n d\Omega + \int_\Omega \mathbf{D}\alpha(T^n)(T^n - T_0)\delta_{ij} \cdot \nabla \mathbf{v} d\Omega = \int_{\Gamma_n} \overline{\mathbf{h}} \cdot \mathbf{v}^n d\Gamma_n \quad \forall \mathbf{v}^n \in \mathcal{V}_{h,0} \tag{5.24}$$

5.3 Error estimation

Several ways have been developed to estimate the discretization errors of finite element solutions. We differentiate between a priori error estimators and a posteriori error estimators. The first traditionally provide information on the asymptotic behavior of discretization errors, while the second employs the finite element solution to calculate error estimations of the analytical solution errors.

When we do the error estimation on our problem we wish to evaluate the error over each element separately in some norm. Since the finite element method minimizes the distance between the actual solution, u, and the approximated one, u_h, the energy norm is obviously a good choice of norm to use in the error analysis. The energy norm is associated with the discretization error from the finite element solution. We can work with the exact error if we have an analytical solution for our test problem. The energy norm depends on the bilinear operator of the problem, $a(\cdot, \cdot)$.

The energy norm that evaluates the exact error is defined as

$$\|u - u_h\|_E = \sqrt{a(u - u_h, u - u_h)} \tag{5.25}$$

In cases where the bilinear operator is self adjoint, the finite element solution, u_h, will be the optimal approximation to the analytical solution, u, as measured in the energy norm (sometimes called a-norm).

It is shown in [4] that the a priori convergence rate of isogeometric analysis based on NURBS is bounded in the same as classical FEA. Given that we are working on a mesh, $\mathcal{M}$, that is quasi-uniform and uniform if the analytical solution $u \in H^{p+1}$, meaning that it is sufficiently smooth, and u is the solution of a variational problem containing first order differentiation the error of the approximate finite element solution is bounded as described in equation 5.26. C is some constant dependent on the problem and h is the element size with p the highest polynomial order in the basis. $\|u\|_{H^{p+1}}$ is the analytical solution in the Sobolev norm with order $p + 1$. Knowing this we can compare the convergence rates the numerical results with the upper bound.

$$\|u - u_h\|_E \leq Ch^p \|u\|_{H^{p+1}} \tag{5.26}$$

In cases where we deal with singularities, the solution is not sufficiently smooth and the error is not bounded in this way. In these instances the error bound depends on a real non-negative parameter α which is connected with how the problem is meshed. If we define λ as the strength of the singularity (limited to real numbers), we have $\alpha = \min\{p, \lambda\}$. The errors convergence is then bounded by

$$\|u - u_h\|_E \leq Ch^\alpha \|u\|_{H^{\alpha+1}} \tag{5.27}$$

Note that when $\lambda < p$ the convergence rate is limited not by the polynomial order, but the strength of the singularity.

These error bounds are proven and further discussed in [14] and [13]

Working with the uncoupled thermoelasticity simplifies the error analysis somewhat. It means that we can work with the two parts belonging to stress and heat of the thermoelastic equation 5.24 independently from each other. We assume that the temperature related error does not affect the stress and focus only on the thermal stress, and separate the error analysis for the elasticity solver. This means that the bilinear operator we use in the energy error is equal to

$$a(u, v) = \int_{\Omega} \frac{\mathbf{D}}{2} \nabla \mathbf{u} \cdot \nabla \mathbf{v} d\Omega \tag{5.28}$$

Keeping in mind the equations from chapter 5.1.1, this gives us an energy norm equal to

$$\|u - u_h\|_E = \sqrt{\int_{\Omega} (\sigma - \sigma_h)^T (\varepsilon - \varepsilon_h) d\Omega} \tag{5.29}$$

which is the energy norm we base our later convergence plots on.

Chapter 6
Numerical examples

In this chapter we present the numerical models that will be tested with IFEM. In order to analyze the robustness of the solver we chose to test it on one model problem where we are in possession of the analytical solution as well as one where we are presented with a singularity. These model problems are presented in chapter 6.3 and 6.4.

As we wish to test not only the correctness of the solver, but also the robustness of the method we will build a numerical example of this geometry with different sets of basis functions. This is done in order to see if (and how) the results are affected by the function space.

In this chapter we will present the numerical results from the IFEM solver's performance on the different model problems. The convergence rates are plotted as a function of the number of degrees of freedom for different function spaces. This is done in order to assess the solver's robustness. We will first and foremost focus on comparing the convergence rates for the different function spaces, and investigate how they affect the error (measured in the energy norm), and if the error behaves as expected.

6.1 IFEM solver

We will use the IFEM solver for our numerical testing done in this chapter. It is a fairly new finite element solver so it is not as polished as other solvers on market, but it has the advantage of being a isogeometric solver. The thermoelastic solver in IFEM has been written from scratch at SINTEF. SINTEF wished to create a solver based on an isogeometric approach to the finite element method. The solver has been developed with two goals in mind, firstly they wanted to address the basic design needs and secondly they wanted to use the most effective possible solution algorithms without any loss in accuracy. We will be testing different input files and function spaces to see if it has any effect on our results. We also wish to see how

Notation	Parameter	Value	SI units
T	Temprature	$T_i = 373$, $T_o = 293$	K
ρ	Mass density	7850	$\frac{\text{Kg}}{\text{m}^3}$
ν	Poisson ratio	0.3	no unit
α	Thermal expansion	1.2×10^{-5}	$\frac{1}{\text{K}}$
E	Young's Modulus	2×10^{11}	Pa

Table 6.1: This table shows the material parameters used in the numerical models.

IFEM handles problems with singularities, and if double or multiple knot lines gives a better result.

The thermal stress is analyzed with a sequential method, first solving for the temperature and then finding the stress with an elasticity solver. We apply results from the first analysis in order to proceed with the second analysis. In our case this entails using the temperature results from the first analysis in our second analysis in order to find the thermal stresses.

The IFEM solver utilizes Gauss integrals when finding the solution of the integrals of the variational formulation and in our simulations the integrals is solved with four Gauss points.

6.2 Model problems

The model problem used for our numerical analysis will be a steel pipe. We will analyze a cross-section of the pipe using a plane strain model. Because the problem is symmetric, we only need to look at a quarter segment of the pipe. We will focus on the plane strain problem so the pipe will be with fixed in both ends, not allowing any movement in z-direction.

The two geometries we will be looking at can be viewed as a circular pipe, called annulus, and a square pipe, called L-shape. We have an analytical solution that applies to the annulus only.

In order to avoid misunderstandings the table 6.2 shows the units and notations used. The notation $\mathcal{S}^{p,p-m}$ refers to the function spaces spanned by the basis functions. As previously the superscript p refers to the order of these basis functions. We will refer to $p - m$ as the regularity of the basis with m being the multiplicity of the knots. The regularity gives the continuity of the basis across knot lines. When we refine the mesh (by inserting knots) it is done uniformly by recursively halving the interval between consecutive knots. Inserting knot lines with a higher multiplicity

might be a good idea if we wish a lower continuity. A lower continuity might be a beneficial if we are working on a geometry modeling or if we know something about the underlying problem. In order to investigate how the multiplicity of knot lines and the degrees of freedom affect the numerical results, we run the geometry of the steel pipe with different sets of bases $\mathcal{S}^{p,p-m}$ with $m = 1 \dots p-1$ and regularities. We will also compare the error for different polynomial orders. Each of the function spaces will be tested with different degrees of freedom, achieved by refining the knot vectors with knot insertion.

The boundary conditions will be identical for each of the geometries. The temperature on the inner and outer surfaces of the pipe will be imposed as Dirichlet conditions and remain constant during the simulations. Since the deformations causing stress in the body is only due to the temperature changes, there are no additional forces causing stresses or strains in the body (apart from gravity of course). The model is one quarter of a pipe so the edges will be restricted, only allowing the cylinder to expand outward in the radial direction. These restrictions are imposed as Dirichlet boundary conditions. In other words both problems will be a plane strain problem.

6.3 The Annulus: a circular pipe

We wish to do a numerical analysis of a circular pipe model with inner radius $r_i = 0.03$ mm and outer radius $r_o = 0.04$ mm. The choice of this geometry was partially motivated by the fact that we are in possession of an analytical solution for this problem, but also because this problem will showcase the advantages of the exact geometry due to NURBS basis functions.

For the circular pipe problem we are considering, we assume that it is symmetrical around the axis and independent of the axial coordinate z. Supposing that the cylinder's edges are fixed the axial displacement, w, is zero throughout. The cylinder will only have three stress components σ_r, σ_θ and σ_z , as the shear strains and stresses will be equal to zero due to symmetry and uniformity in the axial direction.

The relation between the thermal stresses and strains for the plane strain problem is given below

$$\begin{aligned} \varepsilon_r - \alpha T &= \frac{1}{E}\left[\sigma_r - \nu\left(\sigma_\theta + \sigma_z\right)\right] \\ \varepsilon_\theta - \alpha T &= \frac{1}{E}\left[\sigma_\theta - \nu\left(\sigma_r + \sigma_z\right)\right] \\ \varepsilon_z - \alpha T &= \frac{1}{E}\left[\sigma_z - \nu\left(\sigma_r + \sigma_\theta\right)\right] \end{aligned} \tag{6.1}$$

The analytical solution presented in the next chapter only applies if the heat transfer has achieved a steady state, therefore we make sure that the numerical simulations run long enough to reach steady state.

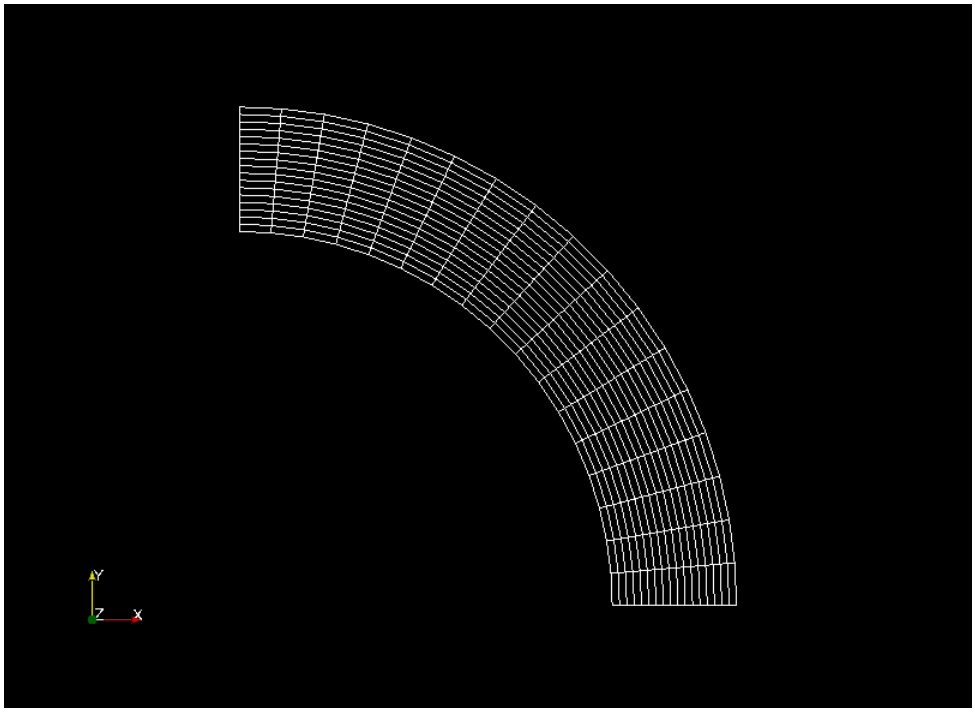

Figure 6.1: Annulus problem: This is an example of one of the meshes we are working with. The figure shows the mesh for knot vectors with 16 individual unique knots, i.e 16 elements in each direction.

6.3.1 Analytical solution for the annulus

For a cylinder with a circular hole experiencing steady state heat flow the plane strain problem for the pipe has an analytical solution. T_i denotes the temperature on the inner radius which is referred to as r_a. The analytical solution for the temperature at any distance r from the center of the cylinder and the thermal stresses caused is given in [18].

$$T(r) = T_i + \frac{T_o - T_i}{\ln\left(\frac{r_b}{r_a}\right)} \ln\left(\frac{r}{r_a}\right) \tag{6.2}$$

Number of elements	Number of degrees of freedom	Relative error (%)
4	40	3.1595
9	60	1.41529
25	112	0.5498
81	264	0.2723
289	760	0.2282
1089	2520	0.2244

Table 6.2: Annulus problem: This table shows the number of elements, number of degrees of freedom and the relative error for the annulus model with cubic NURBS with regularity 2.

Defining the

$$C = \frac{-E\alpha(T_i - T_o)}{2(1-\nu)} \tag{6.3}$$

$$\begin{aligned}
\sigma_r(r) &= C\left[\frac{\ln\left(\frac{r_b}{r}\right)}{\ln\left(\frac{r_b}{r_a}\right)} - \frac{\left(\frac{r_b^2}{r^2} - 1\right)}{\left(\frac{r_b^2}{r_a^2} - 1\right)}\right] \\
\sigma_\theta(r) &= C\left[\frac{\ln\left(\frac{r_b}{r}\right) - 1}{\ln\left(\frac{r_b}{r_a}\right)} + \frac{\left(\frac{r_b^2}{r^2} + 1\right)}{\left(\frac{r_b^2}{r_a^2} - 1\right)}\right] \\
\sigma_z(r) &= \nu\left(\sigma_r + \sigma_\theta\right) - \alpha T
\end{aligned} \tag{6.4}$$

In our error analysis, we want to focus on the error from the elasticity solver. Instead of using the computed temperature field in our elasticity solver, we choose to input the analytical temperature field (calculated from the analytical solution presented in chapter 6.3.1) This is a way to make sure that we have the exact temperature field in our later calculations. The error will then only depend on the elasticity solver, thus ensuring that the error norm we describe in 5.29 holds.

6.3.2 Results for the annulus

In this chapter we will present the results from the testing done on our circular pipe, the annulus problem. Figure 6.2 shows the temperature profile after it becomes steady state and figures 6.3 and 6.4 shows stresses caused by this (steady state) temperature field (in cylindric coordinates) as it was calculated by IFEM.

For the annulus problem with cubic NURBS the solver gives good results, the relative error given in table 6.3.2.

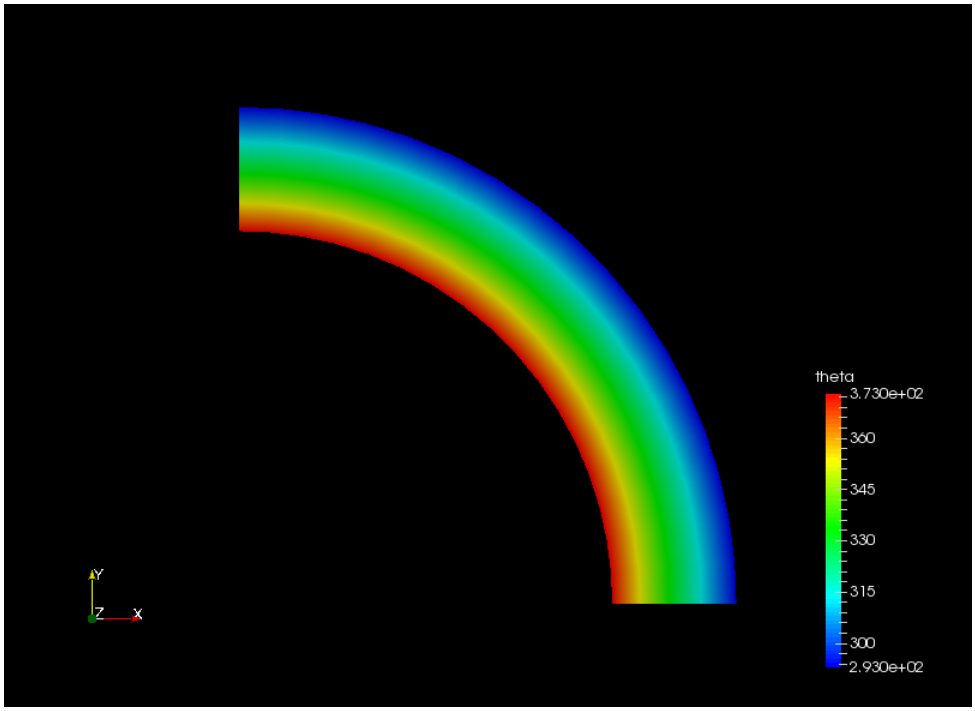

Figure 6.2: Annulus problem: This figure shows the IFEM temperature distribution of the annulus after it has reached steady state.

During our testing we made different bases with different continuities, from maximum C^{p-1} to minimum C^1. Using the notation we introduced in the previous chapter, the function spaces will be $\mathcal{S}^{3,1}$ and $\mathcal{S}^{3,2}$ for polynomial order three. $\mathcal{S}^{4,1}$ and $\mathcal{S}^{4,2}$ and $\mathcal{S}^{4,3}$ for polynomial degree four. For polynomial degree 5 we look at the function spaces $\mathcal{S}^{5,2}$, $\mathcal{S}^{5,3}$ and $\mathcal{S}^{5,4}$. The different convergence rates for these different function spaces are presented in the figures 6.6, 6.7 and 6.8

In figure 6.5 we see that the convergence rates for the different polynomial degrees behave as we might expect from classic FEA, namely that a higher polynomial order gives a better convergence rate. It can be seen that a higher polynomial also gives a lower error. The convergence rates are dependent on the polynomial order as well as the size of the elements, h. The error is bounded as described in chapter 5.25.

When looking at the results presented in figure 6.6 we clearly see that the choice of function space affects the error. A high regularity gives a lower error versus degrees of freedom for the same polynomial order. The fact that the same number of elements with a lower regularity gives better accuracy, makes sense since the function space

$\mathcal{S}^{p,p-1}$ is contained in $\mathcal{S}^{p,0}$ and

$$\mathcal{S}^{p,p-1} \subset \mathcal{S}^{p,p-m} \subset \mathcal{S}^{p,0} \quad \text{where } m = 2 \dots p-2$$

We get the same results for the other polynomial orders and their error plots can be seen in 6.7 and 6.8. A low multiplicity is more accurate when the relative error is evaluated with respect to the number of degrees of freedom. In the cases of $p = 3$ and $p = 4$ the polynomial order has a bigger effect on the accuracy than the regularity, as can be seen in figure 6.13. However, for $p = 5$ it can be seen that if we increase the multiplicity of the knot lines we negate the effect of the higher polynomial with regards to accuracy of solution.

It should also be noted that multiple knot line insertion gives a fast growth of degrees of freedom, especially when refining the mesh, so increasing the multiplicity should not be done without thought.

6.4 L-shape: the square pipe

An important difference between the circular and the square pipe is that the square pipe, from now on called L-shape, has a singularity at the inner corner. This means that the solution function has a sharp edge here and the derivative of the function is not defined.

In chapter 5.25 we talked about how the error bound changes for a problem with a singularity. Such a problem depends on the parameter $\alpha = \min\{p, \lambda\}$, and if $\lambda < p$ it will depend on the strength of the singularity rather than the polynomial degree of the basis functions. In [32] the strength of a the singularity from an L-shape is given as $\lambda = 0.5444$. This means that for L-shape problem $\alpha = \lambda = 0.5444$ should determine the convergence rate regardless for the polynomial order of the basis functions.

Unfortunately we do not posses an analytical solution for this problem. In our error analysis we will therefore use the solution of a very fine mesh and with small time steps as a reference solution and compare it to the global energy error from our different test cases. Denoting the fine meshed approximated solution as u and the approximated solution as u_h, the relative error is calculated as follows

$$\text{Relative error } = \frac{\sqrt{a(u,u) - a(u_h,u_h)}}{\|u\|}$$

The L-shape is not a particularly suited for splines, as it is perfectly possible to describe it with other and more suited, basis functions. It is more common to use LR-B-splines or T-splines instead of NURBS as basis functions, since the geometry benefits from adaptive or locally refined meshes. However, we wanted to see how the

IFEM solver handles the singular problem of the L-shape compared to the smooth problem tested with the annulus, as well as how the approximated solution is affected by the different function spaces. We tested the performance of NURBS on a uniform and on a graded mesh. A graded mesh is meshed finer as we approach the problematic area of the model, in our case in the L-shape's inner corner.

Figure 6.10 shows an example of the meshes we are working with. This figure shows the mesh with 17 elements in each direction for a cubic polynomial with regularity 2. The reader might note that the two models do not have the exact same geometric shape (one is slightly longer than the other) but the thickness of the walls is the same, as well as the boundary conditions which is what is important for our numerical test.

The geometry was constructed with a knot in the edge at the L-shape's inner corner with a high enough multiplicity to ensure that the problem is C^0 continuous across this element knot line regardless of the refinement or regularity.

6.4.1 Results for the L-shape

The tests for the L-shape gave results that were less the satisfactory both the uniform and graded meshes. The convergence rate evens out too quickly, and is not as sensitive to the number of degrees of freedoms as we would expect. Smaller elements gives a minimal improvement. Even though we expect a reduced convergence rate for a uniform refinement for a geometry with a singularity as the L-shape, it should be better than the results we ended up with. It would seem that the IFEM solver is not yet equipped to solve problems where the solution is $u \notin H^{p+1}$. We see that we do not achieve anything close to an optimal convergence, nor a convergence with a upper bound such as described in 5.27. We also observe that the use of higher polynomials are has a minimal improvement on the accuracy of the solver.

In an attempt to obtain where the IFEM solver faces difficulties we did an independent study of the thermal problem, to see if the error lies in the thermal solver or elasticity solver. The results from this analysis suggest that the problem might lie in the thermal solver because the convergence rates were clearly wrong, as can be seen from figure 6.18.

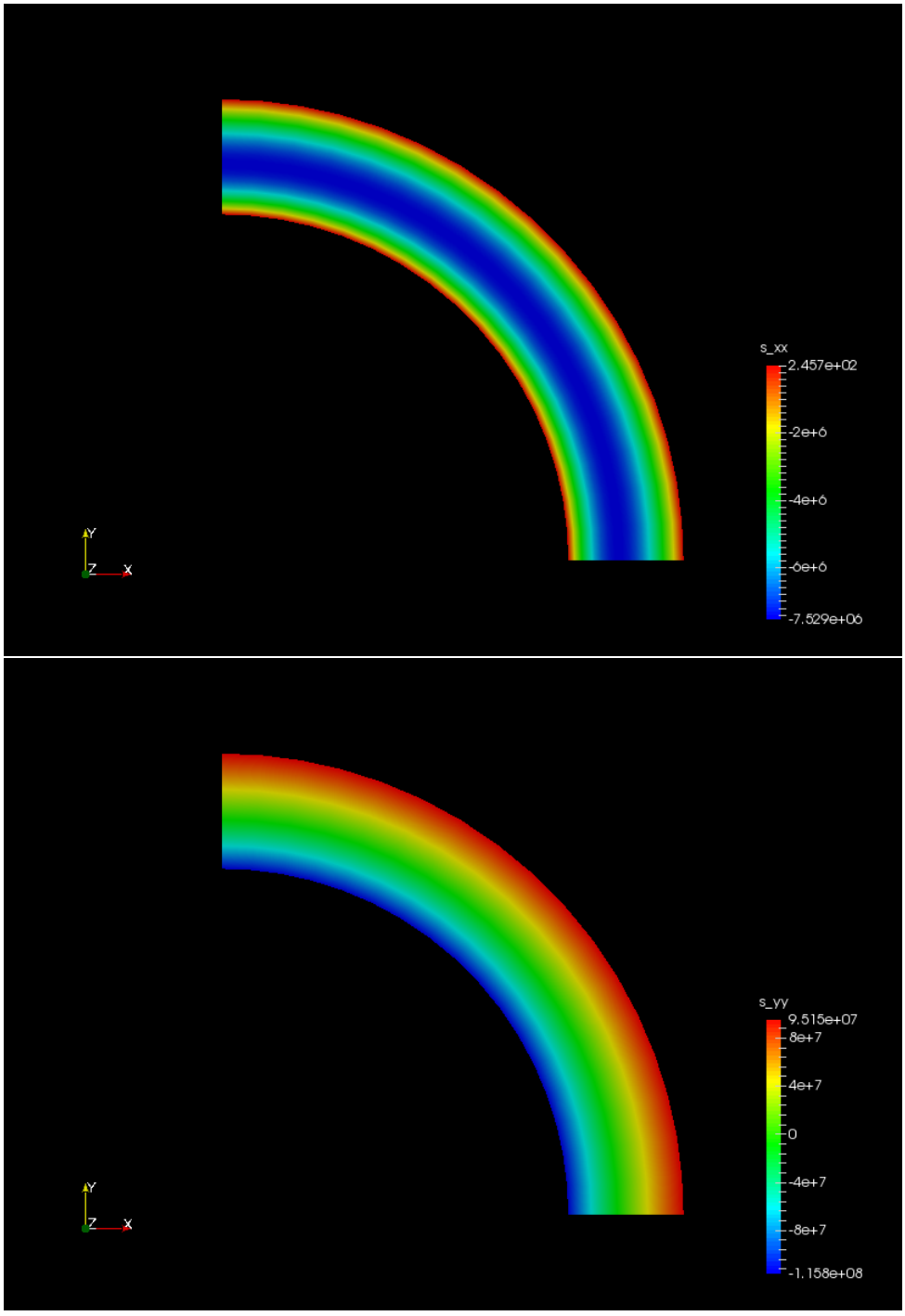

Figure 6.3: Annulus problem: This figure shows the IFEM plots of the radial stress and the hoop stress respectively.

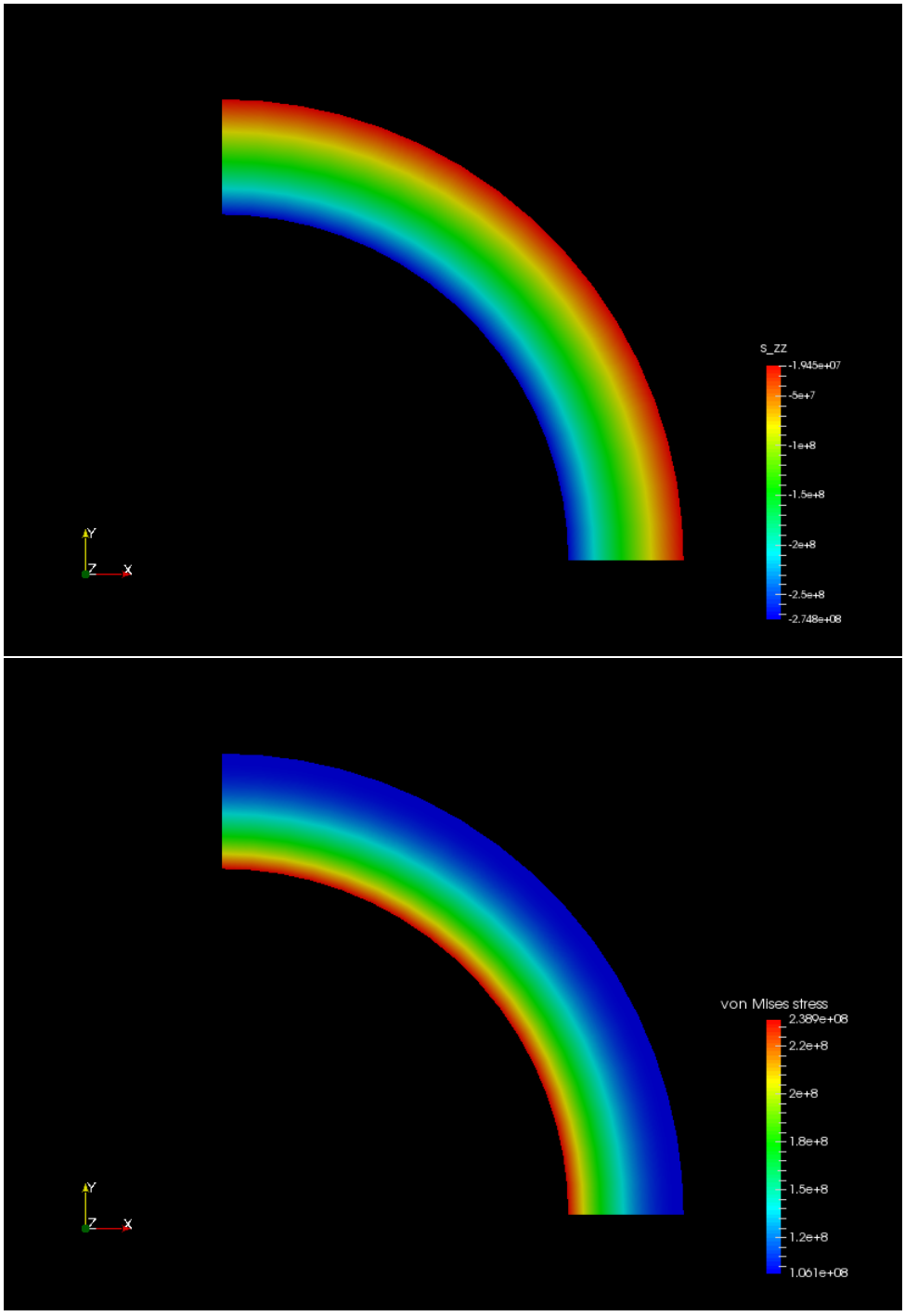

Figure 6.4: Annulus problem: This figure shows the IFEM plots of the axial stress and the von Mises stress respectively.

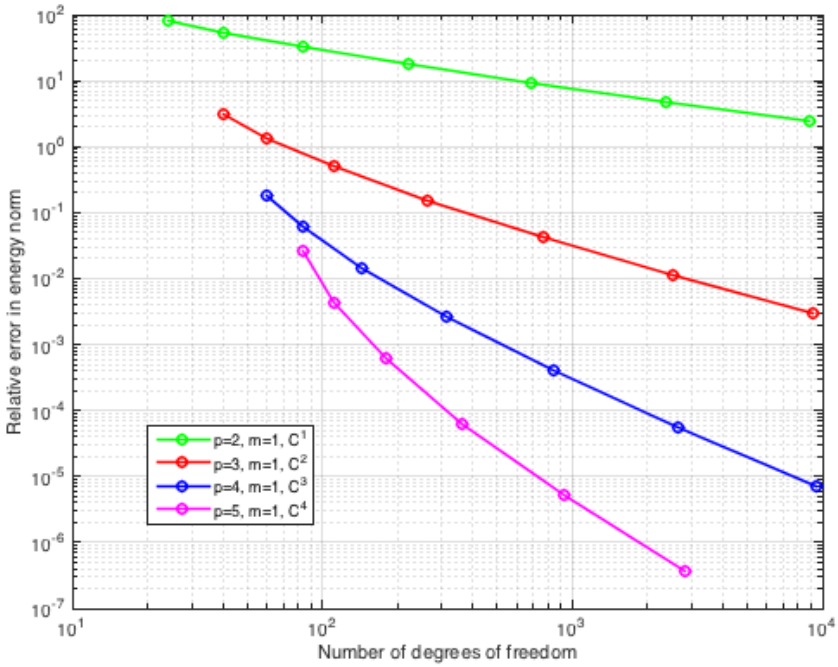

Figure 6.5: Annulus problem: This figure plots the relative error (%) of the different polynomial degrees.

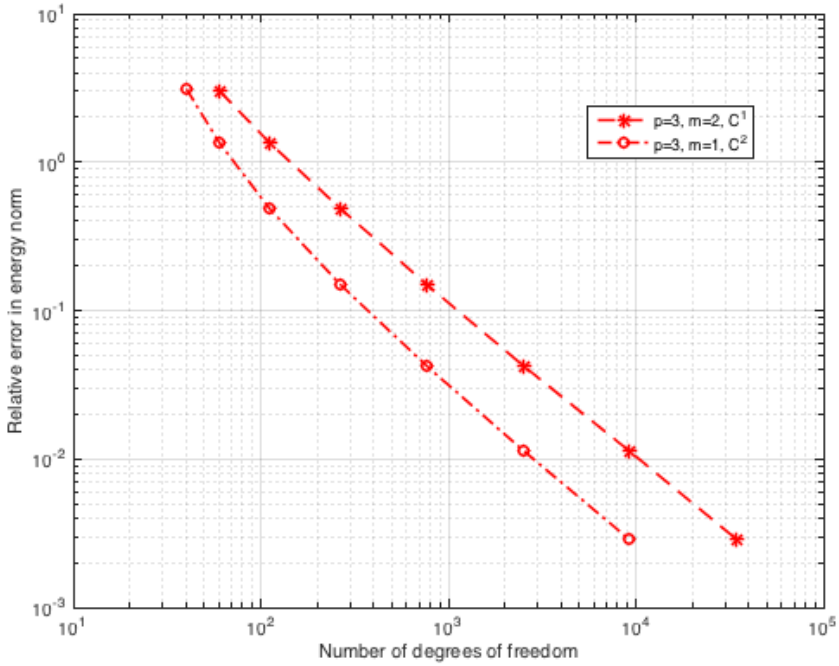

Figure 6.6: Annulus problem: This figure shows the relative error(%) vs degrees of freedom for different function spaces for polynomial order 3.

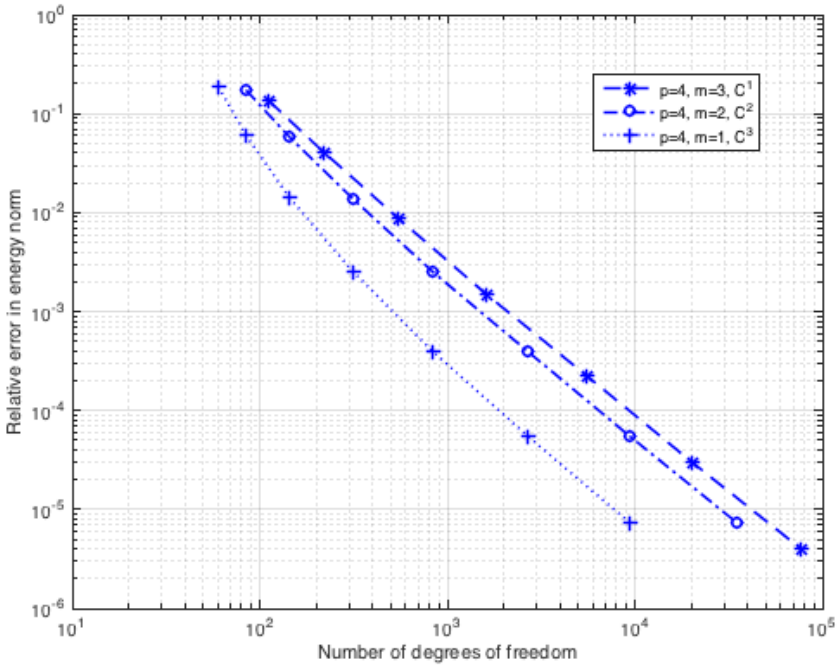

Figure 6.7: Annulus problem: This figure shows the relative error(%) vs degrees of freedom for different function spaces for polynomial order 4.

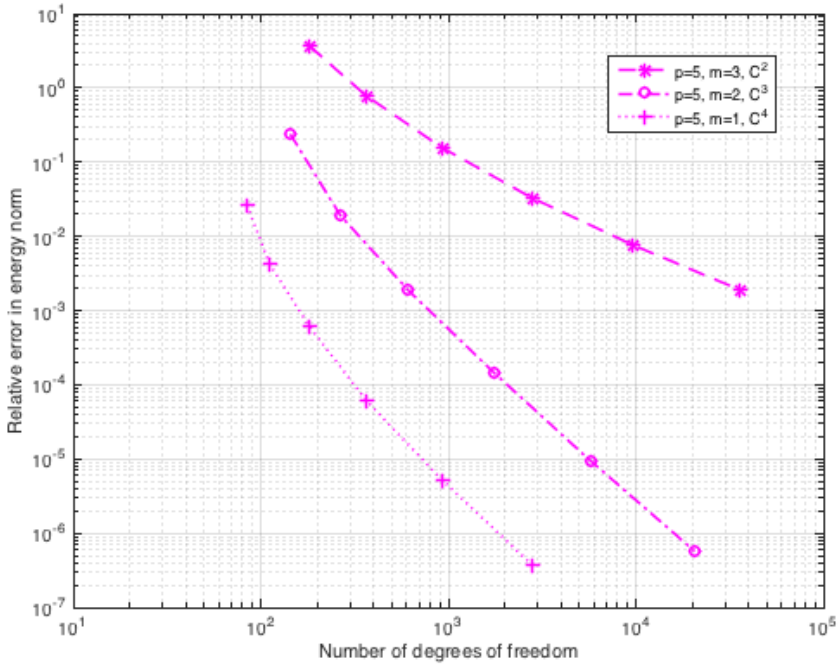

Figure 6.8: Annulus problem: This figure shows the relative error(%) vs degrees of freedom for different function spaces for polynomial order 5.

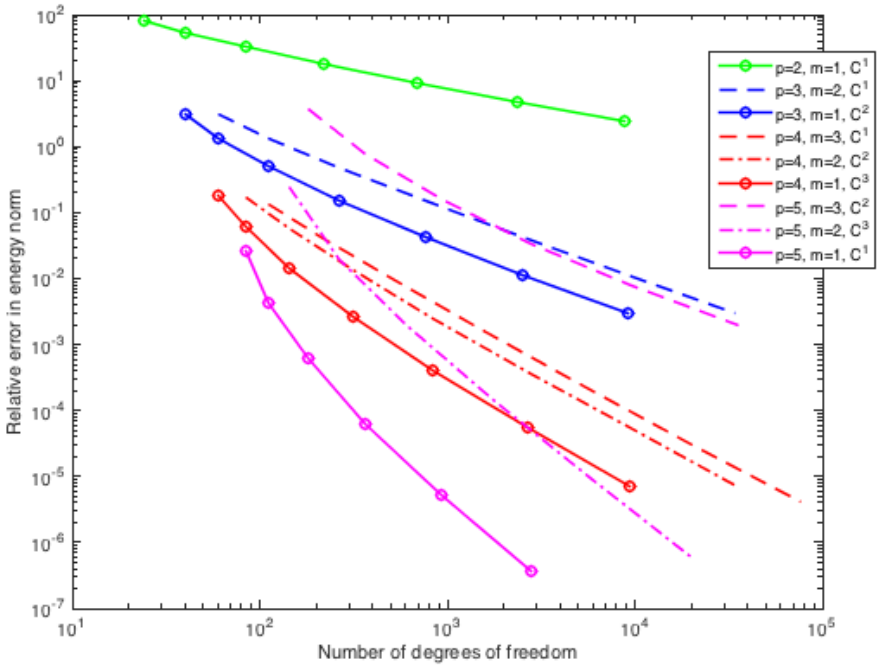

Figure 6.9: Annulus problem: This figure shows the relative error(%) vs degrees of freedom for different function spaces for polynomial orders 3, 4 and 5.

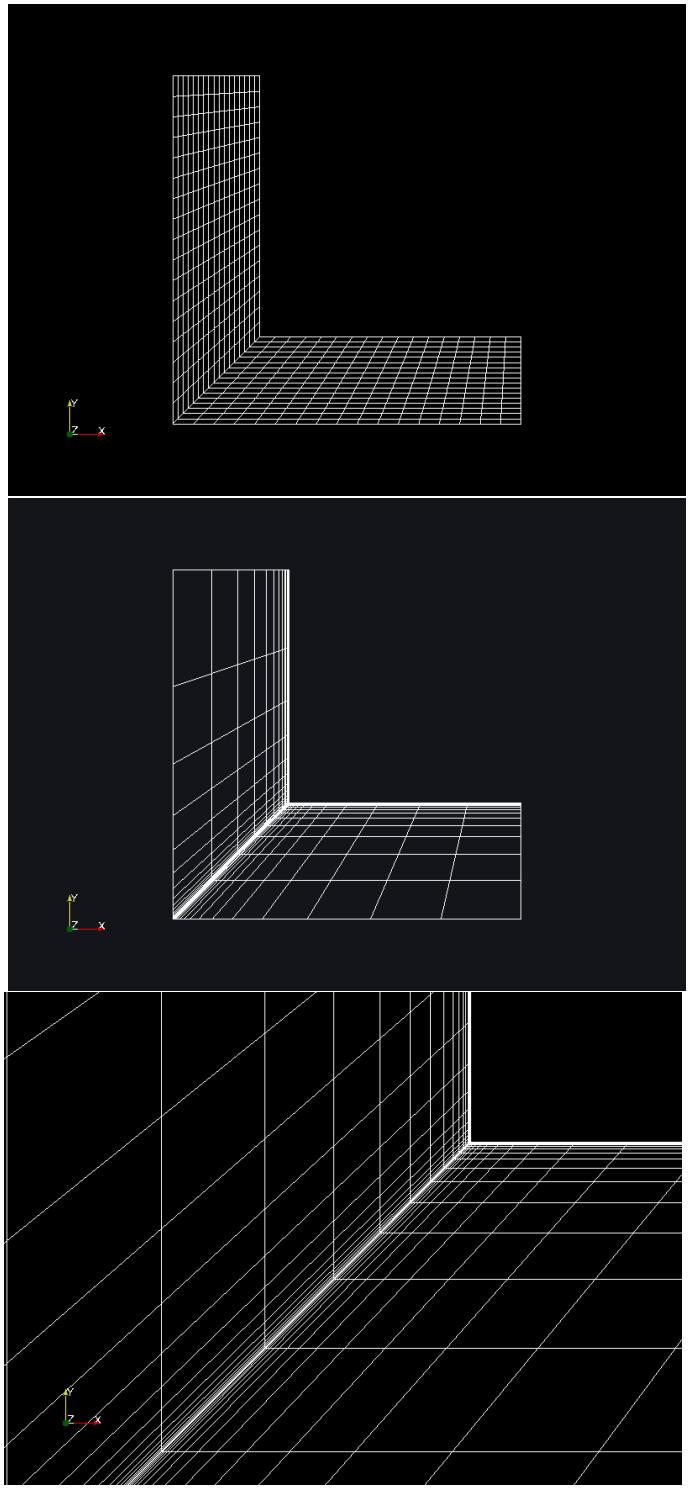

Figure 6.10: L-shape problem: This figure shows the different meshes we use when testing our L-shape. The last figure shows a zoom-in of the graded mesh.

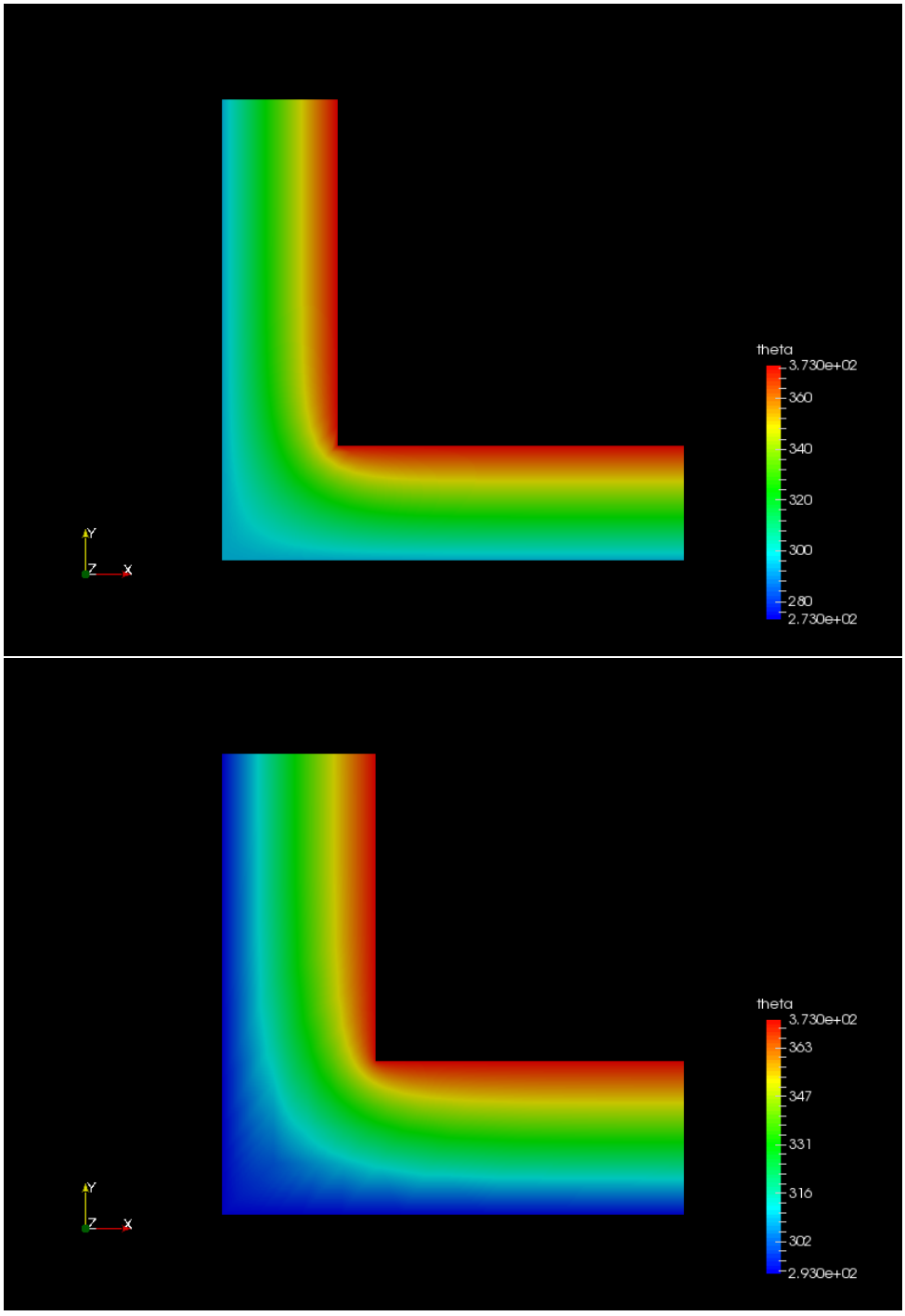

Figure 6.11: L-shape problem: This figure shows the IFEM temperature distribution of the L-shape after it has reached steady state.

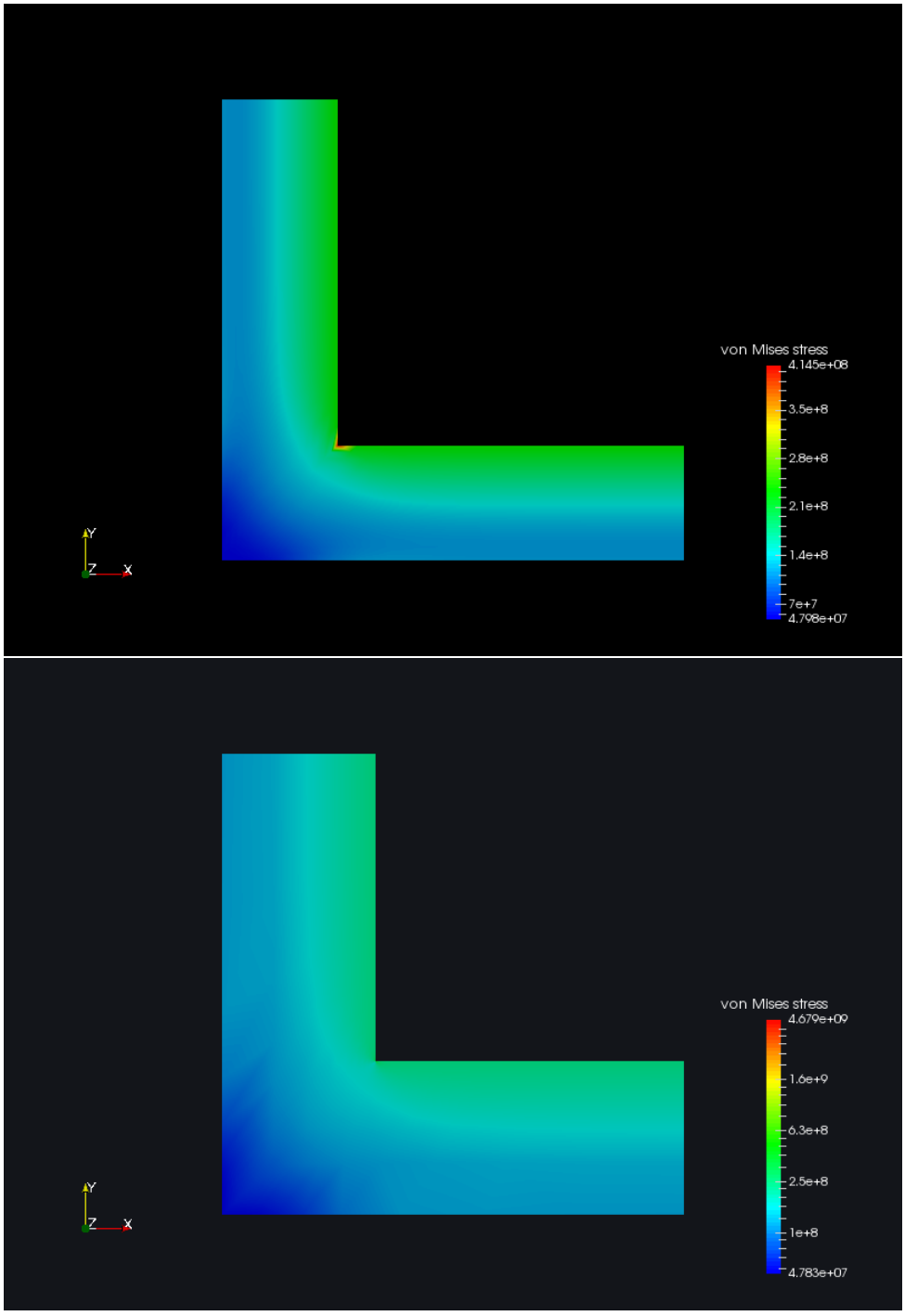

Figure 6.12: L-shape problem: This figure shows the IFEM plots of the von Mises stress.

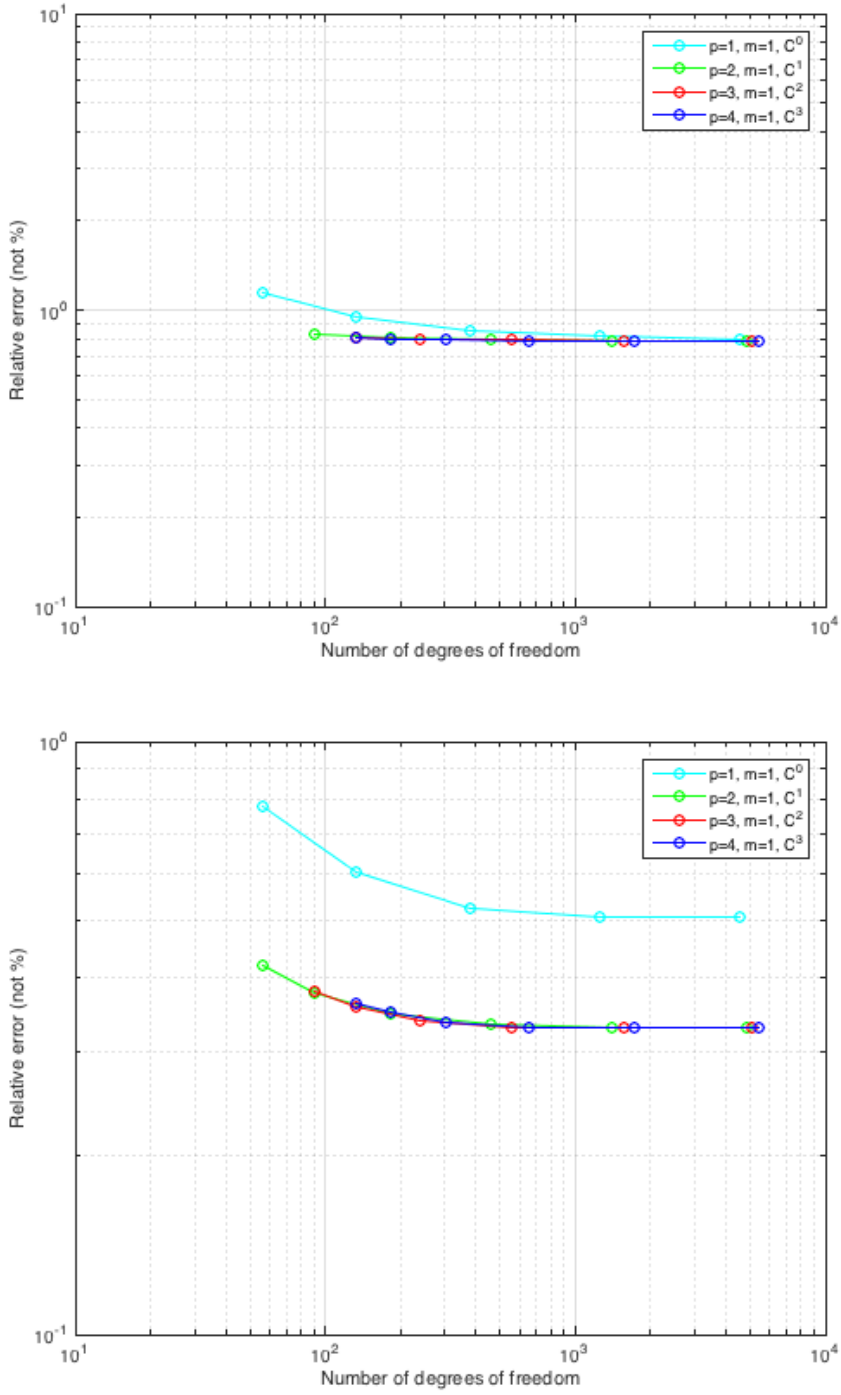

Figure 6.13: L-shape problem: This figure plots the relative error of the different polynomial degrees.

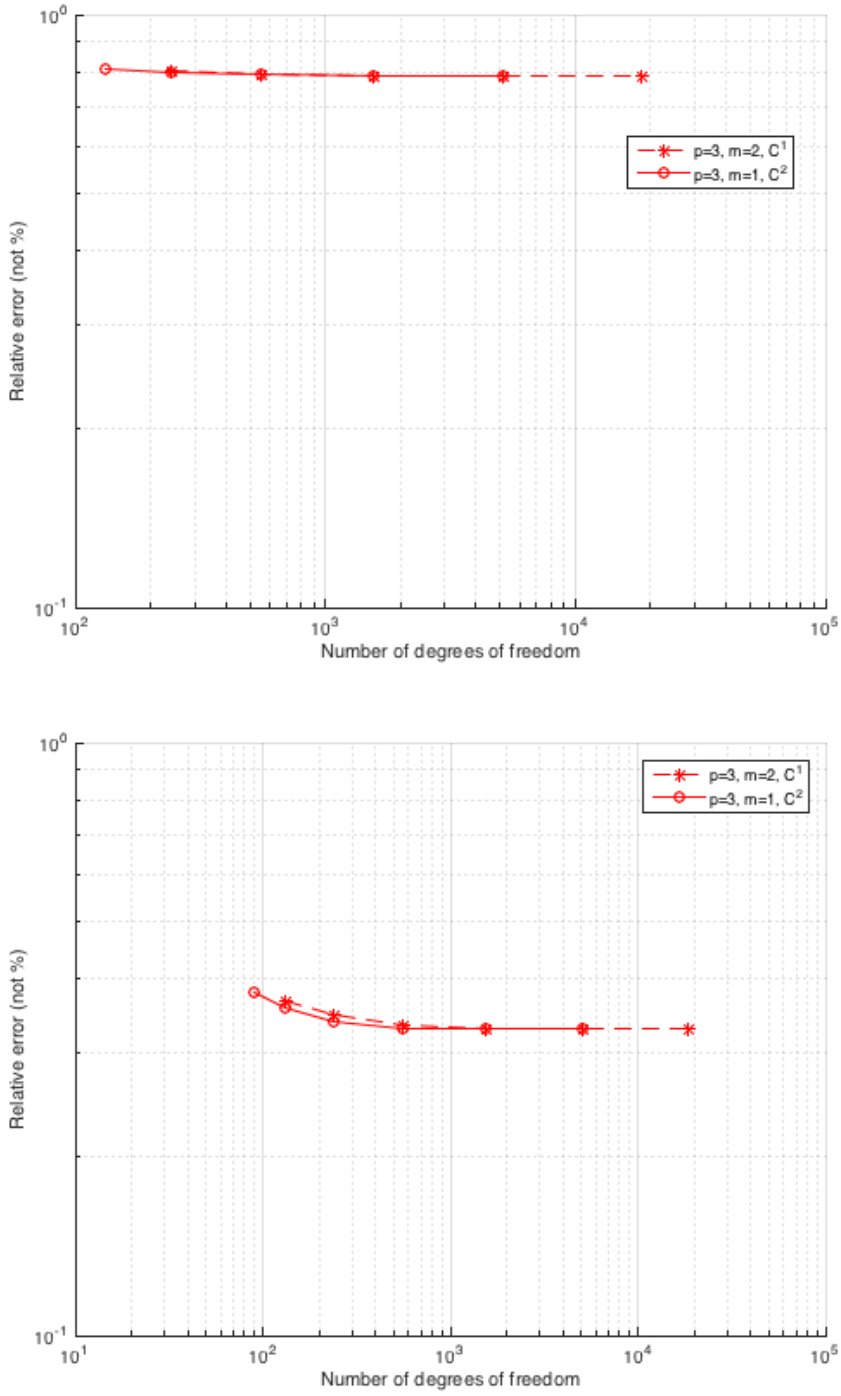

Figure 6.14: L-shape problem: This figure plots the relative error of the polynomial order 3 for the uniform and graded mesh respectively.

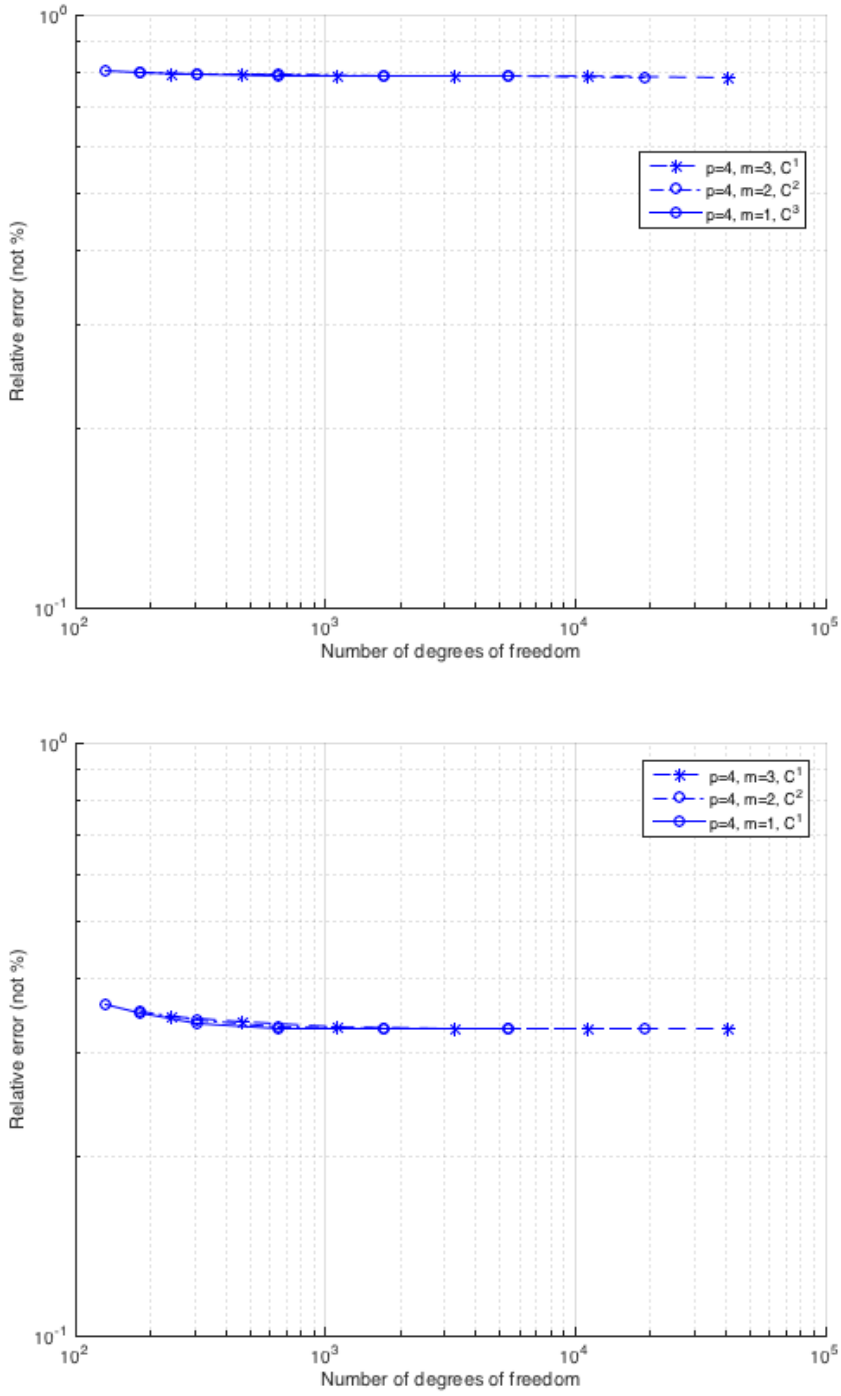

Figure 6.15: L-shape problem: This figure plots the relative error of the polynomial order 4 for the uniform and graded mesh respectively.

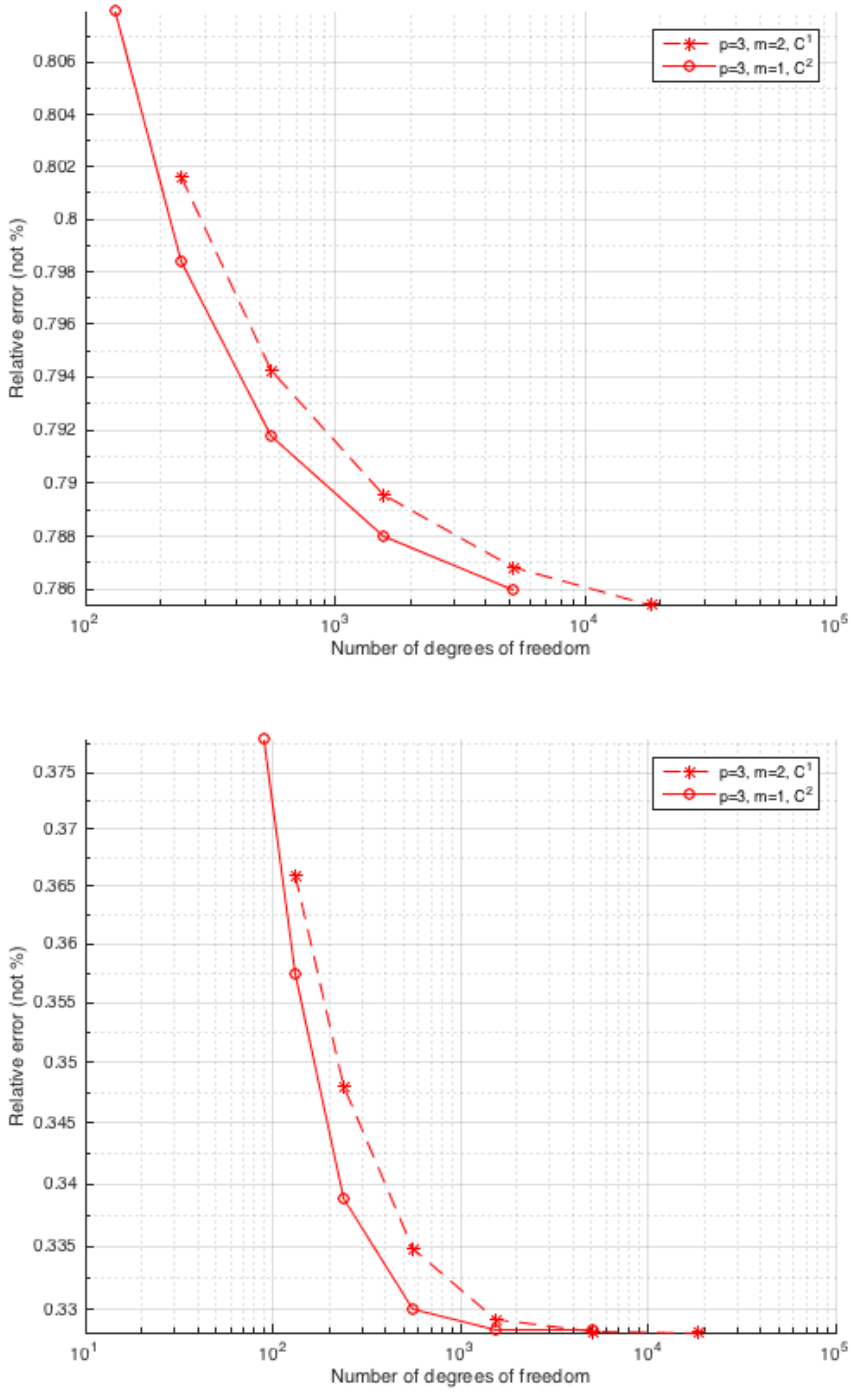

Figure 6.16: L-shape problem: This figure plots the relative error of the polynomial order 3 for the uniform and graded mesh respectively. This figure shows the same plots as figure 6.14, but is rescaled to highlight the differences

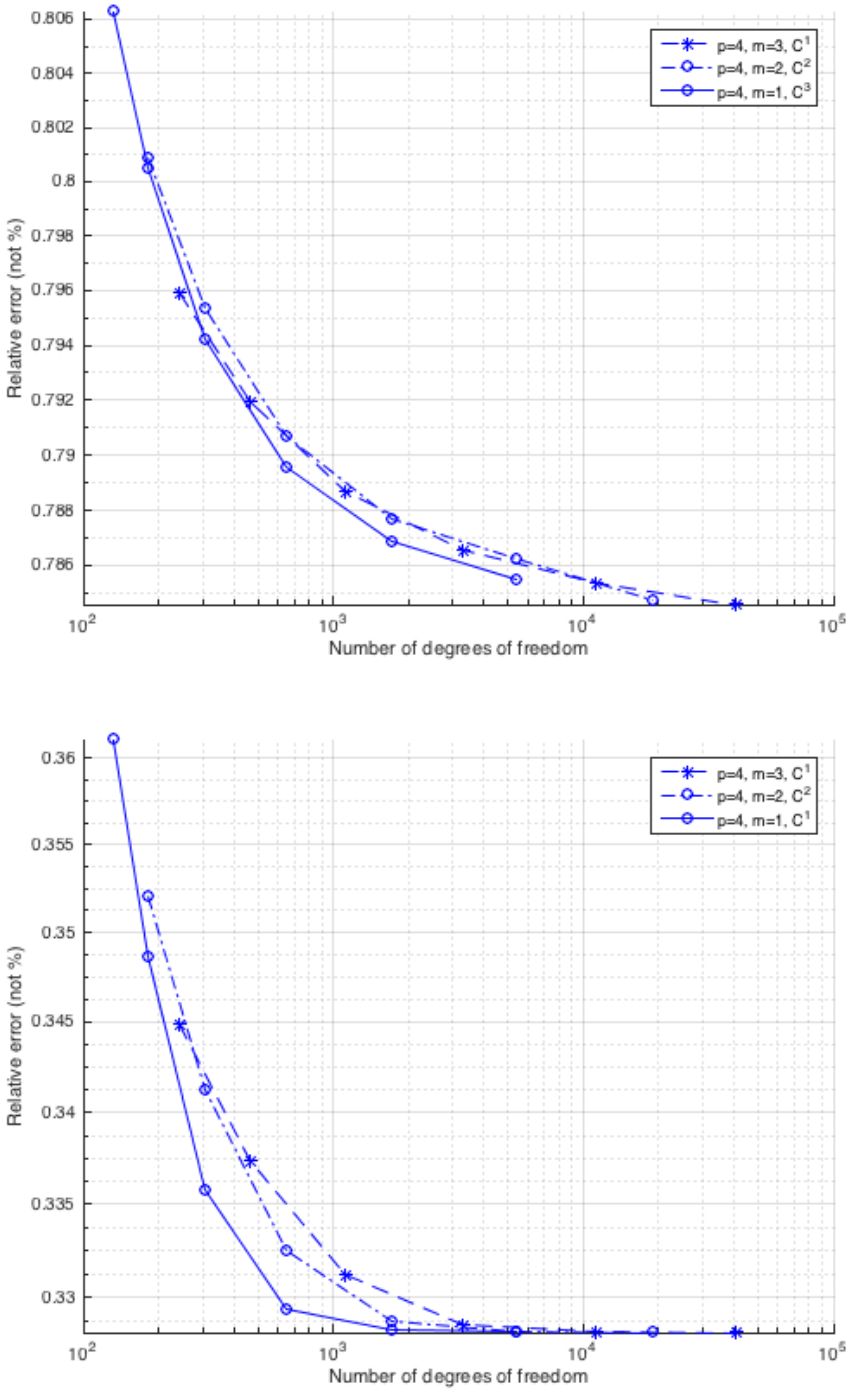

Figure 6.17: L-shape problem: This figure plots the relative error of the polynomial order 4 for the uniform and graded mesh respectively. This figure shows the same plots as figure 6.15, but is rescaled to highlight the differences.

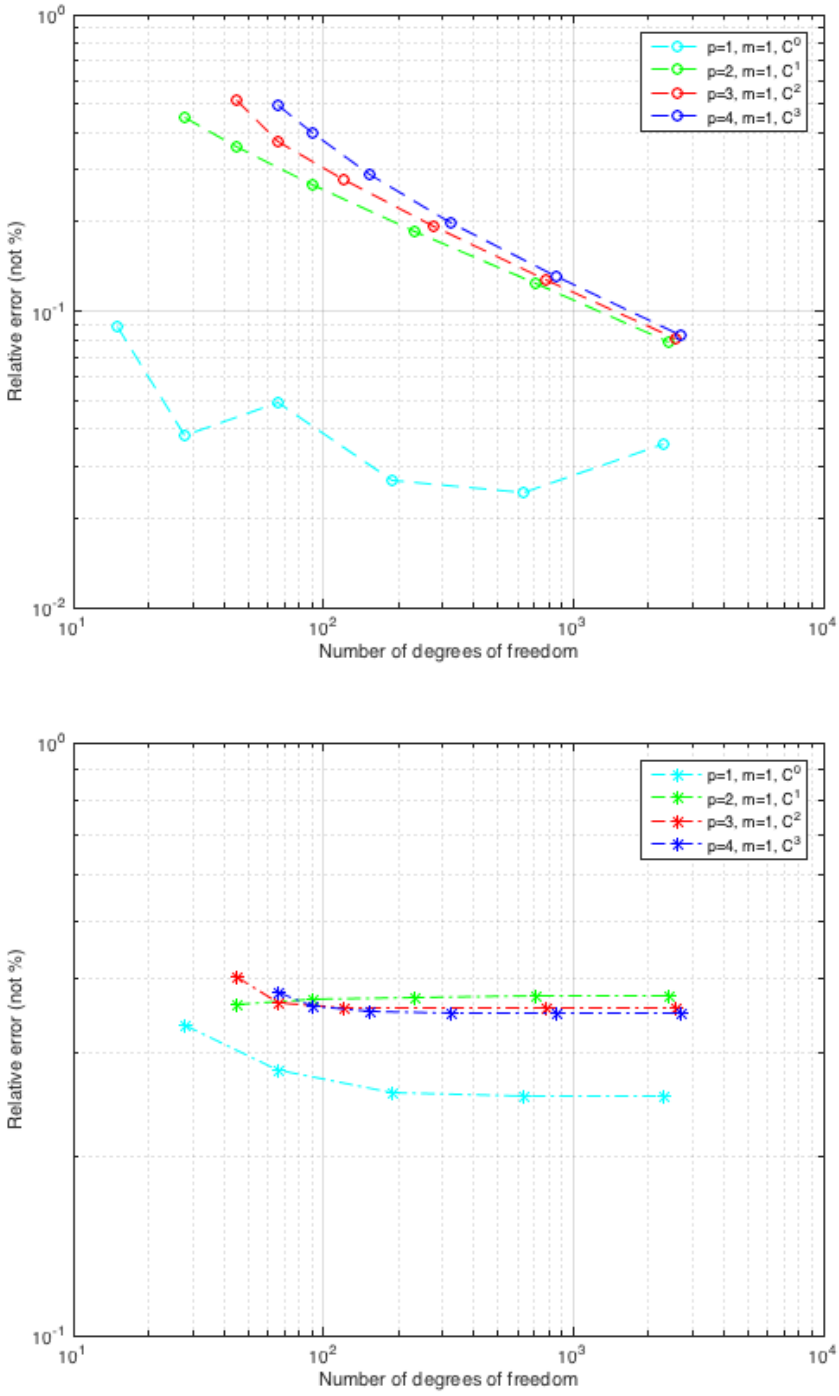

Figure 6.18: L-shape problem: This figure plots the relative error of the heat equation for the polynomial orders 1, 2, 3 and 4 for the uniform and graded mesh respectively.

Chapter 7
Conclusion

In this report the basic theory for isogeometric analysis using finite elements was introduced, with a special emphasis on B-splines and NURBS. The isogeometric solver IFEM was tested on uncoupled thermoelastic problems. The thermoelastic equation was solved on two different model problems, one where the solution was smooth and one where it was not. IFEM was able to approximate a correct convergence rate for the smooth problem, but unable to achieve a satisfactory convergence rate for the problem with a singularity. When testing to see if IFEM was able to solve the heat equation correctly for the singular problem the results were off as well.The fact that we have it on good authority that IFEM has previously been able to achieve good results for the heat equation suggests that we might have a bug in the solver, most likely the thermal solver.

The convergence rates were measured in the energy norm.

The isogeometric thermoelasticity solver was tested on different function spaces, for different polynomial orders and regularities. It was found that, for a given polynomial order, high regularity gives lower error versus degrees of freedoms compared to low regularity, without exceptions.

7.1 Future work

The next natural step for IFEM solver is to make sure the solver is able to handle singular thermoelastic problem. A good place to start would be to make sure IFEM is able to solve the heat problem correctly. Extending the solver to handle the coupled version of the thermal elasticity equations, which are problems where the deformations due to thermal expansion are big enough to cause heat, would also be a good way for the solver to evolve.

There is a possibility that the elements close to the singularity in the graded meshes are so small that the determinant of the Jacobi matrix is in fact smaller

than the solvers zero-tolerance, we would recommend a more thorough testing with different levels of refinement. It would also be interesting to see how the thermoelastic IFEM solver preform on locally refined or adaptive meshes, with LR-Bsplines or T-splines as basis functions.

Bibliography

[1] Mark Ainsworth and J. Tinsley Oden. *A posteriori Error Estimation in Finite Element Analysis*. John Wiley and Sons, Inc, 2000.

[2] Mark Ainsworth and J.Tinsley Oden. A posteriori error estimation in finite element analysis. *Computer Methods in Applied Mechanics and Engineering*, 142(1–2):1 – 88, 1997.

[3] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1–4):173 – 201, 2007.

[4] Y. Bazilevs, L. Beirao de Veiga, J.A Cotrell, T.J.R Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, (16):1031–1090, 2006.

[5] Y. Bazilevs, M.-C. Hsu, and M.A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. *Computer Methods in Applied Mechanics and Engineering*, 249–252(0):28 – 41, 2012.

[6] Kolbein Bell. *An Engineering approach to finite element analysis of linear structural mechanics problems*. Akademika Publishing, 2013.

[7] S.C Brenner and L.R Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 3 edition, 2008.

[8] H.S Carslaw and J.C Jaeger. *conduction of Heat in Solids*. Oxford University Press, 2 edition, 1959.

[9] E. Celledoni and T. Kvamsdal. Parallelization in time for thermo-viscoelastic problems in extrusion of aluminum. *International Journal for Numerical Method in Engineering*, 196(41–44):4160 – 4183, 2007.

[10] E Cohen, R.F Reisenfeld, and F Elber. *Geometric Modeling with Splines: An Introduction*. A.K. Peters, Ltd, 2001.

[11] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 79:576 – 598, 2009.

[12] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195(41–43):5257 – 5296, 2006.

[13] L. Beirão da Veida, A. Buffa, Rivas J, and G. Sangalli. Some estimates for h-p-k refinement in isogeometric analysis. *Numerische Matematik*, 118(2):271 – 305, 2011.

[14] L. Beirão da Veida, A. Buffa, G. Sangalli, and R. Vázquez. Mathematical analysis of variational isogeometric methods. *Acta numerica*, 23:157 – 287, 2014.

[15] G.E Farin. *Curves and Surfaces for CAGD, A Practical Guide*. Morgan Kaufmann Publishers, 5 edition, 1999a.

[16] G.E Farin. *NURBS Curves and Surfaces: from Projective Geometry to Practical Use*. A.K. Peters, Ltd, 2 edition, 1999b.

[17] Thomas Graetsch and K.J Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers and Structures*, 83(4–5):235 – 265, 2005.

[18] C.P Gupta and R.Prakash. *Engineering heat transfer*. Nem Chand and Bros., 1979.

[19] Thomas J.R Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publication, 2000.

[20] Thomas J.R Hughes, Yuri Bazilevs, and J. Austin Cottrell. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[21] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39–41):4135 – 4195, 2005.

[22] Kjetil André Johannessen. An adaptive isogeometric finite element analysis. Master's thesis, Norwegian University of Science and Technology, 7 2009.

[23] Kjetil André Johannessen, Trond Kvamsdal, and Tor Dokken. Isogeometric analysis using {LR} b-splines. *Computer Methods in Applied Mechanics and Engineering*, 269(0):471 – 514, 2014.

[24] George A. Keramidas and Edward C. Ting. A finite element formulation for thermal stress analysis. part i: Variational formulation. *Nuclear Engineering and Design*, 39(2–3):267 – 275, 1976.

[25] George A. Keramidas and Edward C. Ting. A finite element formulation for thermal stress analysis. part ii: Finite element formulation. *Nuclear Engineering and Design*, 39(2–3):277 – 287, 1976.

[26] K.M Okstad and T. Kvamsdal. *Simulation of Aluminum*. Springer Berlin Heidelberg, 2003.

[27] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication)*. Springer-Verlag, 2 edition, 1997.

[28] Alfio Quarteroni. *Numerical Models for Differential Problems*. Springer, 3 edition, 2008.

[29] Siv Bente Raknes. Isogeometric analysis and degenerated mappings. Master's thesis, Norwegian University of Science and Technology, 2011.

[30] R.F Riesenfeld. *Application of B-spline Approximation to Geometric Problems of Computer Aided Design*. PhD thesis, Syracuse University, 1972.

[31] D.F Rogers. *An Introduction to NURBS: With Historical Perspective*. Morgan Kaufmann Publishers, 2000. pages 577-609.

[32] Barna Szabó and Ivo Babushka. *Finite element analysis*. Wiley, 1991.

[33] S.P. Timoshenko and J.N Goodier. *Theory of elasticity*. McGraw-Hill Book Company, 1970.

[34] K.J Versprille. *Computer-aided Design Applications of the Rational B-spline Approximation Form*. PhD thesis, Syracuse University, 1975.

[35] D.V Widder. *Heat Equation*. Academic Press, 1975.

[36] Yongjie Zhang, Yuri Bazilevs, Samrat Goswami, Chandrajit L. Bajaj, and Thomas J.R. Hughes. Patient-specific vascular nurbs modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(29–30):2943 – 2959, 2007.