A comparison of accuracy and computational efficiency between the Finite Element Method and the Isogeometric Analysis for two-dimensional Poisson Problems

Per Ståle Larsen

Master of Science in Physics and Mathematics
Submission date: June 2009
Supervisor: Trond Kvamsdal, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

The purpose of this work is to study the similarities and the differences between the finite element method and the isogeometric analysis. Focus should be put on computational efficiency, and the rapport should include one or more illustrative examples.

Assignment given: 22. January 2009
Supervisor: Trond Kvamsdal, MATH

Preface

This master thesis is the result of my work with the discipline TMA4910 and completes my time as a student with NTNU. The thesis has been accomplished during the spring semester of 2009. The implementations have been executed in MATLAB, the figures have been constructed in Xfig and the thesis has been written in LaTeX.

I would like to thank my supervisor Trond Kvamsdal, who has contributed with useful advice and suggestions during my work. I would also like to thank Kjetil Andre Johannesen for interesting and meaningful discussions about NURBS and isogeometric analysis in general. I owe a big thanks to my family and all my friend who have made the time I spent with the thesis both both fun and worthwhile. Especially, I would like to thank Johannes Fauske and Jens Helge Grutle Larsen for helping me out when LaTeX would not cooperate. Finally, I would like to thank my amazing girlfriend Ida Marie Wærstad Hansen. Your support and proofreading has been of vital importance both for me and for the quality of this thesis.

Per Ståle Larsen, June 19. 2009, Trondheim

Abstract

A comparison between the finite element method (FEM) and the isogeometric analysis (IA) is performed in this thesis. In the FEM approach Lagrange basis functions, Bezier decomposition basis function and Hierarchical basis functions are considered. In the IA approach NURBS basis functions and B-splines basis functions are considered. It is shown that IA basis has a higher continuity than the FEM basis, because of the non-interpolating nature of B-splines and that IA basis functions are better suited than the FEM basis functions to represent geometry. In addition, they do not require communication with the physical domain when the knot vectors are refined. The fact that all, except for a few of the IA basis functions are equal results in accurate solutions of a one-dimensional eigenvalue problem compared with the FEM approach. A linear system of equations on the form $\underline{A}\ \underline{u}_h = \underline{b}$ derived from a two-dimensional Poisson problem is solved. Because of the recursive nature of the NURBS basis functions and the IA refinement schemes, very general and robust methods is implemented compared to the FEM approach. The IA approach result in lower condition numbers than the FEM approach and is more efficient for small errors, approximately $e < 0.0075$. It is argued that the IA approach is superior to the FEM approach, and that the IA approach has a higher potential because the numerical integration in the IA approach can be improved significantly and the fact that the condition number will be more decisive for three-dimensional problems.

Contents

1 Introduction

Since the 1960s, the *Finite Element Method* (FEM) has been one of the most common and widely used numerical method for solving ordinary differential equations (ODE) or partial differential equations (PDE). The concept was first discussed by Argyris & Kelsey (1960). Since then a vast amount of literature has emerged.

The concept of *Isogeometric Analysis* was introduced by Hughes, Cottrell & Bazilievs (2005). The main goal behind the concept was to simplify the translation between CAD (Computer Aided Design) files and FEA (Finite Element Analysis) codes. In order to accomplish this task, they use the same functions to represent both CAD and FEA. Since the first article, a number of additional papers on the subject have appeared, for instance (Cottrell, Reali, Bazilevs & Hughes 2006, Bazilevs, de Veiga, Cottrell, Hughes & Sangalli 2006, Cottrell, Hughes & Reali 2007). In addition, in the resent years new improvement in the NURBS technology such as the T-splines (Sederberg, Zhengs, Bakenov & Nasiri 2003, Sederberg, Cardon, Finnigan, Zhengs & Lyche 2004) have been introduced.

In this thesis we will compare the finite element method and the isogeometric anslysis. We aim to find out in which areas the methods are similar and in which area they differ. We will consider both the theory behind the methods, and their ability to solve simple, smooth PDEs. In addition, we will discuss how the methods can be implemented, and how the different theory behind the two methods may affect the implementation.

In section 2 of this thesis we will discuss some of the theoretical concepts behind FEM and IA. We will begin this section with a brief introduction to FEM where we wish to highlight some of the fundamental concepts behind the method. Some of the concepts we discuss are the weak statement, discretization, basis functions and elemental contribution, and how we can apply the method in order to solve ODEs and PDEs. Our approach to the theory behind FEM is inspired by the works of Patera (2003), Patera (2004) and Braess (2007). Secondly, we will give a similar introduction to IA, but because the theory behind IA is somewhat harder to comprehend then the theory behind FEM, we will discuss the theoretical tools that are needed in order to use the IA approach more thoroughly. Among the concept we discuss in IA are spline curves, B-splines, knot insertion, NURBS, patches and elements. Our approach to the theory behind IA is inspired by the works of Lyche & Mørken (2006) and Cottrell, Hughes & Bazilievs (n.d.). Finally in this section, we will compare different aspects in the two methods. Some of the aspects we will compare are distribution of the degrees of freedom (DOF), refinement, the stiffness matrix, the mass matrix and the solution of a one-dimensional eigenvalue problem. When we compare the matrices we consider three different basis functions in the FEM approach, namely Lagrange basis functions, Bezier decomposition basis functions and Hierarchical basis functions.

Section 3 of this thesis consists of a comparison between the different implementation procedures required in a FEM approach and an IA approach. We will discuss how the different theoretical concepts behind the two methods affect the implementation procedures. The implementation procedure will be separated into three steps: preprocessing, processing and postprocessing.

Numerical results are the topic in section 4 of this thesis. We will use both a FEM approach with the three mentioned basis functions and an IA approach in order to solve two Poisson problems with homogeneous Dirichlet boundary on two different domains. The first domain is a square, while the second domain is a square with a hole. The first domain will only be represented by one patch, while the second domain will be represented with both one and two patches. We are going to find out how the condition number of the stiffness matrix and the error in the numerical solution propagates with respect to the number of DOFs. In addition, we are going to find out how the error in the numerical solution propagates with respect to the total computational time. What is meant with the total computational time is the sum of the time it takes to assembly the system of algebraic equations and the time it takes to solve the system with the conjugate gradient method. We know the analytic solutions to both Poisson problems and the error in the numerical solution can thus be measured in the relative energy norm $e = (a(u-u_h, u-u_h)/a(u,u))^{1/2}$.

2 Theory

In this section of this thesis we will consider some of the theoretical concepts behind the *finite element method* and the *isogeometric analysis.* First we will give a brief introduction to the finite element method and how the method can be applied in order to solve ODEs and PDEs. Secondly we will discuss the isogeometric analysis. Because the theory behind the isogeometric analysis is somewhat harder to comprehend, we will give a more thoroughly introduction to the basic theoretical concepts than we did in the finite element method. Finally in this section we aim to highlight similarities and differences between the different approaches.

2.1 The Finite Element Method (FEM)

In general, FEM is a numerical technique for solving ordinary (ODE) or partial differential equations (PDE). Braess (2007) defines a finite element on the abstract form

FEM-1: $\mathcal{K} \subseteq \mathbb{R}^n$.
("The element domain").
FEM-2: $\mathcal{P}$ be a finite dimensional space of functions on $\mathcal{K}$.
("The shape functions").
FEM-3: Let $\mathcal{N}$ be a basis for the dual space of $\mathcal{P}$.
("The nodal basis or the degrees of freedom").
The triplet $(\mathcal{K}, \mathcal{P}, \mathcal{N})$ is called a finite element.

From a disciplinarian point of view, there are some procedures that have to be executed in order to apply the FEM approach on a differential equation. In general, these procedures will be the same for every differential equation that is solved with a FEM approach. We have to introduce *the weak statement/the variational formulation*, discretize the physical domain, choose a set of suitable basis functions, apply the Galerkin approach and finally obtain a set of algebraic equations. In order to show what these procedures mean, what they involve and how they are applied to a problem, we will illustrate with an example. Consider the strong form of a two-dimensional Poisson problem with homogeneous Dirichlet boundary conditions.

Strong Form
Domain: $\Omega = (0,1) \times (0,1)$.
Find u such that

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma \end{aligned} \tag{1}$$

for a given f.

2.1.1 The Weak Statement/Variational Formulation

In order to derive the weak statement, we introduce the linear space of functions

$$X = \{v \in H^d(\Omega) | v|_\Gamma = 0\} \equiv H_0^d(\Omega), \tag{2}$$

where X is a inner product space, or a Hilbert space (Young 1988), with zero values on the boundary, usually denoted as $H_0^d(\Omega)$, and d is the polynomial order of the functions. If we suppose that u solves the Poisson problem, then the following will be true for all $v \in X$

$$-\int_\Omega \Delta u \cdot v \; \mathrm{d}A = \int_\Omega fv \; \mathrm{d}A. \tag{3}$$

v is commonly referred to as a *test function.* If we apply the first Green's identity (Kreyszig 1999) on the left side of the equation, we obtain

$$\int_\Gamma (\nabla u \cdot \underline{n}) \cdot v \; \mathrm{d}s + \int_\Omega \nabla u \cdot \nabla v \; \mathrm{d}A = \int_\Omega fv \; \mathrm{d}A, \tag{4}$$

where $\nabla u \cdot \underline{n}$ is the derivative of u in the outward normal direction. Since v is zero along the boundary, the first integral on the left side of the equation will also be zero. The final expression can thus be stated as

$$\int_\Omega \nabla u \cdot \nabla v \; \mathrm{d}A = \int_\Omega fv \; \mathrm{d}A. \tag{5}$$

In order to write this equation more succinctly, it is common to define the bilinear functional $a(u, v)$ and the linear functional $\ell(v)$ where

$$a(u,v) = \int_\Omega \nabla u \cdot \nabla v \; \mathrm{d}A \text{ and } l(v) = \int_\Omega fv \; \mathrm{d}A. \tag{6}$$

We observe that for the Poisson problem $a(u, v)$ is symmetric, $a(u, v) = a(v, u)$, and positive definite, $a(u, u) > 0$ for all $u \neq 0$ (SPD). In general, this is not necessarily the case. When $a(u, v)$ is SPD the stiffness matrix in the linear algebraic system will also be SPD, and we can use fast iterative methods like the conjugate gradient method (CG) (Saad 1999) in order to derive the solution vector.

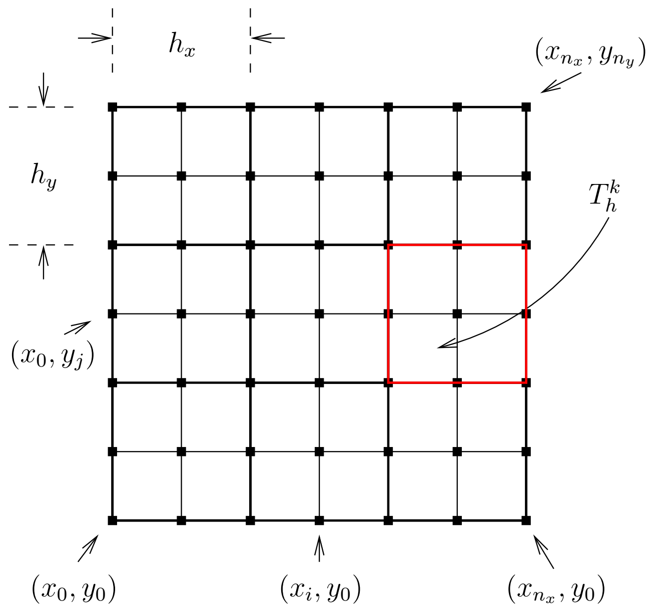
After we have introduced X, $a(u, v)$ and $\ell(v)$ the original problem can be stated as:

Weak Statement/Variational Formulation
Find $u \in X$ such that

$$a(u,v) = \ell(v) \text{ , for all } v \text{ in } X \tag{7}$$

for a given f.

Depending on the ODE/PDE, domain and boundary conditions, the functionals $a(u, v)$ and $\ell(v)$ will vary, but the structure will remain unchanged.



$\overline{\Omega} = \bigcup_{k=1}^{K} \overline{T}_h^k$ $\quad$ T_h^k, $k = 1, ..., K = (\frac{n_x}{2}) \cdot (\frac{n_y}{2})$: elements

(x_i, y_j), $i = 0, ..., n_x$ and $j = 0, ..., n_y$: nodes

Figure 1: A rectangular discretization of Ω.

2.1.2 Discretization, basis functions and the Galerkin Approach

The next procedure in order to solve the Poisson problem is to divide Ω in smaller subdomains, called *elements*, and distribute the nodal points/nodes. This procedure is called *discretization*. The most common way to discretize the domain in a FEM approach is to divide Ω in elements shaped as triangles with nodes in the corners. In this example, and also later in this thesis, we choose to use rectangles instead. This is because rectangular elements in a FEM approach will be more similar to the elements resulting from an Isogeometric approach, and thus more suitable for comparison.

The discretization of Ω is shown in figure 1. Because of the boundary conditions, the total number of *nodal variables* or *degrees of freedom* (DOF) is $(n_x - 1) \cdot (n_y - 1) = N$. The nodes are placed equidistantly on Ω and thus $h_x = h_y = h$. For problems where the error in the numerical solution is not uniformly distributed, it can be wise to let the element width vary in order to obtain faster convergence rates. We now introduce a set of elements $\mathcal{T}_h$ which contains all the elements, T_h^k, in Ω. After we have discretized Ω we introduce a new discrete function space $X_h \subset X$. The new discrete function space is defined as

$$X_h = \{v \in X \mid v|_{T_h} \in \mathbb{P}_2(T_h)\ \forall T_h \in \mathcal{T}_h\}. \tag{8}$$

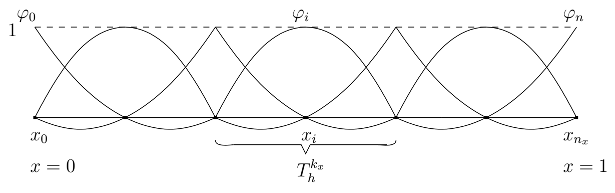


Figure 2: One dimensional quadratic Lagrange basis functions

We choose to use biquadratic *Lagrange basis functions* (Hughes 1987) in this example because the major part of the later results in this thesis use a biquadratic basis. This means that every element needs three nodes in each spatial direction. This can be viewed in figure 1. The biquadratic Lagrange basis functions are given by the tensor product

$$\varphi_i(x)\varphi_j(y), \text{ for } i = 1,...,n_x - 1 \text{ and } j = 1,...,n_y - 1. \tag{9}$$

Figure 2 shows the one dimensional quadratic Lagrange basis functions. We observe from figure 2 that the basis functions have *compact support.* This means that the basis functions are *active*, not equal to zero, on only a small part of Ω. Basis functions that interpolates in the nodes are called *nodal basis functions.* Two important properties of the Lagrange nodal basis functions are

$$\varphi_i(x_j) = 1 \text{ for } i = j \text{ , } \varphi_i(x_j) = 0 \text{ for } i \neq j \tag{10}$$

and

$$\sum_{i=0}^{n_x} \varphi_i(x) = 1 \text{ for } x \in (0, 1). \tag{11}$$

In order to simplify the notation we define a new set of functions as

$$\{\phi_1(x, y), \phi_2(x, y), ..., \phi_N(x, y)\} = \{\varphi_1(x)\varphi_1(y), \varphi_2(x)\varphi_1(y), ..., \varphi_{n_x-1}(x)\varphi_{n_y-1}(y)\} \tag{12}$$

Clearly span$\{\phi_1(x, y),...,\phi_N(x, y)\} = X_h$ and dim$\{\phi_1(x, y),...,\phi_N(x, y)\} = N$. If we apply these basis functions, all $v \in X_h$ can be written as

$$v(x) = \sum_{m=1}^{N} v_m\phi_m(x, y) \text{ , for all } v \text{ in } X_h. \tag{13}$$

If we use the same basis functions to represent the solution u_h we use what is know as the *Galerkin approach* or the *Galerkin recipe.* If we use the Galerkin approach the weak statement can be stated on the discrete form

If u in X satisfies $a(u, v) = \ell(v)$ for all v in X. Find u_h in X_h that satisfies

$$a(u_h, v) = \ell(v) \text{ , for all } v \text{ in } X_h. \tag{14}$$

2.1.3 The algebraic equations

In the previous section we derived at a discrete formulation of the weak statement. Now this is going to be the starting point in derive the linear system of equations for the Poisson problem $\underline{A}\ \underline{u}_h = \underline{b}$, where $\underline{u}_h$ is the solution vector, $\underline{A}$ is referred to as the *stiffness matrix* and $\underline{b}$ is referred to as the *load vector*. Since both u_h and v are functions in X_h, we can write them as

$$u_h = \sum_{m=1}^{N} u_{hm}\phi_m \text{ and } v = \sum_{n=1}^{N} v_n\phi_n. \tag{15}$$

If we use u_h and v, we can state the discrete formulation of the weak statement as:

Find $\underline{u}_h \in \mathbb{R}^N$ that satisfies

$$a(\sum_{m=1}^{N} u_{hm}\phi_m, \sum_{n=1}^{N} v_n\varphi_n) = \ell(\sum_{n=1}^{N} v_n\phi_n)\ \text{, for all } \underline{v} \text{ in } \mathbb{R}^N. \tag{16}$$

Since $a(u,v)$ is bilinear and symmetric and $\ell(v)$ is linear, we can write equation 16 as

$$\sum_{n=1}^{N}\sum_{m=1}^{N} v_n a(\phi_n,\phi_m)u_{hm} = \sum_{n=1}^{N} v_n \ell(\phi_n)\ \text{, for all } \underline{v} \text{ in } \mathbb{R}^N. \tag{17}$$

We can write the previous equation more succinctly as

$$\underline{v}^T \underline{A}\ \underline{u}_h = \underline{v}^T \underline{b}\ \text{, for all } \underline{v} \text{ in } \mathbb{R}^N. \tag{18}$$

If we use *unit vectors*, $\underline{e}_i = \{0,0,...,0,\underbrace{1}_{i},0,...,0,0\}$, as test vectors, the final statement becomes:

Find $\underline{u}_h \in \mathbb{R}^N$ that satisfies

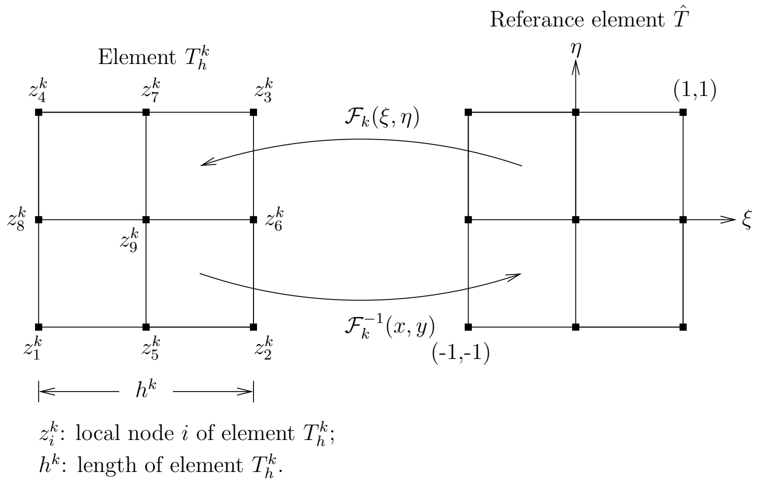
$$\underline{A}\ \underline{u}_h = \underline{b}, \tag{19}$$

where

$$\underline{A} \in \mathbb{R}^{N\times N} : A_{nm} = a(\phi_n,\phi_m) = \int_\Omega \nabla\phi_n \cdot \nabla\phi_m \ \mathrm{d}A \tag{20}$$

and

$$\underline{b} \in \mathbb{R}^N : b_n = \ell(\phi_n) = \int_\Omega f\phi_n \ \mathrm{d}A. \tag{21}$$



Element T_h^k
z_4^k
z_7^k
z_3^k
z_8^k
z_9^k
z_6^k
z_1^k
z_5^k
z_2^k
h^k
z_i^k: local node i of element T_h^k;
h^k: length of element T_h^k.
$\mathcal{F}_k(\xi,\eta)$
$\mathcal{F}_k^{-1}(x,y)$
Referance element $\hat{T}$
η
ξ
(1,1)
(-1,-1)

Figure 3: The local numbering of the nodes, the reference element and the mapping between the physical element and the reference element.

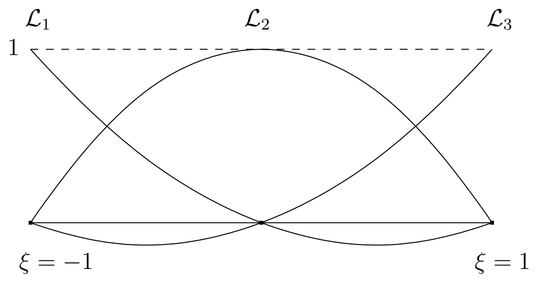


Figure 4: The one dimensional reference basis functions.

2.1.4 Elemental matrices and loads

We have shown how a continuous ODE or PDE can be solved numerically by applying test vectors, discretization and basis functions, but we have not discussed the elements. One of the most important features with the FEM approach is the compact support of the basis functions. We can thus build the stiffness matrix by finding the contribution from each element, where only a small amount of basis functions are active, and assembly the contributions in the stiffness matrix. In order to do that, we have to introduce local nodes in each of the elements that correspond to a global node. In addition, we introduce a *standard reference element*, a mapping between the reference element and the real elements, and a biquadratic basis for the *reference element space* $\hat{X} = \mathbb{P}_2(\hat{T})$. These procedures are shown in figure 3 and figure 4. In our example $h_k = h$, for all T_h^k in $\mathcal{T}_h$.

Because of the compact support of the basis functions, only nine basis functions are active on each element. From this it follows that the contribution to the stiffness matrix from element T_h^k can be written as

$$\sum_{i=1}^{9}\sum_{j=1}^{9}\int_{T_h^k}(\nabla\phi_i^k)^T\nabla\phi_j^k \,\mathrm{d}A, \tag{22}$$

where ϕ_i^k denotes local basis function i on element T_h^k. In general, the evaluation of this integral is done with numerical integration. In order to apply numerical integration we have to express the integral in terms of reference coordinates. In our example the mapping between the physical element and the reference element is affine (Braess 2007) and can be stated as

$$\begin{aligned} x(\xi) &= z_1^k(x)+\tfrac{h}{2}(1+\xi) \\ y(\eta) &= z_1^k(y)+\tfrac{h}{2}(1+\eta). \end{aligned} \tag{23}$$

If we use the affine mapping we derive

$$\begin{bmatrix}\mathrm{d}x\\ \mathrm{d}y\end{bmatrix}=\underbrace{\begin{bmatrix}\frac{h}{2} & 0\\ 0 & \frac{h}{2}\end{bmatrix}}_{\underline{J}}\begin{bmatrix}\mathrm{d}\xi\\ \mathrm{d}\eta\end{bmatrix} \text{ and } \nabla\phi_i^k=\begin{bmatrix}(\phi_i^k)_x\\ (\phi_i^k)_y\end{bmatrix}=\underbrace{\begin{bmatrix}\frac{2}{h} & 0\\ 0 & \frac{2}{h}\end{bmatrix}}_{(\underline{J}^{-1})^T}\begin{bmatrix}(\mathcal{L}_i)_\xi\\ (\mathcal{L}_i)_\eta\end{bmatrix}=\hat{\nabla}\mathcal{L}_i, \tag{24}$$

where $\underline{J}$ is called the *Jacobi matrix* and the determinant of $\underline{J}$, J, is called the *Jacobian.* Rønquist (2008*a*) shows that

$$\int_{T_h^k}\mathrm{d}A=\int_{\hat{T}}J\,\mathrm{d}\hat{A}. \tag{25}$$

Equation 22 can thus be stated in terms of reference variables as

$$\sum_{i=1}^{9}\sum_{j=1}^{9}\int_{\hat{T}}((\underline{J}^{-1})^T\hat{\nabla}\mathcal{L}_i)^T((\underline{J}^{-1})^T\hat{\nabla}\mathcal{L}_j)J\,\mathrm{d}\hat{A}=\sum_{i=1}^{9}\sum_{j=1}^{9}\int_{\hat{T}}\hat{\nabla}\mathcal{L}_i\hat{\nabla}\mathcal{L}_j\,\mathrm{d}\hat{A}. \tag{26}$$

In general, we can not remove J and $(\underline{J}^{-1})^T$ from the integral, but in our example we can do just that because $(2/h)^2\cdot(h/2)^2=1$. In order to evaluate the final integral we use numerical integration. For every element in $\mathcal{T}_h$, except for the elements along the boundary, the contribution from the *elemental stiffness matrix* can be stated as

$$\underline{A}^k = \frac{1}{90}\begin{pmatrix} 56 & -3 & -2 & -3 & -18 & 10 & 10 & -18 & -32 \\ -3 & 56 & -3 & -2 & -18 & -18 & 19 & 10 & -32 \\ -2 & -3 & 56 & -3 & 10 & 10 & -18 & -18 & -32 \\ -3 & -2 & -3 & 56 & 10 & 10 & -18 & -18 & -32 \\ -18 & -18 & 10 & 10 & 176 & -32 & 0 & -32 & -96 \\ 10 & -18 & -18 & 10 & -32 & 176 & -32 & 0 & -96 \\ 10 & 10 & -18 & -18 & 0 & -32 & 176 & -32 & -96 \\ -18 & 10 & 10 & -18 & -32 & 0 & -32 & 176 & -96 \\ -32 & -32 & -32 & -32 & -96 & -96 & -96 & -96 & 512 \end{pmatrix}. \tag{27}$$

In order to assemble the contribution from each element, we use the relations between local and global nodes.

If we apply a similar procedure on the load vector, the contribution from element k can be written as

$$\sum_{i=1}^{9} \int_{T_h^k} f\phi_i^k \; \mathrm{d}A \tag{28}$$

or in terms of reference variables

$$\sum_{i=1}^{9} \int_{\hat{T}} f\mathcal{L}_i J \; \mathrm{d}\hat{A}. \tag{29}$$

Similar to how we solved equation 26, the integral is evaluated with numerical integration.

2.2 Isogeometric Analysis (IA)

In the same manner as for the finite element method, isogeometric analysis is a numerical technique for solving ODEs and PDEs. The philosophy behind IA is that we want to use the same basis functions to represent both the physical domain and the solution. In addition, we want to be able to maintain the representation of the geometry if we include more DOFs. Because of this approach, we want to use basis functions that are well suited to represent different geometries. In the IA approach we use *B-splines* and *NURBS* as basis functions, where NURBS in particular, are very well suited to represent geometries.

The way IA is applied in order to solve a ODE or PDE is very similar to the FEM approach (weak statement, discretization, basis functions and algebraic equation). Because of this similarity, the main focus in this section will be on explaining some of the concept and tools that are needed in order to use an IA approach. We aim to show how B-splines and NURBS are constructed and which abilities they possess. In addition, we will discuss what is understood by *elements* and *patches* in an IA approach.

2.2.1 Spline Curves

A straight line between two points in space $\mathbf{c}_1$ and $\mathbf{c}_2$ can be represented by the convex combination

$$\mathbf{p}(t|\mathbf{c}_1, \mathbf{c}_2; t_2, t_3) = \frac{t_3 - t}{t_3 - t_2}\mathbf{c}_1 + \frac{t - t_2}{t_3 - t_2}\mathbf{c}_2,\ t \in [t_2, t_3]. \tag{30}$$

When we use this representation, we demand that $t_2 < t_3$. If we want to generalize this approach in order to represent a piecewise linear curve $\mathbf{f}$ between a given set of points $((\mathbf{c}_i)_{i=1}^n)$, we choose *n parametric values* $(t_i)_{i=2}^{n+1}$, with $t_i < t_{i+1}$ for all i, and define $\mathbf{f}$ by

$$\mathbf{f}(t) = \begin{cases} \mathbf{p}(t|\mathbf{c}_1, \mathbf{c}_2; t_2, t_3), & t \in [t_2, t_3) \\ \mathbf{p}(t|\mathbf{c}_2, \mathbf{c}_3; t_3, t_4), & t \in [t_3, t_4) \\ \vdots & \vdots \\ \mathbf{p}(t|\mathbf{c}_{n-1}, \mathbf{c}_n; t_n, t_{n+1}), & t \in [t_n, t_{n+1}) \end{cases}. \tag{31}$$

$\mathbf{f}$ is called a *linear spline curve*, the points $(\mathbf{c}_i)_{i=1}^n$ are called *the control points* of the curve, and the parametric values $\mathbf{t} = (t_i)_{i=2}^{n+1}$ are referred to as the *knots*, or *the knot vector*, of the curve. In order to simplify the notation we introduce the piecewise constant function $B_{i,0}(t)$ defined by

$$B_{i,0}(t) = \begin{cases} 1, & t_i \le t < t_{i+1} \\ 0, & \text{otherwise} \end{cases}. \tag{32}$$

This notation will also be useful later in this thesis when we discuss B-splines. In addition, we define $\mathbf{p}_{i,1}(t) = \mathbf{p}(t|\mathbf{c}_{i-1}, \mathbf{c}_i; t_i, t_{i+1})$. Finally we can write $\mathbf{f}(t)$ more succinctly as

$$\mathbf{f}(t) = \sum_{i=2}^{n} \mathbf{p}_{i,1}(t) B_{i,0}(t). \tag{33}$$

Lyche & Mørken (2006) show how this construction can be used in order to recursively produce smooth, piecewise polynomial curves of higher degrees, and how every spline curve constructed in this manner is contained within the *convex hull* of the control points and thus is numerically stable. An example of a quadratic spline curve and the convex hull of the control points is shown in figure 5.

2.2.2 B-splines

Given n control points $(\mathbf{c}_i)_{i=1}^n$ and $n + d - 1$ knots $\mathbf{t} = (t_i)_{i=1}^{n+d}$, Lyche & Mørken (2006) show how we can construct a spline curve of degree d of the form

$$\mathbf{f}(t) = \sum_{i=d+1}^{n} \mathbf{p}_{i,d}(t) B_{i,0}(t),\ t \in [t_{d+1}, t_{n+1}], \tag{34}$$

where $\{B_{i,0}(t)\}_{i=d+1}^n$ and $\mathbf{p}_{i,d}$ are defined in the same manner as in the previous section, but now we allow subsequent knot values to be equal, $t_i = t_{i+1}$. If two subsequent knot

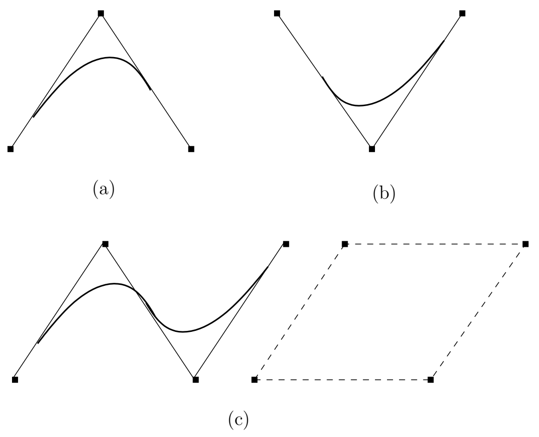


Figure 5: A quadratic spline curve and the convex hull of the control points (c), and the quadratic spline curve's two polynomial segments (a) and (b).

values are equal, we obtain what is called *repeated knots values.* Repeated knot values affect the continuity of the spline curve and will be discussed later in this thesis. A knot vector where the first and last element appears $d + 1$ times is referred to as an *open knot vector.* Open knot vectors will be very useful later in this thesis. If we allow repeated knot values, we get division by zero in the $\mathbf{p}_{i,d}$'s. This is solved by introducing the relation

$$B_{i,1}(t) = \frac{t - t_i}{t_{i+1} - t_i} B_{i,0}(t) + \frac{t_{i+2} - t}{t_{i+2} - t_{i+1}} B_{i+1,0}(t), \tag{35}$$

where $B_{i,0} = 0$ when $t_i = t_{i+1}$ by definition. By adding two extra knots, t_1 and t_{n+d+1}, Lyche & Mørken (2006) show that we can write the spline curve $\mathbf{f}(t)$ as

$$\mathbf{f}(t) = \sum_{i=1}^{n} \mathbf{c}_i B_{i,d}(t) \tag{36}$$

where $B_{i,d}(t)$ is given by the recurrence relation

$$B_{i,d}(t) = \frac{t - t_i}{t_{i+d} - t_i} B_{i,d-1}(t) + \frac{t_{i+1+d} - t}{t_{i+1+d} - t_{i+1}} B_{i+1,d-1}(t), \tag{37}$$

where $(\mathbf{c}_i)_{i=1}^{n}$ are the control points, d is the degree of the spline curve and $(t) = (t_i)_{i=1}^{n+d+1}$ is the knot vector with nondecreasing knot values. The function $B_{i,d}$ is called a B-spline of degree d or a B-spline basis function of degree d. In the context of IA, it is common to denote the B-spline basis function as $N_{i,p}(\xi)$, where p is the degree of the B-spline, and to denote the knot vector as $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$, where the ξ's are nondecreasing parametric values. When we later refer to all values of ξ, we assume that ξ is contained in the interval $[\xi_1, \xi_{n+p+1}]$ An illustration of constant, linear and quadratic B-spline basis functions can be viewed in figure 6. In addition, an example of a spline curve with an

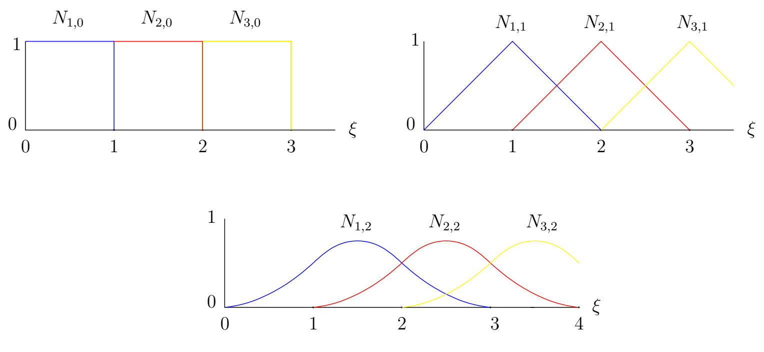


Figure 6: Constant, linear and quadratic B-spline basis functions.

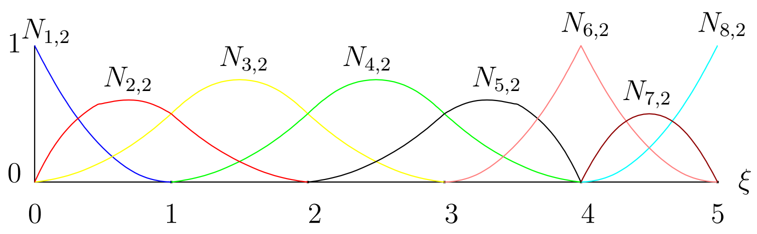


Figure 7: Quadratic B-spline functions with knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$.

open knot vector containing repeated knot values is shown in figure 7.

Both Cottrell et al. (n.d.) and Lyche & Mørken (2006) discuss the nature and properties of the B-spline basis functions, and a collection of some of the most important properties of the B-spline basis functions is:

1. They constitute a partition of unity, that is, for all ξ

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1 \tag{38}$$

2. The support of each $N_{i,p}$ is compact and contained in the interval $[\xi_i, \xi_{i+p+1}]$.

3. Every $N_(i,p)$ is non-negative for all ξ. Consequently, every coefficient of a mass matrix computed from a B-spline basis is greater than, or equal to, zero.

4. The B-splines are linearly independent, and thus the dimension of span$\{N_{1,p}, \dots, N_{n,p}\}$ equals n.

5. If the number z occurs m times among $\xi_i, \dots, \xi_{i+d+1}$ then the derivatives of $N_{i,p}$

of order $0, 1, \ldots, p-m$ are all continous at z.

6. The derivatives of the basis functions can be computed recursively.

If we use a B-spline basis, we can represent a wide variety of curves, surfaces and solids. Surfaces or solids are represented by a tensor products of B-spline basis functions with knot vectors containing different parameters. Given a set of control points, a *control net*, $\{B_{i,j}\}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$ and knot vectors $\Xi = \{\xi_1, \ldots, \xi_{n+p+1}\}$ and $\Theta = \{\eta_1, \ldots, \eta_{m+q+1}\}$, a tensor product B-spline surface is represented by the tensor product

$$S(\xi, \eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,p}(\xi)M_{j,q}(\eta)B_{i,j}, \tag{39}$$

where $N_{i,p}$ and $M_{j,q}$ are basis functions of B-spline curves.

2.2.3 Knot insertion

As mention earlier in this section, we want to be able to maintain the representation of the geometry when we include more DOFs. If we use B-spline basis functions, there exists a procedure called *knot insertion* which enable us to maintain the exact same representations. Knot insertion, also called h-refinement, is a refinement procedure where knots are inserted in an existing knot vector without changing a curve geometrically or parametrically. We will in this thesis only discuss some of the basic ideas behind the knot insertion procedure. In order to learn in detail how the procedure is applied and why it works we refer to Lyche & Mørken (2006). Insertion of knot values has parallels with the classical h-refinement in FEM, but the repeating of existing knot values in order to decrease the continuity in the basis does not have an analogue in FEM.

The basic idea behind the knot insertion procedure is that if a curve can be represented by

$$\mathbf{f}(\xi) = \sum_{i=1}^{n} B_i N_{i,p}(\xi) \tag{40}$$

where in $\Xi = \{\xi_1, \xi_2, ..., \xi_{n+p+1}\}$, then it must be possible to represent the same curve by

$$\mathbf{f}(\xi) = \sum_{i=1}^{n+m} \hat{B}_i \hat{N}_{i,p}(\xi) \tag{41}$$

where $\hat{\Xi} = \{\hat{\xi}_1 = \xi_1, \hat{\xi}_2, ..., \hat{\xi}_{n+m+p+1} = \xi_{n+p+1}\} \supset \Xi$, because the space spanned by the functions $\{N_{1,p}, ..., N_{n,p}\}$ is a subspace of the space spanned by the functions $\{\hat{N}_{1,p}, ..., \hat{N}_{n,p}\}$. $n+m$ new basis functions are constructed by using equation 37 and the extended knot vector. Cottrell et al. (2007) show that the new $n+m$ control points, $\underline{\hat{B}} = \{\hat{B}_1, \hat{B}_2, ..., \hat{B}_{n+m}\}^T$, are formed from the original control points, $\underline{B} = \{B_1, B_2, ..., B_n\}^T$, by the relation

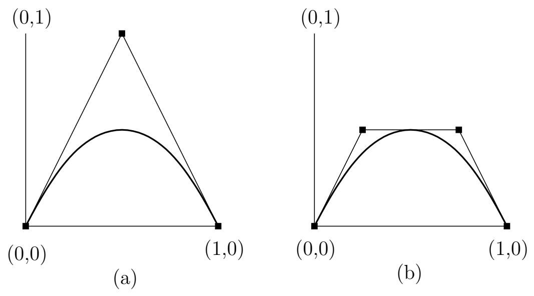


Figure 8: Figure (a) shows the original curve with three control points and knot vector $\Xi = \{0, 0, 0, 1, 1, 1\}$. Figure (b) shows the refined curve with four control points and knot vector $\hat{\Xi} = \{0, 0, 0, 0.5, 1, 1, 1\}$.

$$\underline{\hat{B}} = \underline{T}^p \underline{B} \tag{42}$$

where

$$T_{ij}^0 = \begin{cases} 1 & \text{if } \hat{\xi}_i \text{ in } [\xi_j, \xi_{j+1}), \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

and

$$T_{ij}^{q+1} = \frac{\hat{\xi}_{i+q} - \xi_j}{\xi_{j+q} - \xi_j} T_{ij}^q + \frac{\xi_{j+q+1} - \hat{\xi}_{i+q}}{\xi_{j+q+1} - \xi_{j+1}} T_{ij+1}^q \tag{44}$$

for $q = 0, 1, 2, ..., p - 1$. An example of a single knot value insertion is shown in figure 8. In this thesis we will only use knot insertion as a refinement strategy and thus maintain the same degree on the B-spline basis functions, but there exists other refinement strategies as well. Other B-spline refinement strategies are *p-refinement*, order elevation, and *k-refinement*, higher order and higher continutiy Cottrell et al. (n.d.).

2.2.4 NURBS

In this section we going to discuss the concept of NURBS. We aim to show how they are constructed, what their advantages are and what separates them from B-splines. NURBS is an abbreviation for *Non Uniform Rational B-Splines*, and they are, as the name implies constructed from B-splines. If we alter the values in the knot vector the resulting curves will be different. This change in the curves is what is meant by non uniform, and is an ability both NURBS and B-splines possess. If we use NURBS we will have to assign a *weight* to each of the control points. What is meant by rational is that the weights do not have to be equal. In fact, if all the weights are equal NURBS become B-splines.

The main advantage of NURBS compared to B-splines is their ability to represent geometric entities. NURBS can precisely represent conic sections, such as circles and ellipses,

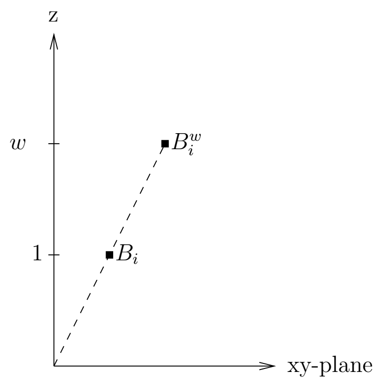


Figure 9: The projective control point in $\mathbb{R}^3$, B_i^w, with coordinates (x_i, y_i, w_i) and the control point in $\mathbb{R}^2$, B_i, with coordinates $(x_i/w_i, y_i/w_i)$ and weight w_i.

by projective transformations of piecewise quadratic curves. This is not possible with B-splines. Cottrell et al. (n.d.) show that desired geometric entities in $\mathbb{R}^d$ can be obtained by projective transformations of B-spline entities in $\mathbb{R}^{d+1}$. The set of control points $\{B_i^w\}$ for a B-spline curve in $\mathbb{R}^{d+1}$ with knot vector Ξ are referred to as *the projective control points* for the desired NURBS curve in $\mathbb{R}^d$. The control points in $\mathbb{R}^d$ are derived by the relation

$$(B_i)_j = (B_i^w)_j/w_i,\ j = 1, \dots, d. \tag{45}$$

An example of the projection of one projective control point in $\mathbb{R}^3$ to one control point in $\mathbb{R}^2$ is shown in figure 9.

The NURBS basis functions and NURBS curves are given by

$$R_i^p(\xi) = \frac{N_{i,p}(\xi)w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi)w_{\hat{i}}} \tag{46}$$

and

$$C(\xi) = \sum_{i=1}^n R_i^p(\xi)B_i. \tag{47}$$

The expansion to rational surfaces and solids are done in the same manner as with B-splines. From Hughes et al. (2005) we learn that some om the important properties of NURBS are:

1. NURBS basis functions form a partition of unity.

2. The continuity and support of NURBS basis functions are the same as for B-splines.

3. Affine transformations in physical space are obtained by applying the transformation to the control points, that is, NURBS possess the property of *affine covariance.*

4. If weights are equal, NURBS become B-splines.

5. NURBS surfaces and solids are the projective transformations of tensor product, piecewise polynomial entities.

2.2.5 Patches and Elements

We have earlier in this section defined the different basis functions we use in an IA approach and which properties they possess. Now we want to explain what a patch is and how an element is defined in an IA approach. In the literature there exists different opinions on how the "elements" should be defined. Kagan, Fischer & Bar-Yoseph (1998) refers to a patch as an element, but we are in this thesis going to follow the definitions that are used by Cottrell et al. (n.d.).

A physical domain can be represented by one or several patches. A patch consists of a control net and the mapping of the parametric space on to the physical space. If we consider figure 10, we can see how the physical domain (a) is represented by one patch consisting of the control net (b) and the mapping from the parametric space on to the physical space (c). The parametric space is local to patches and within each patch material modes are assumed to be uniform. Many simple domains can be represented by only one patch.

Instead of using subdomains of the physical domain, patches play the role of subdomains in an IA approach, to define elements as we did in the finite element context, we will now use the knot vectors in the parametric space to define the elements. The parametric space is illustrated in figure 10 (e). If two subsequent knots in a knot vector are different from each other, they define a element in the parametric space. In the same manner, all knots in the knot vector partition the parametric space into elements. If $\xi_i = \xi_{i+1}$ or $\eta_j = \eta_{j+1}$, then $[\xi_i, \xi_{i+1}] \times [\eta_j, \eta_{j+1}]$ does not define an element. When we use NURBS or B-spline as the basis functions, we use open knot vectors in order to interpolate the control points on the boundary of the patches. Because the first and last value are repeated $p + 1$ times, only the knots $\{\xi_{p+1}, ..., \xi_{n+1}\}$ partition the parametric space into elements. The partition of the parametric space as a result of open knot vectors can be viewed in figure 11.

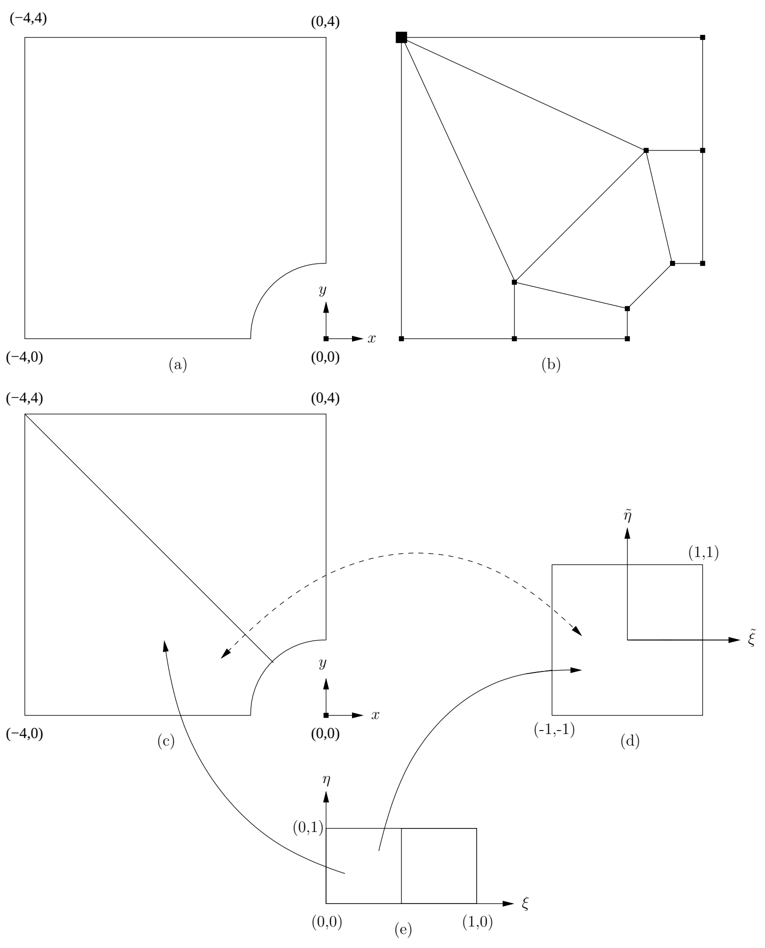
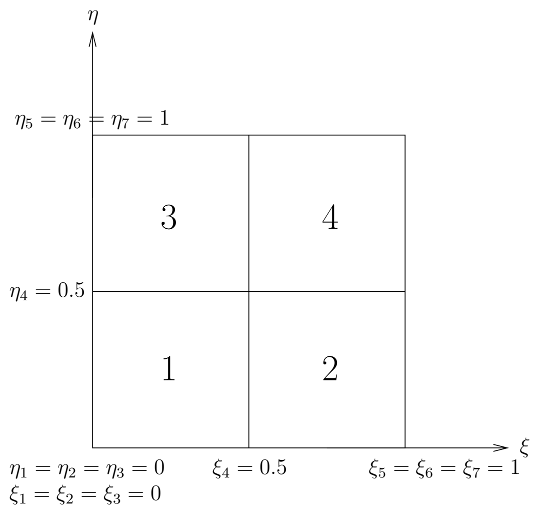


Figure 10: The IA approach with one patch and NURBS basis functions of degree two. (a) the physical domain, (b) the control net, (c) the mapping of the parametric space on to the physical space, (d) the reference element, (e) the parametric space. In addition, the different mappings beween (c), (d) and (e) are illustrated.



η
$\eta_5 = \eta_6 = \eta_7 = 1$
3
4
$\eta_4 = 0.5$
1
2
ξ
$\eta_1 = \eta_2 = \eta_3 = 0$
$\xi_1 = \xi_2 = \xi_3 = 0$
$\xi_4 = 0.5$
$\xi_5 = \xi_6 = \xi_7 = 1$

Figure 11: The partition of the parametric space with the knot vectors $\Xi = \{0, 0, 0, 0.5, 1, 1, 1\}$ and $\Theta = \{0, 0, 0, 0.5, 1, 1, 1\}$. These knot vectors results in four elements.

Compared to the FEM approach, there is an additional mapping in the IA approach. This mapping can be viewed in figure 10. In order to show how this additional mapping affects the calculations, we illustrate with an example. Consider the two-dimensional domain in figure 10 which is represented with one patch and two elements. We want to solve $a(u, v)$ on one of the elements with numerical integration.

$$a(u, v)_{T^1} = \int_{T^1} (\nabla v(x, y))^T \cdot \nabla u(x, y) \, \mathrm{d}x\mathrm{d}y. \tag{48}$$

We know that there exists mappings such that

$$\begin{array}{rcl} x &=& x(\xi, \eta) \\ y &=& y(\xi, \eta) \\ \xi &=& \xi(\hat{\xi}) \\ \eta &=& \eta(\hat{\eta}) \end{array} . \tag{49}$$

We can thus express $u(x, y)$ as

$$u(x, y) = u(x(\xi, \eta), y(\xi, \eta)) = u(x(\xi(\tilde{\xi}), \eta(\tilde{\eta})), y(\xi(\tilde{\xi}), \eta(\tilde{\eta}))) = \tilde{u}(\tilde{\xi}, \tilde{\eta}). \tag{50}$$

The same can obviously be done for $v(x, y)$. First we consider the mapping between the physical space and the parametric space

$$\begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \end{bmatrix} = \begin{array}{c} \frac{\partial x}{\partial \xi}\mathrm{d}\xi + \frac{\partial x}{\partial \eta}\mathrm{d}\eta \\ \frac{\partial y}{\partial \xi}\mathrm{d}\xi + \frac{\partial y}{\partial \eta}\mathrm{d}\eta \end{array} = \underbrace{\begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}}_{\underline{J}_{x\xi}} \begin{bmatrix} \mathrm{d}\xi \\ \mathrm{d}\eta \end{bmatrix}. \tag{51}$$

Here $\underline{J}_{x\xi}$ is the Jacobi matrix for the mapping between the physical space and the parametric space. Next we consider the mapping between the parametric space and the parent element

$$\begin{bmatrix} \mathrm{d}\xi \\ \mathrm{d}\eta \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{\partial \xi}{\partial \tilde{\xi}} & 0 \\ 0 & \frac{\partial \eta}{\partial \tilde{\eta}} \end{bmatrix}}_{\underline{J}_{\xi\tilde{\xi}}} \begin{bmatrix} \mathrm{d}\tilde{\xi} \\ \mathrm{d}\tilde{\eta} \end{bmatrix}. \tag{52}$$

Here $\underline{J}_{\xi\tilde{\xi}}$ is the Jacobi matrix for the mapping between the parametric space and the reference space. With the two Jacobi matrices, we can calculate the Jacobi matrix for the mapping between the physical space and the reference space as

$$\underline{J}_{x\tilde{\xi}} = \underline{J}_{x\xi} \cdot \underline{J}_{\xi\tilde{\xi}}. \tag{53}$$

In the same manner as earlier in this thesis, we can express the area of T^1 in terms of reference variables

$$\int_{\Omega^k} \mathrm{d}x\mathrm{d}y = \int_{\tilde{\Omega}} \det(\underline{J}_{x\tilde{\xi}})\mathrm{d}\tilde{\xi}\mathrm{d}\tilde{\eta} = \int_{\tilde{\Omega}} J_{x\tilde{\xi}}\mathrm{d}\tilde{\xi}\mathrm{d}\tilde{\eta}. \tag{54}$$

The chain rule is applied in order to express ∇u and ∇v in terms of reference variables

$$\underbrace{\begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix}}_{\nabla u} = \begin{matrix} \frac{\partial \tilde{u}}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \tilde{u}}{\partial \tilde{\eta}}\frac{\partial \tilde{\eta}}{\partial \eta}\frac{\partial \eta}{\partial x} \\ \frac{\partial \tilde{u}}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \tilde{u}}{\partial \tilde{\eta}}\frac{\partial \tilde{\eta}}{\partial \eta}\frac{\partial \eta}{\partial y} \end{matrix} = \underbrace{\underbrace{\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix}}_{(\underline{J}_{x\xi}^{-1})^T} \underbrace{\begin{bmatrix} \frac{\partial \tilde{\xi}}{\partial \xi} & 0 \\ 0 & \frac{\partial \tilde{\eta}}{\partial \eta} \end{bmatrix}}_{(\underline{J}_{\xi\tilde{\xi}}^{-1})}}_{\underline{G}_{\nabla}} \underbrace{\begin{bmatrix} \frac{\partial \tilde{u}}{\partial \tilde{\xi}} \\ \frac{\partial \tilde{u}}{\partial \tilde{\eta}} \end{bmatrix}}_{\tilde{\nabla}\tilde{u}}. \tag{55}$$

Finally, $a(u,v)_{T^1}$ can be expressed in term of reference variables as

$$a(u,v)_{T^1} = \int_{\tilde{\Omega}} (\underline{G}_{\nabla}\tilde{\nabla}\tilde{v})^T(\underline{G}_{\nabla}\tilde{\nabla}\tilde{u})J_{x\tilde{\xi}}\mathrm{d}\tilde{\xi}\mathrm{d}\tilde{\eta} = \int_{\tilde{\Omega}} (\tilde{\nabla}\tilde{v})^T\underline{G}(\tilde{\nabla}\tilde{u})\mathrm{d}\tilde{\xi}\mathrm{d}\tilde{\eta} \tag{56}$$

where

$$\underline{G} = J_{x\tilde{\xi}}\underline{G}_{\nabla}^T\underline{G}_{\nabla}. \tag{57}$$

The final integral is solved by introducing quadrature points and weights.

Table 1: Comparison of the FEM and the IA based on NURBS

FEM	Both	IA
Nodal points		Control points
Nodal variables		Control variables
Mesh		Knots
Basis interpolate		Basis does not interpolate
nodal points and variables		control points and variables
Approximate geometry		Exact geometry
Polynomial basis		NURBS basis
Subdomains		Patches
	Compact support	
	Partition of unity	

2.3 Comparison of the FEM and the IA

So far in this thesis, we have discussed the theory behind the FEM and the IA, and how we can apply the different approaches on ODEs and PDEs. In this section we are going to use this theory in order to compare the two approaches. We aim to find out where the similarities and differences lie, and how they may affect the procedures we have to execute in order to solve an ODE or a PDE.

The first thing we will do is to highlight some of the disciplinarian concepts in the two approaches. After we have discussed the disciplinarian concepts, we will discuss how nodal points and control points are distributed, how we refine a mesh and knot vectors, and finally what the stiffness matrices and mass matrices will look like.

2.3.1 Overwiev over different aspects

Table 1 shows a comparison between the FEM and the IA with NURBS as basis. Perhaps the most important difference between the FEM and the IA is the fact that a nodal basis interpolates nodal points and variables, while a NURBS basis does not interpolate control points and variables. In addition, nodal basis functions can be both positive and negative, while NURBS basis functions only are positive. Cottrell et al. (n.d.) shows how these features result in the Gibbs phenomena with a nodal basis and the variation diminishing property with a NURBS basis. Of the shared features, the most important property is the compact support which enable us to use an elemental approach.

2.3.2 Distribution of nodal points and control points

In order to get the best possible conditions for a comparison between the FEM approach and the IA approach when we solve an ODE or a PDE, we would like the elements and DOFs to be distributed as similar as possible in both approaches. If we use basis functions of degree one this task is trivial, because the basis functions are equal in both approaches. On the other hand, for basis functions with a degree higher then one, it is not possible to use the same distribution in both approaches. We will show why this is the case.

One way to refine a two-dimensional domain is to use bisection on the elements, which means that one element splits into four new elements by dividing the existing element in half in both directions. In a FEM approach, we divide the physical space in order to refine the mesh. This is not the case in an IA approach. As we have seen in the theory section, we use knot insertion when we refine the mesh. This means that we do not divide the physical space, but the parametric space. Because the NURBS basis functions near the boundary of the domain are not equal to the those in the interior, for $p > 1$, the control points are not evenly spaced, but the elements are. We assume that that the knot vectors we use are uniformly distributed. If we choose to use evenly spaced control points instead, the elements size will vary. These two representations are referred to as *linear parameterization* and *nonlinear parameterization.* In this thesis we have only used linear parameterization, but this is not nessecarily the best representation. Cottrell et al. (2006) show that nonlinear parameterization gives better results when studying structural vibrations.

Figure 28 shows an example of linear and nonlinear parameterization. If we consider the control net in the nonlinear parameterization in the figure, we observe that this control net is precisely the mesh we would use in a FEM approach with four elements. In an IA approach the same control net requires nine elements. In addition, the elements we derive in the linear parameterization, is the same elements we would derive in a FEM approach with 49 uniformly distributed nodes. From the figure we also observe that in an IA approach the number of DOFs equals the total number of elements. This is not the situation in a FEM approach. Table 2 shows how the number of elements and DOFs propagate in a one-dimensional refinement situation. A similar behavior will be present in higher dimensions. The fact that the number of DOFs compared to the number of elements differ in the FEM approach and the IA approach, and the fact that the two IA representations both have similarities to the FEM approach, tells us that for $p > 1$ the elements and DOFs can not be distributed equally in both approaches.

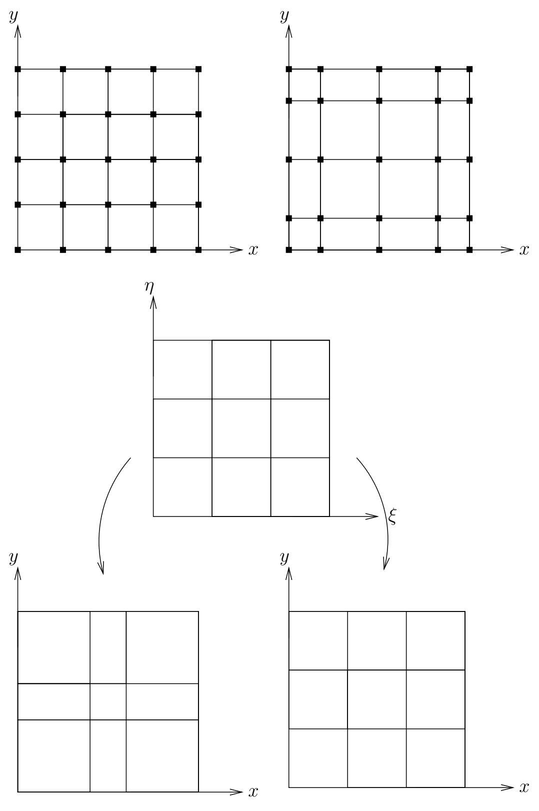


Figure 12: Nonlinear parameterization on the left and linear parametrization on the right.

Table 2: One-dimensional refinement with bisection.

Number of refinements	FEM		IA	
	Elements	Nodes	Elements	Control points
0	1	3	1	3
1	2	5	2	4
2	4	9	4	6
⋮	⋮	⋮	⋮	⋮
n	2^n	$2^{n+1}+1$	2^n	2^n+2

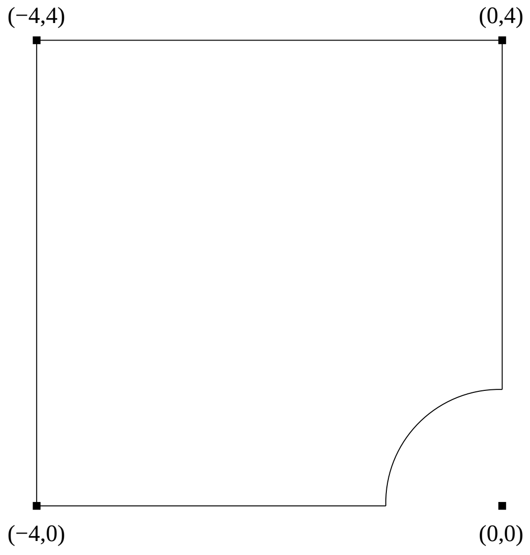


Figure 13: The domain $\Omega = x \in (-4, 0) \cap y \in (0, 4) \cap (x^2 + y^2 \geq 1)$.

2.3.3 Refinement

As we have mentioned earlier in this thesis, the refinement process is different in the FEM and IA approaches. In order to show some of the different features, when using h-refinement, we will now consider a two-dimensional example with second order basis functions, which later will be used in the numerical results section in this thesis. Now we will consider the domain shown in figure 13. We will first show how this domain can be represented and refined in a FEM approach. Secondly, we will show how the same procedures can be executed in an IA approach with one patch. All the coordinates and weights can be viewed in the appendix.

FEM

Figure 14 shows the initial location of the nodes in a FEM approach starting with two elements. When we refine this mesh we have to make sure that every new corner node on the curved boundary actually lies at a distance one from the origin. This means that every time we want to refine the mesh, we have to communicate with the physical domain. The geometry of Ω is impossible to represent exact if we use the standard FEM approach. It should be mentioned that we could represent the arc exactly if we used blending functions Chen & Raju (2003) on the current boundary. We will use bisection to refine the mesh. In order to get more equally spaced elements, the first refinement is from two to four elements. Figure 15 shows the FEM representation of Ω after three additional refinement.

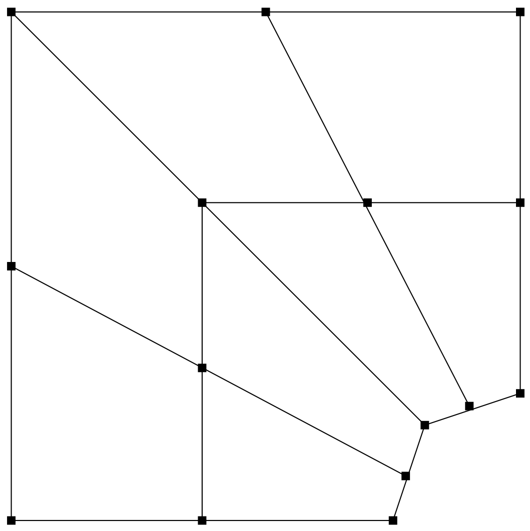


Figure 14: A FEM representation of Ω with two elements.

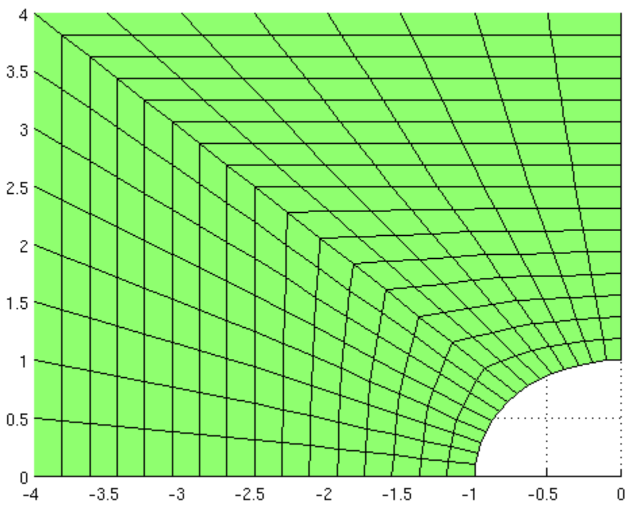


Figure 15: A FEM representation of Ω with 256 elements after four refinements.

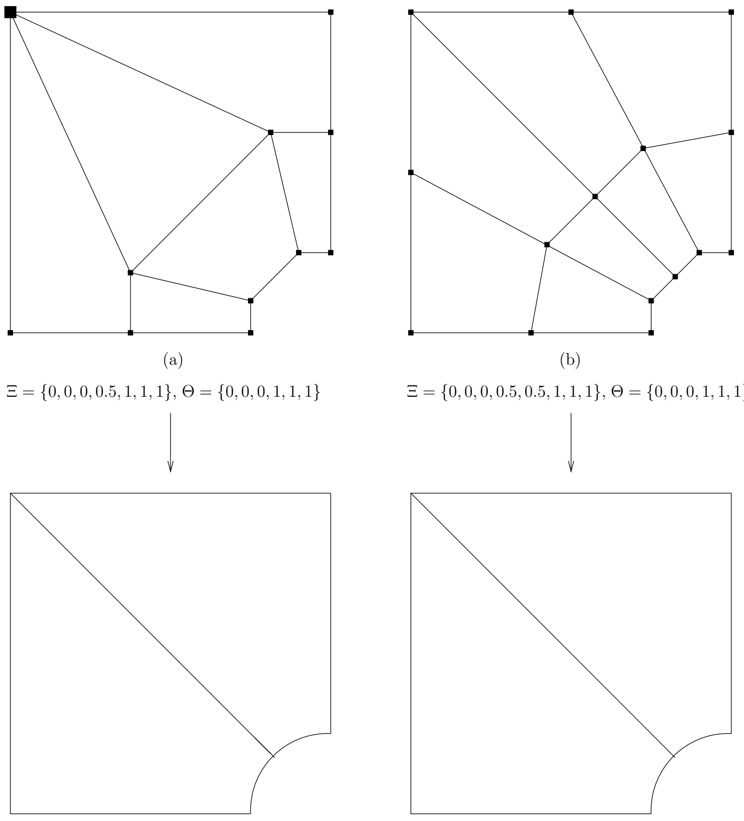

(a)
$\Xi = \{0,0,0,0.5,1,1,1\}$, $\Theta = \{0,0,0,1,1,1\}$
(b)
$\Xi = \{0,0,0,0.5,0.5,1,1,1\}$, $\Theta = \{0,0,0,1,1,1\}$

Figure 16: The two different representation of Ω in a IA approach with one patch and second order NURBS basis functions.

IA

There are primarily two different ways to precisely represent Ω when we use an IA approach with one patch and second order NURBS basis functions. We can either choose to place two control points in the same location in the upper left corner on the patch, or we can repeat a knot value in order to force the spline to interpolate in the same corner. These two approaches are shown in figure 16. After we have represented the geometry exactly, we never communicate with the physical domain under the refinement process, because the knot insertion algorithm ensures us that the geometry is preserved.

These two representations have some different properties. If we use bisection of the knot vectors when we refine, we will eventually get very thin and long elements in the upper left corner with the double control point approach. This can be seen in figure 17. This will affect the condition number of the stiffness matrix. In addition, if the error is not uniformly distributed, the convergence rate will suffer. On the other hand, if we use

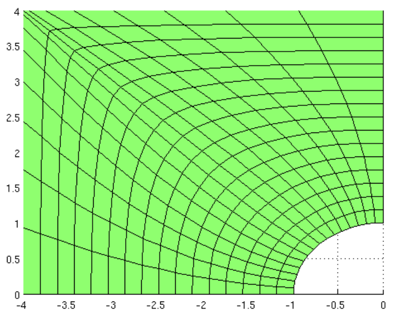
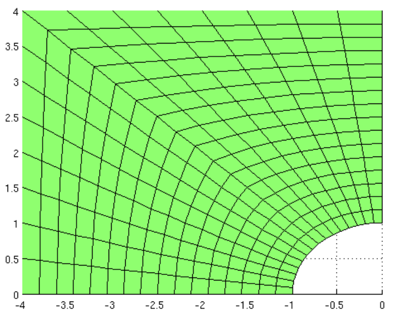



Figure 17: The two different representations of Ω with 256 elements after four refinements.

repeated knot values we lose continuity in the solution. And because we get different basis functions among the diagonal, this too will also affect the condition number of the stiffness matrix.

We can only use the knot insertion procedures on B-splines. This means that we have to convert the NURBS in $\mathbb{R}^2$ back to B-splines in $\mathbb{R}^3$ by the relation in equation 45. After we have converted the NURBS, we can refine the B-splines with the knot insertion procedure. As we did in the FEM approach we choose to use bisection, but now we bisect the knot vectors instead of the physical elements in the FEM approach. After we have refined the B-splines we use equation 45 once more and derive the new NURBS control points and weights. Equivalent with what we did in the FEM approach, we bisect Θ first in order to get more equally spaced elements. Figure 17 shows how the two representations look like after three additional refinements.

2.3.4 The stiffness- and mass matrix for second order basis functions

When we derive a set of algebraic equation from an ODE or a PDE, two of the most commonly derived matrices are the stiffness matrix, A, and the mass matrix ,M. In this section we aim to find out what these matrices will look like, and which qualities they possess in a FEM approach and in an IA approach. In order to find these matrices we will consider an ODE where both A and M occur as an example.

So far in this thesis we have only considered Lagrange functions as a basis in the FEM approach, but there exists other functions that could be used as a basis as well. In this section, and later in the numerical result section in this thesis, we will consider three types of FEM basis functions: Lagrange basis functions, *Bezier decomposition basis functions* (Cottrell et al. n.d.) and *Hierarchical basis functions* (Silvester & Ferrari 1996). In addition, we will consider the IA approach with a B-spline basis. We use a B-spline basis because the domain is a straight line and thus all the weights in the NURBS basis would be equal.

Consider the continuous eigenvalue problem

$$\begin{aligned} -u_{xx} &= \lambda u \quad , \text{ in } \Omega = (0,1) \\ u(0) = u(1) &= 0, \end{aligned} \tag{58}$$

with solutions

$$\begin{aligned} \overline{u}_l(x) &= \sin(l\pi x), \\ \overline{\lambda}_l &= l^2\pi^2, \end{aligned} \quad l = 1, 2, ..., \infty. \tag{59}$$

If we want to solve this problem with a FEM approach or an IA approach we get a discrete eigenvalue problem on the form

$$\underline{A}\underline{q}_l = \lambda_l \underline{M}\underline{q}_l, \tag{60}$$

where

$$\begin{aligned} A_{ij} &= a(\varphi_i, \varphi_j) = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j \, \mathrm{d}A \\ M_{ij} &= (\varphi_i, \varphi_j) = \int_\Omega \varphi_i \cdot \varphi_j \, \mathrm{d}A \end{aligned} \quad . \tag{61}$$

FEM

In order to use quadratic basis functions in the FEM approach, we have to include three nodes on each element. In the following example the nodes are placed equidistantly so that the element width equals h for all elements.

Lagrange basis functions

First we consider Lagrange basis functions. The basis functions and the reference basis

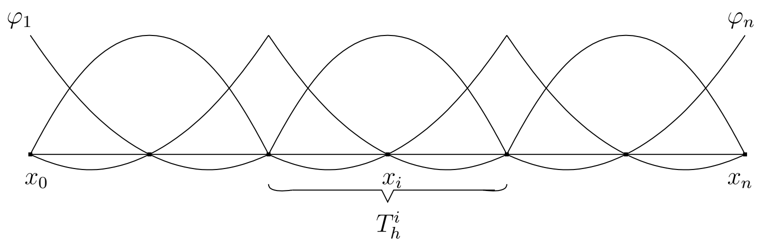


Figure 18: Lagrange nodal basis functions

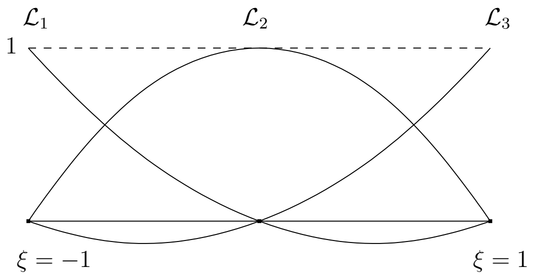


Figure 19: Lagrange reference basis functions

functions are shown in figure 18 and figure 19. We use the same procedures as we did in the FEM section in this thesis to derive $\underline{A}_L$ and $\underline{M}_L$.

The Lagrange reference basis functions and their derivatives are

$$\begin{array}{rclcrcl} \mathcal{L}_1(\zeta) & = & \frac{1}{2}\zeta(\zeta-1), & \frac{\mathcal{L}_1}{d\zeta} & = & \frac{1}{2}(2\zeta-1) \\ \mathcal{L}_2(\zeta) & = & (1-\zeta^2), & \frac{\mathcal{L}_2}{d\zeta} & = & -2\zeta \\ \mathcal{L}_3(\zeta) & = & \frac{1}{2}\zeta(\zeta+1), & \frac{\mathcal{L}_3}{d\zeta} & = & \frac{1}{2}(2\zeta+1) \end{array}. \tag{62}$$

The first matrix we want to derive is $\underline{A}_L$. The contribution from element k to $\underline{A}_L$ in terms of reference variables is

$$\int_{-1}^{1} (\frac{d\mathcal{L}_{1,2 \text{ or } 3}}{d\zeta}\frac{2}{h})(\frac{d\mathcal{L}_{1,2 \text{ or } 3}}{d\zeta}\frac{2}{h})(d\zeta\frac{h}{2}). \tag{63}$$

If we solve this integral for all possible combinations, we obtain the elemental stiffness matrix

$$\underline{A}_L^k = \frac{1}{3h}\begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix}. \tag{64}$$

If we assembly the contributions from all the elemental stiffness matrices, the global stiffness matrix is on the form

$$\underline{A}_L = \frac{1}{3h}\begin{pmatrix} 7 & -8 & 1 & & & & & & \\ -8 & 16 & -8 & & & & & & \\ 1 & -8 & 14 & -8 & 1 & & & & \\ & & -8 & 16 & -8 & & & & \\ & & 1 & -8 & 14 & -8 & 1 & & \\ & & & & & \ddots & & & \\ & & & & 1 & -8 & 14 & -8 & 1 \\ & & & & & & -8 & 16 & -8 \\ & & & & & & 1 & -8 & 7 \end{pmatrix}. \tag{65}$$

All other entries in the matrix is equal to zero. As expected because of the compact support of the basis functions, we get a sparse matrix. We notice that the band width is consistent with the support of the basis functions and alternate between 3 and 5. In addition, the matrix is SPD.

Now we want to derive the mass matrix. The contribution from element k to $\underline{M}_L$ in terms of reference variables is

$$\int_{-1}^{1} \mathcal{L}_{1,2 \text{ or } 3}\mathcal{L}_{1,2 \text{ or } 3}\, \mathrm{d}\zeta \frac{h}{2}. \tag{66}$$

If we solve this integral for all possible combinations, we obtain the elemental mass matrix

$$\underline{M}_L^k = \frac{h}{30}\begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix}. \tag{67}$$

If we assembly the contributions from all the elemental mass matrices, the global mass matrix is on the form

$$\underline{M}_L = \frac{h}{30}\begin{pmatrix} 4 & 2 & -1 & & & & & & \\ 2 & 16 & 2 & & & & & & \\ -1 & 2 & 8 & 2 & -1 & & & & \\ & & 2 & 16 & 2 & & & & \\ & & -1 & 2 & 8 & 2 & -1 & & \\ & & & & & \ddots & & & \\ & & & & -1 & 2 & 8 & 2 & -1 \\ & & & & & & 2 & 16 & 2 \\ & & & & & & -1 & 2 & 4 \end{pmatrix}. \tag{68}$$

We notice that some of the entries in $\underline{M}_L$ is negative. This results from of the fact that about half of the basis functions takes on both positive and negative values. In addition, the sum of all the entries in $\underline{M}_L$ is equal to one. This can be shown with the following

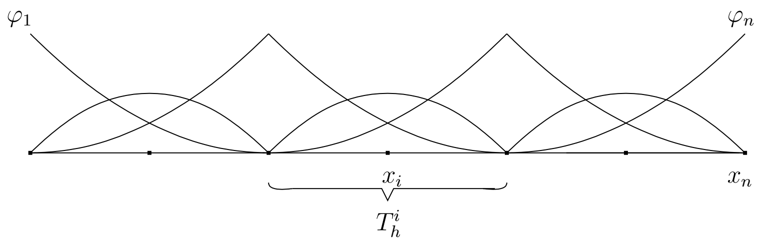


Figure 20: Bezier decompositon basis functions.

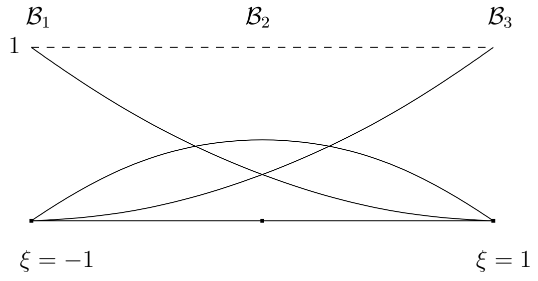


Figure 21: Bezier decompositon reference basis functions.

equation, and is correct because the Lagrange basis functions constitutes a partition of unity.

$$\sum_{i=0}^{n}\sum_{j=0}^{n}\underline{M}_{ij} = \int_0^1 \sum_{i=0}^{n}\sum_{j=0}^{n}\varphi_i\varphi_j \,\mathrm{d}x = \int_0^1 \sum_{i=0}^{n}\varphi_i \sum_{j=0}^{n}\varphi_j \,\mathrm{d}x = \int_0^1 1\cdot 1 \,\mathrm{d}x = 1 \tag{69}$$

Bezier decomposition basis functions

Next we consider Bezier decomposition basis functions. From now on we will refer to them as Bezier basis functions. We observe from figure 20 that the Bezier basis functions are positive for all values of x, and that only about half of the basis functions interpolate the nodal points. At a first glance, it can look like the Bezier basis functions are a hybrid of the Lagrange basis functions and the B-spline basis functions. In fact Bezier basis functions are C^{-1}-continous B-spline basis functions with repeated knot values and Cottrell et al. (n.d.) show that Bezier basis functions can play a major part in an IA approach. When we use a Bezier basis in a FEM approach, we consider the Bezier basis functions as polynomials in the same manner as for the Lagrange basis functions, and not as spline curves. The basis functions and the reference basis functions are shown in figure 20 and figure 21.

The Bezier reference basis functions and their derivatives are

$$\begin{array}{rclrcl} \mathcal{B}_1(\zeta) & = & \frac{1}{4}(1-\zeta)^2, & \frac{\mathcal{B}_1}{d\zeta} & = & \frac{1}{2}(\zeta-1) \\ \mathcal{B}_2(\zeta) & = & \frac{1}{2}(1-\zeta^2), & \frac{\mathcal{B}_2}{d\zeta} & = & -\zeta \\ \mathcal{B}_3(\zeta) & = & \frac{1}{4}(1+\zeta)^2, & \frac{\mathcal{B}_3}{d\zeta} & = & \frac{1}{2}(\zeta+1) \end{array} . \tag{70}$$

We use the same procedure here as for the Lagrange basis functions and derive the stiffness matrix

$$\underline{A}_B = \frac{1}{3h}\begin{pmatrix} 4 & -2 & -2 & & & & & & \\ -2 & 4 & -2 & & & & & & \\ -2 & -2 & 8 & -2 & -2 & & & & \\ & & -2 & 4 & -2 & & & & \\ & & -2 & -2 & -8 & -2 & -2 & & \\ & & & & & \ddots & & & \\ & & & & -2 & -2 & 8 & -2 & -2 \\ & & & & & & -2 & 4 & -2 \\ & & & & & & -2 & -2 & 4 \end{pmatrix} \tag{71}$$

and the mass matrix

$$\underline{M}_B = \frac{h}{30}\begin{pmatrix} 6 & 3 & 1 & & & & & & \\ 3 & 4 & 3 & & & & & & \\ 1 & 3 & 12 & 3 & 1 & & & & \\ & & 3 & 4 & 3 & & & & \\ & & 1 & 3 & 12 & 3 & 1 & & \\ & & & & & \ddots & & & \\ & & & & 1 & 3 & 12 & 3 & 1 \\ & & & & & & 3 & 4 & 3 \\ & & & & & & 1 & 3 & 6 \end{pmatrix} . \tag{72}$$

In the same manner as for the Lagrange basis functions, the stiffness matrix is SPD and the band width of $\underline{A}_B$ is consistent with the support of the basis functions. Because the Bezier basis functions is positive for all x, all the entries in $\underline{M}_B$ are positive. In addition, because Bezier basis functions also constitute a partition of unity, $\underline{M}_B$ sums to one.

Hierarchical basis functions

Finally in the FEM approach, we consider hierarchical basis functions. Hierarchical basis functions are constructed from Lagrange basis functions. The hierarchical basis functions we consider are constructed from linear and quadratic Lagrange basis functions as shown in figure 22. Because the Hierarchical basis functions are constructed from Lagrange basis functions, they interpolate all the nodal points, but they do not constitute a partition of unity. In addition, unlike the second order Lagrange basis functions they are positive

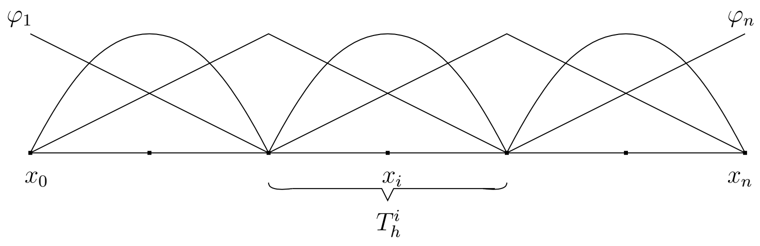


Figure 22: Hierachical basis functions.

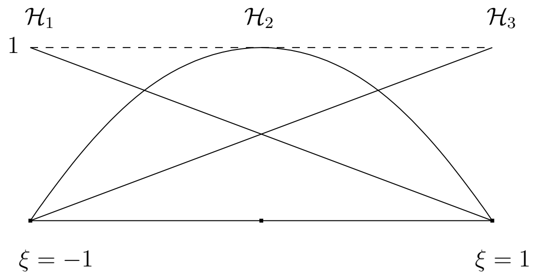


Figure 23: Hierachical reference basis functions.

for all x. The basis functions and the reference basis functions are shown in figure 22 and figure 23.
The hierarchical reference basis functions and their derivatives are

$$\begin{array}{rclrcl} \mathcal{H}_1(\zeta) & = & \frac{1}{2}(1-\zeta), & \frac{\mathcal{H}_1}{d\zeta} & = & -\frac{1}{2} \\ \mathcal{H}_2(\zeta) & = & (1-\zeta^2), & \frac{\mathcal{H}_2}{d\zeta} & = & -2\zeta \;. \\ \mathcal{H}_3(\zeta) & = & \frac{1}{2}(1+\zeta), & \frac{\mathcal{H}_3}{d\zeta} & = & \frac{1}{2} \end{array} \tag{73}$$

We use the same procedure as before and derive the stiffness matrix

$$\underline{A}_H = \frac{1}{3h}\begin{pmatrix} 3 & 0 & -3 & & & & & & & \\ 0 & 16 & 0 & & & & & & & \\ -3 & 0 & 6 & 0 & -3 & & & & & \\ & & 0 & 16 & 0 & & & & & \\ & & -3 & 0 & 6 & 0 & -3 & & & \\ & & & & & \ddots & & & & \\ & & & & -3 & 0 & 6 & 0 & -3 \\ & & & & & & 0 & 16 & 0 \\ & & & & & & -3 & 0 & 3 \end{pmatrix} \tag{74}$$

and the mass matrix

$$\underline{M}_H = \frac{h}{30}\begin{pmatrix} 10 & 10 & 5 & & & & & & \\ 10 & 16 & 10 & & & & & & \\ 5 & 10 & 20 & 10 & 5 & & & & \\ & & 10 & 16 & 10 & & & & \\ & & 5 & 10 & 20 & 10 & 5 & & \\ & & & & & \ddots & & & \\ & & & & 5 & 10 & 20 & 10 & 5 \\ & & & & & & 10 & 16 & 10 \\ & & & & & & 5 & 10 & 10 \end{pmatrix} \tag{75}$$

In the same manner as for the two previous FEM basis functions, the stiffness matrix is SPD, but unlike $\underline{A}_L$ and $\underline{A}_B$ the band width of $\underline{A}_H$ is not consistent with the support of the basis functions. As expected, all entries in $\underline{M}_H$ are positive, but the sum of all entries does not equal one, since the hierarchical basis functions do not constitute a partition of unity.

Mapping between the FEM basis functions

If we define three new vectors as

$$\begin{array}{rcl} \underline{\mathcal{L}} & = & [\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3]^T \\ \underline{\mathcal{B}} & = & [\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3]^T \\ \underline{\mathcal{H}} & = & [\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3]^T \end{array}, \tag{76}$$

there will exists matrices $\underline{C}_{xy}$ such that for every $\underline{x}, \underline{y} \in (\underline{\mathcal{L}}, \underline{\mathcal{B}}, \underline{\mathcal{H}})$

$$\underline{C}_{xy}\underline{x} = \underline{y}. \tag{77}$$

As an example consider the matrix

$$\underline{C}_{BL} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ 0 & 2 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}. \tag{78}$$

$$\underline{C}_{BL}\underline{\mathcal{B}} = \begin{bmatrix} \mathcal{B}_1 & -\frac{1}{2}\mathcal{B}_2 & \\ & 2\mathcal{B}_2 & \\ & -\frac{1}{2}\mathcal{B}_2 & +\mathcal{B}_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(\xi^2 - \xi) \\ 1 - \xi^2 \\ \frac{1}{2}(\xi^2 + \xi) \end{bmatrix} = \begin{bmatrix} \mathcal{L}_1 \\ \mathcal{L}_2 \\ \mathcal{L}_3 \end{bmatrix} = \underline{\mathcal{L}} \tag{79}$$

Because of the tensor product structure this can be extended to higher dimensions as well. This can be illustrated with the two-dimensional example

$$\begin{array}{rcl} \mathcal{L}_i(\xi)\mathcal{L}_j(\eta) & = & (c_{i1}\mathcal{H}_1(\xi) + c_{i2}\mathcal{H}_2(\xi) + c_{i3}\mathcal{H}_3(\xi))(c_{j1}\mathcal{H}_1(\eta) + c_{j2}\mathcal{H}_2(\eta) + c_{j3}\mathcal{H}_3(\eta)) \\ & = & d_{ij1}\mathcal{H}_1(\xi)\mathcal{H}_1(\eta) + d_{ij2}\mathcal{H}_1(\xi)\mathcal{H}_2(\eta) + \ldots + d_{ij9}\mathcal{H}_3(\xi)\mathcal{H}_3(\eta). \end{array} \tag{80}$$

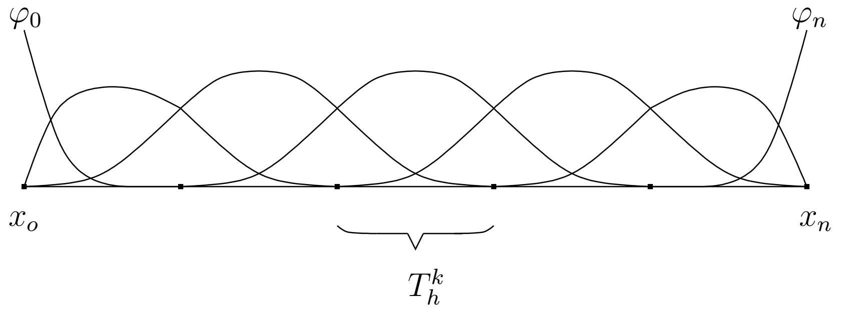


Figure 24: Quadratic B-spline basis functions.

Because of the linear mapping between the basis functions, all of the basis functions have to span the same linear space. Even though we can use different basis functions in the FEM approach, the elements and the location on the DOFs will be equal. Because of this equality and the fact that the three different basis functions all span the same linear space, we expect the different basis functions to give the same solution for problems with homogeneous Dirichlet boundary conditions.

IA

In the same manner as for the FEM approach, we get three active basis functions on each element in the IA approach if we use basis functions of degree two. In the following example the knot vector is open, the knots are uniformly distributed and the width of each element is equal to h. In this one-dimensional example we can use the physical space as the parameterization. This means that the knot vector contains the knots $\Xi = [x_0, x_0, x_0, x_1, x_2, \ldots, x_{n-1}, x_{n-1}, x_n, x_n, x_n]$. Figure 24 shows the elements and the basis functions. Like we did in the FEM approach, we will find the contribution from each element and assembly the contributions in the global matrices. Because the first and last knot is repeated $p+1$ times, we expect different contributions from the elements that connects with the boundary and the elements that do not. This is consistent with figure 24, where we observe the different nature of the basis functions. We will first find the contribution from the elements on the interior, and secondly find the contributions from the elements on the boundary.

Figure 25 shows the reference basis functions. In order to find the equations to the basis functions, we use the recursive relation from equation 37. If we use this relation, we can express every basis function as a sum of three parts. On a element each of these three parts will behave as one of the basis functions.

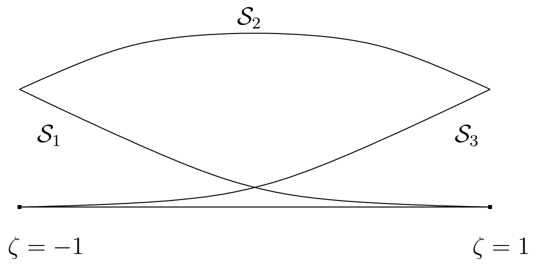


Figure 25: The reference basis functions.

$$\begin{aligned}
N_{i,2}(x|x_i, x_{i+1}, x_{i+2}, x_{i+3}) &= \frac{(x-x_i)^2}{(x_{i+2}-x_i)(x_{i+1}-x_i)} N_{i,0}(x|x_i, x_{i+1}) \\
&+ \left(\frac{(x-x_i)(x_{i+2}-x)}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} + \frac{(x_{i+3}-x)(x-x_{i+1})}{(x_{i+3}-x_{i+1})(x_{i+2}-x_{i+1})}\right) N_{i+1,0}(x|x_{i+1}, x_{i+2}) \\
&+ \frac{(x_{i+3}-x)^2}{(x_{i+3}-x_{i+1})(x_{i+3}-x_{i+2})} N_{i+2,0}(x|x_{i+2}, x_{i+3})
\end{aligned} \tag{81}$$

If we use this relation, the reference basis functions and their derivatives are can be stated as

$$\begin{aligned}
\mathcal{S}_1(\zeta) &= \frac{(1-\zeta)^2}{8}, & \frac{\mathcal{S}_1}{d\zeta} &= \frac{\zeta-1}{4} \\
\mathcal{S}_2(\zeta) &= \frac{(\zeta+3)(1-\zeta)}{8} + \frac{(3-\zeta)(\zeta+1)}{8}, & \frac{\mathcal{S}_2}{d\zeta} &= -\frac{\zeta}{2} \\
\mathcal{S}_3(\zeta) &= \frac{(\zeta+1)^2}{8}, & \frac{\mathcal{S}_3}{d\zeta} &= \frac{\zeta+1}{4}
\end{aligned} . \tag{82}$$

We proceed as we did for the FEM basis functions and derive the elemental stiffness matrix

$$\underline{A}_S^k = \frac{1}{6h} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \tag{83}$$

and the elemental mass matrix

$$\underline{M}_S^k = \frac{h}{120} \begin{bmatrix} 6 & 13 & 1 \\ 13 & 54 & 13 \\ 1 & 13 & 6 \end{bmatrix} . \tag{84}$$

The next thing we want to find is the contribution from the elements that are connected to the boundary. Figure 26 shows the reference basis functions at the left end of the patch. If we use the same recursive relation in equation 37, we can find the basis functions for the two elements on the boundary. It is sufficient to find the contribution from one of these elements because of the symmetry. We chose to consider the element at the left end of the patch where the reference basis functions and their derivatives are

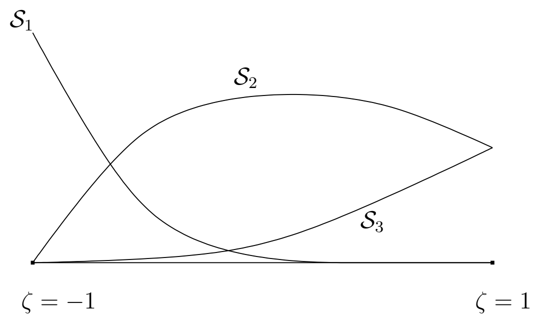


Figure 26: The reference basis functions at the left end of the patch.

$$\begin{array}{rclrcl} \mathcal{S}_1(\zeta) & = & \frac{(1-\zeta)^2}{4}, & \frac{\mathcal{S}_1}{d\zeta} & = & \frac{\zeta-1}{2} \\ \mathcal{S}_2(\zeta) & = & \frac{(\zeta+1)(1-\zeta)}{4} + \frac{(3-\zeta)(\zeta+1)}{8}, & \frac{\mathcal{S}_2}{d\zeta} & = & \frac{2-6\zeta}{8} \\ \mathcal{S}_3(\zeta) & = & \frac{(\zeta+1)^2}{8}, & \frac{\mathcal{S}_3}{d\zeta} & = & \frac{\zeta+1}{4} \end{array} . \tag{85}$$

We proceed as before and derive the elemental stiffness matrix

$$\underline{A}_S^1 = \frac{1}{6h} \begin{bmatrix} 8 & -6 & -2 \\ -6 & 6 & 0 \\ -2 & 0 & 2 \end{bmatrix} \tag{86}$$

and the elemental mass matrix

$$\underline{M}_S^1 = \frac{h}{120} \begin{bmatrix} 24 & 14 & 2 \\ 14 & 34 & 12 \\ 2 & 12 & 6 \end{bmatrix} . \tag{87}$$

If we assembly the contributions from all the elements we finally derive the stiffness matrix

$$\underline{A}_S = \frac{1}{6h} \begin{pmatrix} 8 & -6 & -2 & & & & & & & \\ -6 & 8 & -1 & -1 & & & & & & \\ -2 & -1 & 6 & -2 & -1 & & & & & \\ & -1 & -2 & 6 & -2 & -1 & & & & \\ & & -1 & -2 & 6 & -2 & -1 & & & \\ & & & & & \ddots & & & & \\ & & & & -1 & -2 & 6 & -1 & -2 \\ & & & & & -1 & -1 & 8 & -6 \\ & & & & & & -2 & -6 & 8 \end{pmatrix} \tag{88}$$

and the mass matrix

$$\underline{M}_S = \frac{h}{120}\begin{pmatrix} 24 & 14 & 2 & & & & & & \\ 14 & 40 & 25 & 1 & & & & & \\ 2 & 25 & 66 & 26 & 1 & & & & \\ & 1 & 26 & 66 & 26 & 1 & & & \\ & & 1 & 26 & 66 & 26 & 1 & & \\ & & & & & \ddots & & & \\ & & & & 1 & 26 & 66 & 25 & 2 \\ & & & & & 1 & 25 & 40 & 14 \\ & & & & & & 2 & 14 & 24 \end{pmatrix}. \tag{89}$$

As in the FEM approach, the IA matrices are sparse and SPD. In addition, the band width is consistent with the support of the basis functions. All the entries in $\underline{M}_S$ are positive because of the nature of the B-spline basis functions, and the sum of all the entries in M are equal to one because the basis functions constitute a partition of unity. All these properties are shared with the matrices derived from the Bezier basis functions. However, there is one aspect with $\underline{A}_S$ and $\underline{M}_S$ that are not found in any of the matrices in the FEM approach. Because all of the B-spline basis functions, except the ones near the boundary, are equal, the band width of $\underline{A}_S$ and $\underline{M}_S$ does not alternate. In addition, all the rows and columns in the matrices, except the first and last few, are equal. This is a vital difference between the FEM and IA approach, in the advantage of the IA approach as we will see next.

Numerical results of the eigenvalue problem

So far in this sections have we derived the matrices that are needed in order to solve the one-dimensional eigenvalue problem. Now we will show what the numerical solutions look like. Figure 27 shows the numerical solutions, where the normalized eigenvalues are plotted versus the fraction of the total DOFs. As we expected, all the different FEM basis functions give the same solution to this problem. As we observe from figure 27, the IA approach gives a better approximation for the whole specter of eigenvalues. In addition, the FEM approach depict a so called *acoustical branch* for $n/N < 0.5$ and a *optical branch* for $n/N > 0.5$. This is due to the fact that there are two different types of basis functions in the FEM approach. This branching does not occur with IA basis functions because all the basis functions are equal, except a small number near the boundary. Cottrell et al. (n.d.) gives a more thorough description of this phenomena.

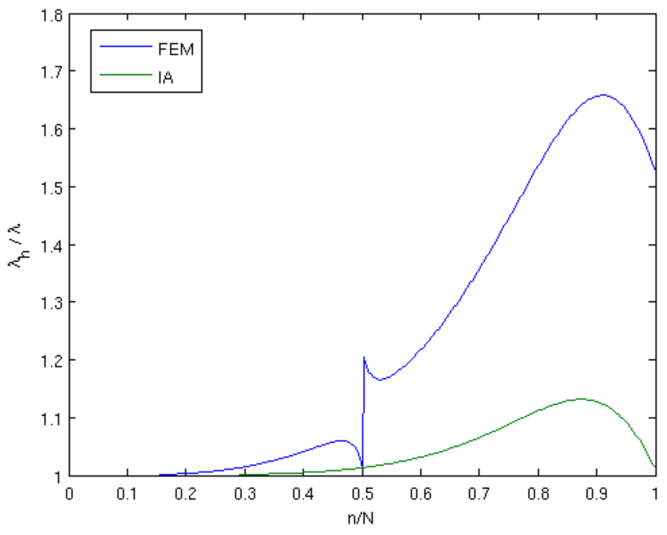


Figure 27: Solution of the eigenvalue problem with both a FEM approach and an IA approach. The normalized eigenvalues results versus the fraction of the total DOFs

3 Implementation

In this section we will show how we have implemented the FEM approach and IA approach with second order basis functions in order to solve a two dimensional Poisson problem with homogeneous Dirichlet boundary conditions. In general, the IA approach will be more complicated to implement because of the parametric space, the knot insertion procedure, the weights and the recursive nature of the basis functions. On the other hand, because of these features, the implementation it is more general and more easily adaptable to different geometries and degrees on the basis functions. Since one of the main goals in this thesis is to compare the solution times for the two different approaches, we have tried to make the implementations as similar as possible. The pseudocode in this section will show the disciplinarian procedures for the two implementations. All implementation have been executed in the MATLAB.

The implementation is separated into three parts for both the FEM approach and the IA approach: preprocessing, processing and postprocessing. The idea behind this implementation approach, is that in order to solve different problems, only changes in some of the parts have to be done. As typical for different problems, only changes in the preprocessing part have to be made.

3.1 Preprocessing

In the preprocessing part of the implementation, we define parameters and relations that are specific for a given problem. In addition, we include the refinement process. This is because whilst refinement in the IA approach is very general, refinement in the FEM approach often involves communication with the original domain.

The finite element method

Input: $gridtype, NGAUSS, Nref$
Which grid that should be used in the calculations ($gridtype$), the number of Gauss points to be used in each spatial direction on each element ($NGAUSS$) and how many bisections of the mesh we want to make ($Nref$).

Output: $INN, IEN, n2f, p, NEL, DOF, gp, gw$
A matrix with relations beween global and local indices (INN), a matrix with global indices for each element (IEN), the node to freedom matrix ($n2f$), a matrix containing the nodal coordinates (p), the total number of elements (NEL), the total number of degrees of freedom (DOF) and the Gaussian points and weights (gp, gw) in terms of reference variables.

The isogeometric analysis

Algorithm 1 Preprocessing: FEM

```
$p \leftarrow$ load node coordinates
for $i = 1$ to $Nref$ do
    $p \leftarrow p$ refined
end for
$np \leftarrow$ number of nodes in $p$
$n2f \leftarrow$ zeros$(2, np)$
for $i = 1$ to $np$ do
    $n2f(1, i) \leftarrow i$
    if node $i$ on the boundry then
        $n2f(2, i) \leftarrow -1$
    end if
end for
$INN \leftarrow$ define node topology
$IEN \leftarrow$ define element node topology
$xN, yN \leftarrow$ total number of nodes in the x- and y-direction
$NEL \leftarrow$ total number of elements
$DOF \leftarrow$ total number of degrees of freedom
$gp, gw \leftarrow$ Gauss points and weights
```

Input: $gridtype, NGAUSS, Nref$
Which grid that should be used in the calculations ($gridtype$), the number of Gauss points to be used in each spatial direction on each element ($NGAUSS$) and how many bisection of the knot vector we want to make ($Nref$).

Output: $INN, IEN, n2f, BNET, ttabx, ttaby, NEL, DOF, gp, gw$
A matrix with relations beween global and local indices (INN), a matrix with global indices for each element (IEN), a node to freedom matrix ($n2f$), a matrix containing the coordinates and weights of the control points ($BNET$), an open knot vector in each spatial direction ($ttabx, ttaby$), the total number of elements (NEL), the total number of degrees of freedom (DOF), the Gauss points (gp) and the Gauss weights (gw) in terms of reference variables.

If we compare the preprocessing pseudocodes we observe that from a disciplinarian point of view, the pseudocodes are very similar. In both cases we need to know how many elements there are and how many DOFs there are. In addition, we need to know the coordinates of the nodal points/control points, and we need to define local to global relations. However, as we have discussed earlier in this thesis, there are several differences between the two approaches concerning both the refinement process and in the distribution of the nodal points/control points. Finally, in the IA implementation we need to construct knot vectors and store the weights of the control point. This has no analogue in the FEM implementation.

Algorithm 2 Preprocessing: IA

```
$BNET \leftarrow$ load control points coordinates and weights
$ttabx, ttaby \leftarrow$ define open knot vectors in x- and y direction
for $i = 1$ to $Nref$ do
    $p, ttabx, ttaby \leftarrow p, ttabx, ttaby$ refined
end for
$np \leftarrow$ number of nodes in $p$
$n2f \leftarrow$ zeros$(2, np)$
for $i = 1$ to $np$ do
    $n2f(1, i) \leftarrow i$
    if node $i$ on the boundry then
        $n2f(2, i) \leftarrow -1$
    end if
end for
$INN \leftarrow$ define node topology
$IEN \leftarrow$ define element node topology
$NEL \leftarrow$ total number of elements
$DOF \leftarrow$ total number of degrees of freedom
$gp, gw \leftarrow$ Gauss points and weights
```

3.2 Processing

In the processing part of the implementation, we construct and solve the system of algebraic equation $\underline{A}\ \underline{u}_h = \underline{b}$. We construct the algebraic system in the same manner as we have discussed earlier in this thesis by assembly the contribution from each element. In order to solve the algebraic system we will use the iterative CG.

The finite element method

Input: $INN, IEN, n2f, p, NEL, DOF, gp, gw$

Output: $u_h, t_{assembly}, t_{CG}$
The solution vector (u_h), the assembly time ($t_{assembly}$) and the time it takes to solve the algebraic system using CG (t_{CG}).

The isogeometric analysis

Input: $INN, IEN, n2f, BNET, ttabx, ttaby, NEL, DOF, gp, gw$

Output: $u_h, t_{assembly}, t_{CG}$

The course in the two pseudocode are almost identical. In fact, in the IA pseudocode we have only included the assembly procedure, because the rest is equal to the FEM pseudocode. The reason that the pseudocodes are so much alike, is of course the fact that the two processing approaches in essence do the same. Both approaches loop over all the elements in order to find the elemental stiffness matrices and load vectors, and use local to global relationships in order to construct $\underline{A}$ and $\underline{b}$. In concern of the computational time, the main difference between the FEM processing and the IA processing is the fact that the IA processing will be more time consuming. Because of their weights and recursive nature, NURBS and their derivatives are more time consuming to calculate then the FEM basis functions and their derivatives. In addition, we need to calculate two Jacobi matrices in the IA processing, compared with the one in the FEM processing. However, if we build a solid framework of methods for computing the values of the NURBS basis functions and their derivatives, we can easily find the values of higher order NURBS basis functions and their derivatives because of the recurrence relations. This is not possible in our FEM processing.

Algorithm 3 Processing: FEM

```
$A_h \leftarrow$ zeros($DOF, DOF$), the global stiffness matrix
$F_h \leftarrow$ zeros($DOF, 1$), the global load vector
$ng \leftarrow$ number of Gauss points
$tic \leftarrow$ start time
**for** $k = 1$ to $NEL$ **do**
  $A_k \leftarrow$ zeros($9, 9$), the elemental stiffness matrix
  $F_k \leftarrow$ zeros($9, 1$), the elemental load vector
  **for** $i = 1$ to $ng$ **do**
    **for** $j = 1$ to $ng$ **do**
      $x, y, Jmat, P, DP \leftarrow$ the real coordinates, Jacobi matrix, local basis functions
      and their derivatives calculated in the quadrature point $(gp(i), gp(j))$
      $J \leftarrow$ the determinant of $Jmat$
      $Jinv \leftarrow$ the inverse of $Jmat$
      $G \leftarrow$ the transpose of $Jinv$
      **for** $m = 1$ to $9$ **do**
        **if** local basis function $m$ active **then**
          $F_k(m) \leftarrow F_k(m) + gw(i) \cdot gw(j) \cdot f(x, y) \cdot P(m) \cdot J$
          **for** $n = 1$ to $9$ **do**
            **if** local basis function $n$ active **then**
              $A_k(m, n) \leftarrow A_k(m, n) + gw(i) \cdot gw(j) \cdot (G \cdot DP(m))^T \cdot (G \cdot DP(n)) \cdot J$
            **end if**
          **end for**
        **end if**
      **end for**
    **end for**
  **end for**
  $A_h \leftarrow A_h +$ the contribution from $A_k$
  $F_h \leftarrow F_h +$ the contribution from $F_k$
**end for**
$t_{assembly} \leftarrow toc$, end time
$tic \leftarrow$ start time
$u_h \leftarrow CG(A_h, F_h)$
$t_{CG} \leftarrow toc$, end time
```

Algorithm 4 Processing: IA

```
for $k = 1$ to $NEL$ do
  if area of element $k \neq 0$ then
    $A_k \leftarrow$ zeros$(9,9)$, the elemental stiffness matrix
    $F_k \leftarrow$ zeros$(9,1)$, the elemental load vector
    for $i = 1$ to $ng$ do
      for $j = 1$ to $ng$ do
        $x, y, Jmat1, Jmat2, N, DN \leftarrow$ the real coordinates, Jacobi matrix between physical and parametric domain, Jacobi matrix between parametrical domain and parent domain, local NURBS basis functions and their derivatives calculated in the quadrature point $(gp(i), gp(j))$
        $Jmat \leftarrow Jmat1 \cdot Jmat2$
        $J \leftarrow$ the determinant of $Jmat$
        $Jinvtrans1 \leftarrow$ the transpose of the inverse of $Jmat1$
        $Jinv2 \leftarrow$ the inverse of $Jmat2$
        $G \leftarrow Jinvtrans1 \cdot Jinv2$
        for $m = 1$ to $9$ do
          if local basis function $m$ active then
            $F_k(m) \leftarrow F_k(m) + gw(i) \cdot gw(j) \cdot f(x,y) \cdot N(m) \cdot J$
            for $n = 1$ to $9$ do
              if local basis function $n$ active then
                $A_k(m,n) \leftarrow A_k(m,n) + gw(i) \cdot gw(j) \cdot (G \cdot DN(m))^T \cdot (G \cdot DN(n)) \cdot J$
              end if
            end for
          end if
        end for
      end for
    end for
    $A_h \leftarrow A_h +$ the contribution from $A_k$
    $F_h \leftarrow F_h +$ the contribution from $F_k$
  end if
end for
```

Algorithm 5 Post processing: FEM and IA

```
$enererr \leftarrow 0$
$ng \leftarrow$ number of Gauss points
$Xval \leftarrow$ zeros$(NEL \cdot ng, ng)$, the x-coordinates
$Yval \leftarrow$ zeros$(NEL \cdot ng, ng)$, the y-coordinates
$Zval \leftarrow$ zeros$(NEL \cdot ng, ng)$, the z-coordinates
for $k = 1$ to $NEL$ do
  $enererr \leftarrow enererr +$ the contribution from element $k$
  $Xval \leftarrow$ the x-coordinates in element $k$
  $Yval \leftarrow$ the y-coordinates in element $k$
  $Zval \leftarrow$ the z-coordinates in element $k$
end for
$relerr \leftarrow (enererr/anaerr)^{\frac{1}{2}}$
plot$(X, Y, Z)$
```

3.3 Postprocessing

In the postprocessing part of the implementation we want to find the relative error of the numerical solution measured in the energy norm. We assume that we know the analytic solution to the problem. In addition, we will plot the solution. In order to obtain the relative error, we loop over all the elements and find the error contribution from each element. The post processing pseudocode will be almost identical for both the FEM approach and the IA approach, and in addition, very similar to the pseudocode in the processing section. Because of these similarities, we will only show one pseudocode which is applicable for both approaches.

When we enumerate all the parts in the implementation, it seems evident that the IA approach is more difficult to implement, but this approach has many advantages. The most important advantage in the IA approach compared to the FEM approach is the generality and robustness. If we have implemented a solid framework of methods, we can easily change the degree of the basis functions and the refinement scheme. In addition, we do not have to communicate with the physical domain when we refine. If we want to change the order on the basis functions the FEM approach, we have to define the new basis functions and construct a new or adapted mesh, and that can be both difficult and time consuming. However, the assembling procedure in the FEM approach is faster then the assembling procedure in the IA approach.

4 Numerical Results

So far in this thesis we have considered differences and similarities between the FEM approach and IA approach from a theoretical and implementational point of view. In this section we will solve two Poisson problems on different domains using both a FEM approach and an IA approach. In the FEM approach, we will use the three different basis functions that were discussed in the theory section in this thesis. We will use basis functions of degree two in both approaches. We aim to find out how the condition number of the stiffness matrix and the relative error propagates with respect to the total amount of DOFs. In addition, we aim to find out how the relative error propagates with respect to the total computational time (assemble time and iteration time). We know the analytic solutions to both Poisson problem and can thus measure the numerical error in the relative energy norm

$$e = \sqrt{\frac{a(u-u_h, u-u_h)}{a(u,u)}}. \tag{90}$$

On the first domain we will only consider a one-patch representation of the physical domain, while on the second domain we consider both a one-patch and a two-patch representation. We will use numerical integration with three Gauss Legendre quadrature points in each spatial direction on the elements in order to derive the algebraic system. Hughes, Reali & Sangalli (2008) show that in an IA approach the number of quadrature points needed will be roughly half the number of of DOFs.

In order to solve the algebraic system, we use CG with a tolerance limit small enough for the iteration error to be neglectable compared with the relative error between the numerical and analytical solution. The total amount of iterations required in order for CG to converge depends on the condition number of the stiffness matrix. For the Poisson problem, the condition number will depend on the element size squared (Rønquist 2008*b*), and thus higher dimensions will produce better condition numbers compared to the number of degrees of freedom.

As described earlier, the Poisson problem is on the form

Strong form

Domain: $\Omega = (-1,1) \times (-1,1)$.
Find u such that

$$\begin{array}{rcll} -\Delta u & = & f & \text{in } \Omega \\ u(x,y) & = & 0 & \text{on } \partial\Omega \end{array} \tag{91}$$

for a given f.

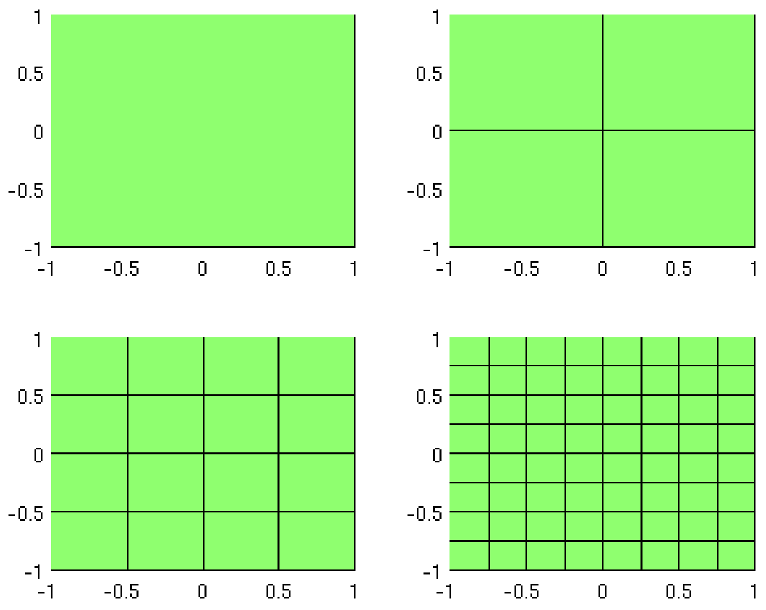


Figure 28: Ω represented with 1, 4, 16 and 64 elements.

4.1 Square domain

First we consider a Poisson problem with homogeneous Dirichlet boundary conditions on the square domain $\Omega = (-1, 1) \times (-1, 1)$. On this domain the location of the DOFs will differ in the FEM approach and the IA approach, but the element size and location will be the same for both approaches. Figure 28 shows Ω and the three first bisection refinements. On this square domain all the weights in the IA approach will be equal and we can use B-spline basis functions instead of NURBS. If we use B-spline basis functions we expect the computational time to be less compared to NURBS basis functions with different weights.

4.1.1 u is a C^∞ function

The first problem we will consider has the analytic solution

$$u(x, y) = \sin(\pi(x^2 - 1)(y^2 - 1)). \tag{92}$$

Figure 29 shows the analytic solution of the problem. When we use this $u(x, y)$, the right hand side of Poisson problem becomes

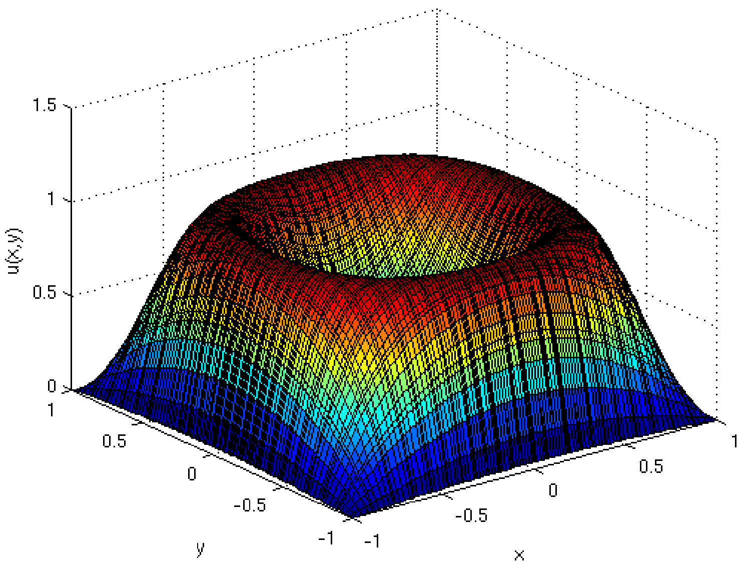


Figure 29: The analytic solution.

$$\begin{aligned} f &= (4x^2(y^2-1)^2 + 4y^2(x^2-1)^2)\sin(\pi(x^2-1)(y^2-1)) \\ &\quad -(2\pi(x^2-1) + 2\pi(y^2-1))\cos(\pi(x^2-1)(y^2-1)) \end{aligned} . \tag{93}$$

Figure 30 shows how the condition number of the stiffness matrices propagates with respect to the total number of degrees of freedom. As expected, because the elements are bisected, we observe a linear growth of the condition number in all four cases when we use a log-log plot. Even though the growth rate is the same in all four cases for $DOF \gg 1$, the condition number differs. For equally sized elements it looks like that Lagrange basis functions gives the highest condition number. Bezier basis functions and Hierarchical basis functions appear to result in approximately the same condition number, a bit lower then the condition number derived from the Lagrange basis functions. And finally, B-spline basis functions result in the lowest condition number. If we consider the growth of the condition number for the NURBS basis functions in figure 30, we observe a very slow growth for small number of DOFs. This is probably due to the fact that NURBS basis functions near the boundary differ from the ones in the interior of the domain, and for small numbers of DOFs the fraction of boundary basis function are high.

In figure 31 the relative error with respect to the number of DOFs is displayed. As discussed in the theory section in this thesis, since we have homogeneous Dirichlet boundary conditions, we expect the error to be equal for all three types of basis functions in the FEM approach. When we apply the three basis functions numerically in the FEM approach, this turns out to be correct. For $DOF \gg 1$, the convergence rate is the same for

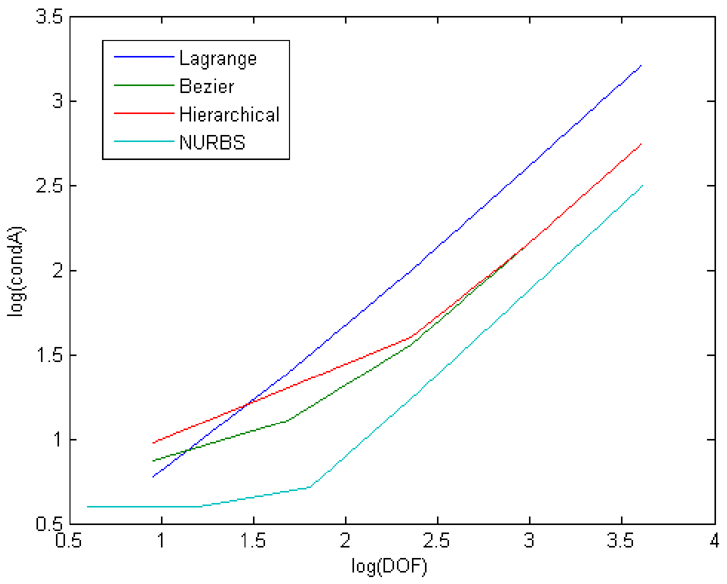


Figure 30: The total number of degrees of freedom plotted against the condition number of the stiffnesmatrix A.

both approaches, but the error is smaller in the IA approach.

As we have discussed earlier in the paper, there will be more elements in the IA case then the FEM case compared to the number of DOFs. This means that in order to construct the stiffness matrix for a fixed number of DOFs, we have to integrate over a greater number of elements in the IA approach. Figure 32 shows the relative error with respect to the total computational time. As we observe from the figure, the FEM approach is more efficient for relative large errors, approximately $0.1 > e > 0.01$, but for smaller errors, approximately $0.01 > e$, the IA approach is more efficient. From the figure we also observe that the Bezier basis functions and Hierarchical basis functions are slightly more efficient then Lagrange basis functions for DOFs $\gg 1$. This is because Bezier basis functions and Hierarchical basis functions produce lower condition numbers then the Lagrange basis functions. The reason why this only shows when DOFs $\gg 1$ is that for small numbers of DOFs, the assembling time will be far greater than the iteration time.

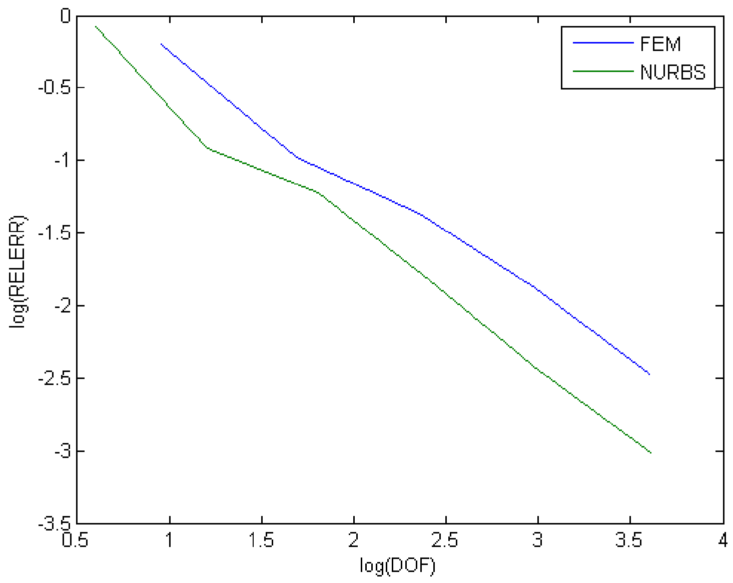


Figure 31: The total number of degrees of freedom plotted against the relative error measured in the energy norm.

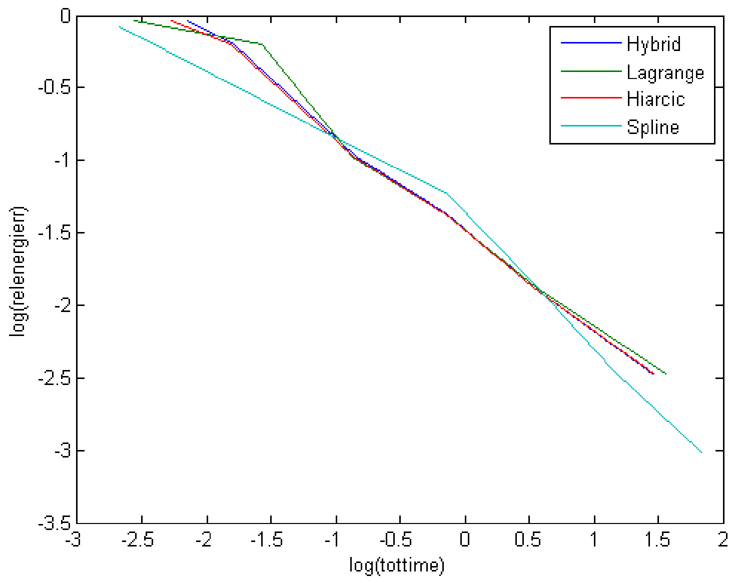


Figure 32: The total computational time plotted against the relative error measured in the energy norm.

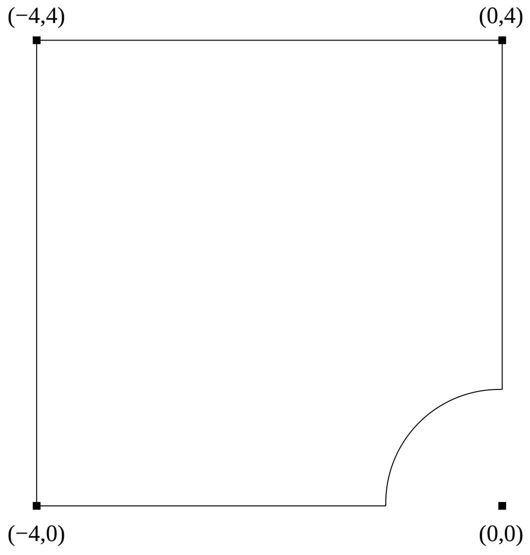


Figure 33: The domain $\Omega = x \in (-4, 0) \cap y \in (0, 4) \cap (x^2 + y^2 \geq 1)$.

4.2 Domain with circular boundary

Now we want to examine how the different approaches behave on a domain with circular boundary. Like we did on the square domain, we will solve a Poisson problem with homogeneous boundary conditions. When we compare the solutions between the different approaches, the same problem with the orientation of the DOFs appear. They can not be distributed in the same manner for both the FEM approach and the IA approach. In addition, on this domain the element size and location will differ in the FEM approach and the IA approach as discussed the theory section in this thesis. In figure 33 the shape of Ω is displayed. As mentioned in the theory section of this thesis, there are in general two ways of representing this domain precisely if we use one patch. That is by either using repeated knot values (NURBS1) or repeated control points (NURBS2). We compare both these IA approaches with the FEM approach. In addition, we will consider a two-patch representation of the physical domain. In order to represent Ω precisely we have to use NURBS basis functions with different weights. We thus expect the assembling procedures in the IA approach to be more time consuming in this problem compared with the previous problem.

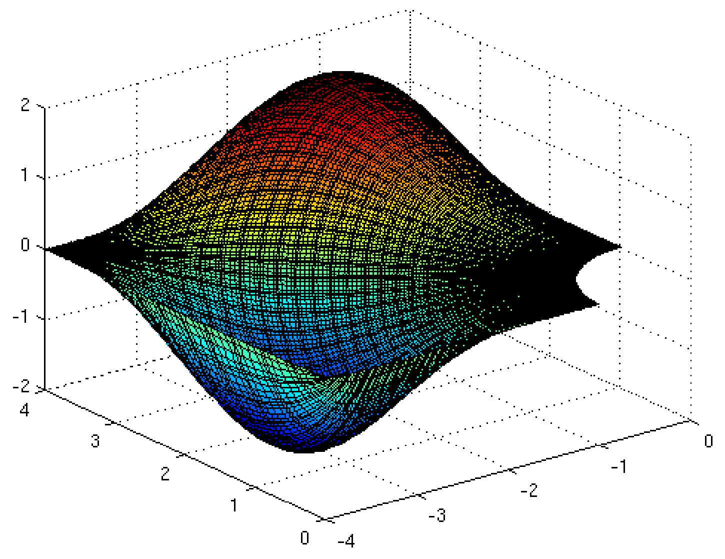


Figure 34: The analytic solution.

4.2.1 u is a low order polynomial

We will now consider the Poisson problem with the analytic solution

$$u(x,y) = \frac{1}{100}xy(x^2+y^2-1)(x-4)(y-4)(x+y). \tag{94}$$

Figure 34 shows the analytical solution to the problem. When we use this $u(x,y)$, the right hand side of Poisson problem becomes

$$f = \frac{1}{100}\cdot\left(\begin{array}{l} y(x^2+y^2-1)(y-4)(x+4)(y+x)+2x^2y(y-4)(x+4)(y+x) \\ +xy(x^2+y^2-1)(y-4)(y+x)+xy(x^2+y^2-1)(y-4)(x+4) \\ +x(x^2+y^2-1)(y-4)(x+4)(y+x)+2xy^2(y-4)(x+4)(y+x) \\ +xy(x^2+y^2-1)(x+4)(y+x)+xy(x^2+y^2-1)(y-4)(x+4) \end{array}\right). \tag{95}$$

One patch

Both in the theory section in this thesis when we discussed refining, and in the implementation section in this thesis, we represented the physical domain with only one patch. As we discussed in the theory section in this thesis, each of the two possible representations in the IA approach has weaknesses when we use one patch to represent

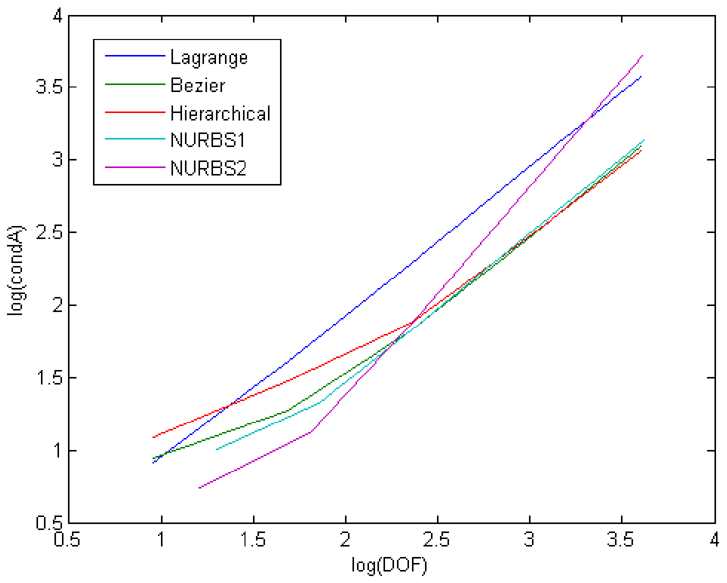


Figure 35: The total number of degrees of freedom plotted against the condition number of the stiffnesmatrix A.

this domain. We thus expect slightly different results than for the previous problem.

Figure 35 shows how the condition number of the stiffness matrices propagates with respect to the total number of DOFs. The three FEM basis functions appear to result in approximately the same growth as they did for the previous problem. The two NURBS approaches on the other hand appear to result in larger condition numbers than the B-spline basis functions did for the previous problem. As we remember from the theory section of this thesis, bisection of the knot vector will result in long, thin elements if we use the NURBS2 approach. This is why the condition number in this approach behaves poorly. If we use the NURBS1 approach we get different, C^0-continuous, basis functions along the diagonal of Ω. This means that even though we have evenly shaped elements, the condition number will be higher because of the different basis functions. As we observe from the figure, the condition number in the NURBS1 approach is approximately equal to the condition number derived form the Bezier basis functions and the Hierarchical basis functions in the FEM approach. This could be due to the fact that on the diagonal, the bezier basis functions and the NURBS basis functions are very similary.

If we compare the relative error with the total number of DOFs, we observe from figure 36 that both IA approaches are more accurate than the FEM approach. In the same manner as with the previous problem, the convergence rate is the same for all approaches when $DOF \gg 1$, and all the FEM basis functions results in the same solution. Even

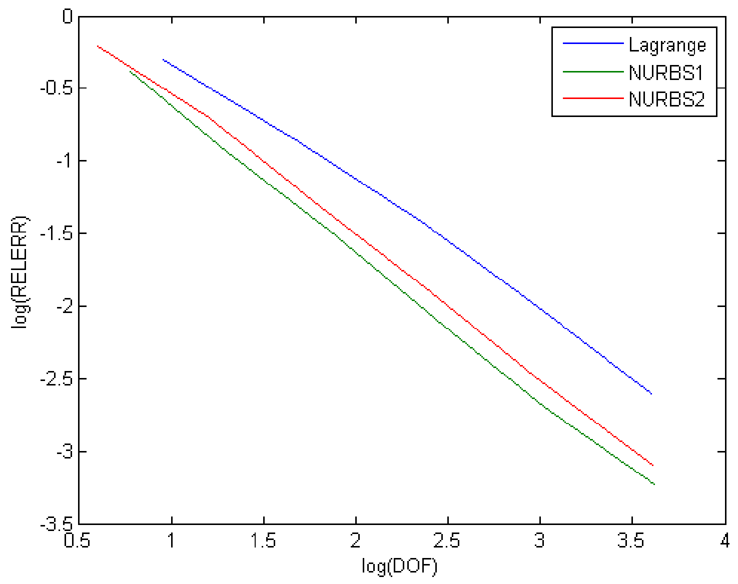


Figure 36: The total number of degrees of freedom plotted against the relative error measured in the energy norm.

though the NURBS2 approach has higher continuity than the NURBS1 approach along the diagonal, the NURBS1 approach is more accurate. This is probably due to the bad mesh in the NURBS2 approach and the fact that $u(x, y) = 0$ along the diagonal. In general, the C^1-continuous NURBS basis functions will produce more accurate solutions.

In the same manner as in the previous problem, we want to compare the relative error with the total solution time for the different approaches. Because of the different weights in the IA approaches we expect the FEM approaches to be more efficient for a larger range of error. Figure 37 shows that this turns out to be correct. The IA approach is more efficient for approximately $e < 0.0075$. The NURBS1 approach is always more efficient than the NURBS2 approach. In addition, the NURBS2 approach will be even more time consuming for larger numbers of DOFs because of the fast growth of the condition number.

Two patches

If we represent Ω with two patches instead of one, we can obtain approximately the same results as we did in the first problem. We will still have C^0-continuity on the diagonal, because it will be on the boundary of the patches, but we will derive much lower condition number in the IA approach. Figure 38 shows the analytic solution to one of the two patches. Figures 39, 40 and 41 show the behavior of the condition number and

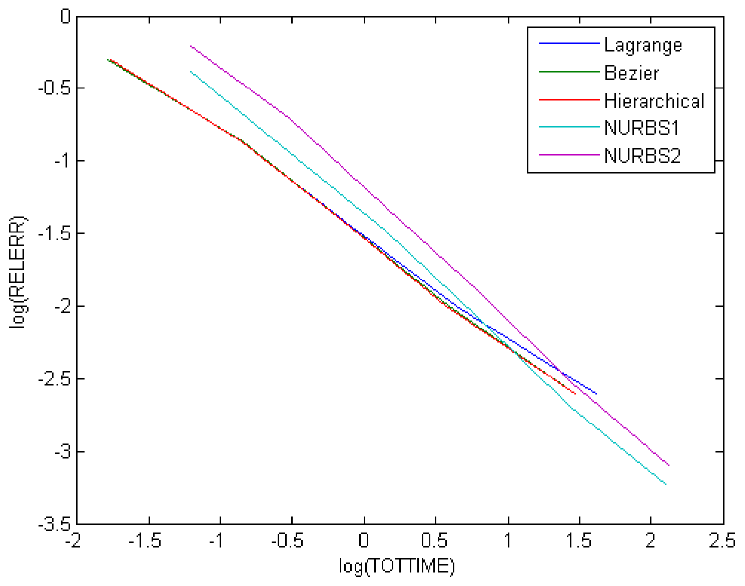


Figure 37: The total computational time plotted against the relative error measured in the energy norm.

relative error when we solve the problem for one of the two patches. From the figures, we observe that the behavior of the condition number and the relative error is very similar to the first problem, and that the condition number in the IA approach is lower in a two-patch representation then in the one-patch representation. In addition, because of the weights we need to require small errors, approximately 0.005, before the IA approach is more efficient then the FEM approach.

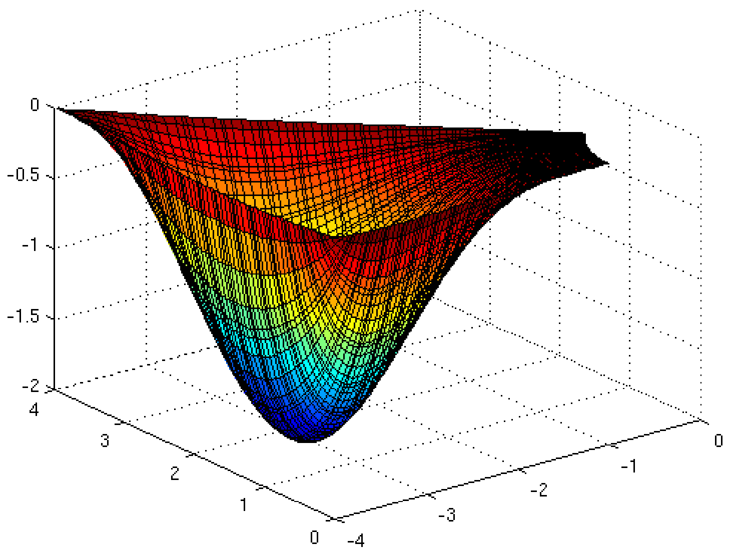


Figure 38: The analytic solution.

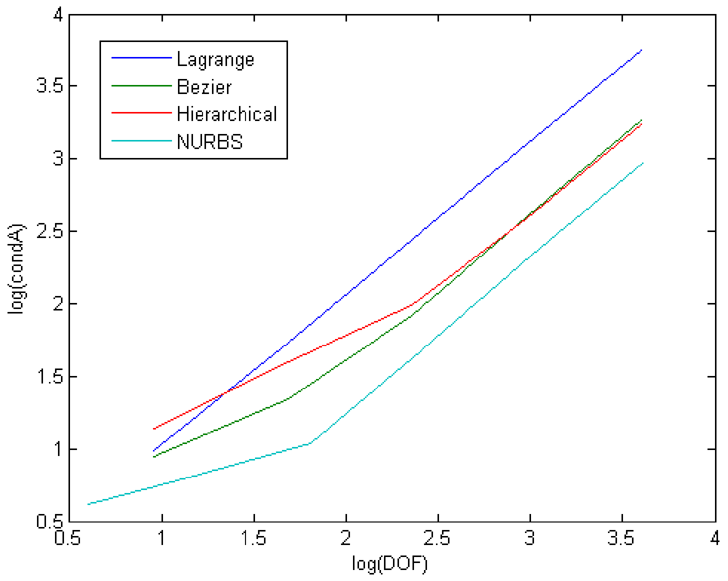


Figure 39: The total number of degrees of freedom plotted against the condition number of the stiffness matrix A.

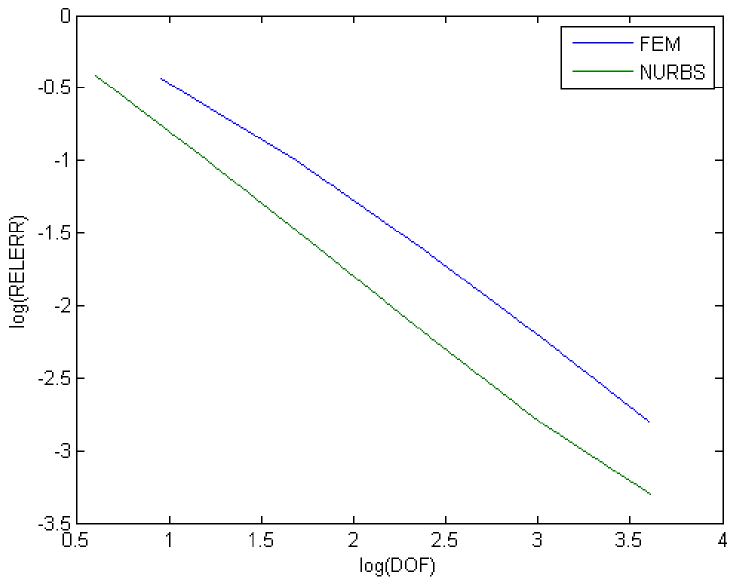


Figure 40: The total number of degrees of freedom plotted against the relative error measured in the energy norm.

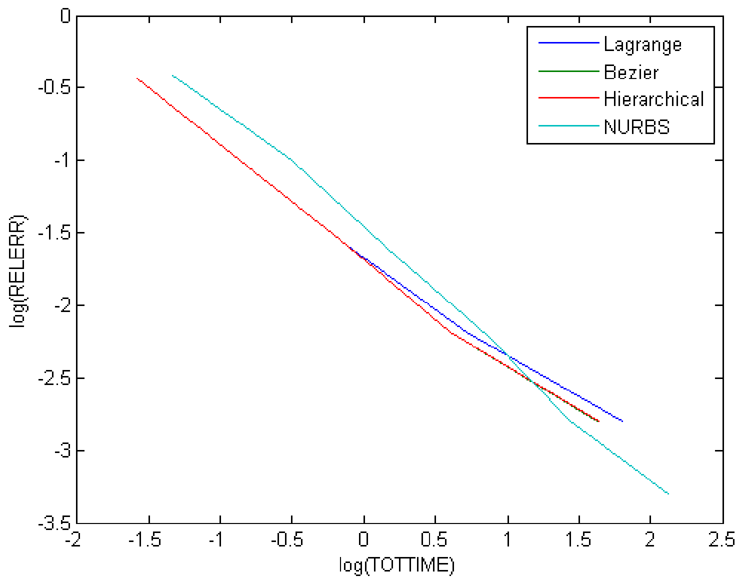


Figure 41: The total computational time plotted against the relative error measured in the energy norm.

5 Conclusion

Throughout this thesis, we have considered differences and similarities between the two numerical methods the finite element method and the isogeomtric analysis, and shown how the two methods can be applied in order to solve ODEs and PDEs. Perhaps the most striking similarity between the two methods, is the disciplinarian approach to solve a problem. In both approaches, we use the weak statement, discretize the domain, introduce a set of basis functions, define the elements and derive a set of algebraic equations. However, within the details there are several fundamental differences.

The theory behind FEM is pretty straightforward. The elements are distributed directly on a physical domain and we use polynomial basis functions that interpolate in at least half the nodal points. The theory behind IA is somewhat harder to comprehend. We have to introduce a parametric space where the elements are defined, and the basis functions are constructed recursively with convex combinations of spline curves, and do not interpolate in the control points. Because the IA basis functions do not interpolate the control points, they produce solution with higher continuity then the FEM basis functions. The "elements" on the physical domain in a IA approach is in fact the image of the elements defined in the parametric space. In addition, in the IA approach the parametric space is local to patches, while in the FEM approach the basis functions are local to subdomains. If we use linear basis functions, $p = 1$, the basis functions and distribution of nodal points/control points will be equal in both approaches.

The fact that we define the elements in an IA approach in a parametric space affects the distribution of the control point on the patch, and the distribution of control points will differ from the distribution of nodal points in a FEM approach for $p > 1$. In addition, because of this representation we derive a much higher number of elements in the IA approach compared to the FEM approach for the same number of DOFs.

In a FEM approach, we will have to communicate with the physical domain when we refine the mesh. In an IA approach the physical domain will be precisely represented with a small amount of patches and elements. Due to the qualities in the refinement schemes in the IA approach where we maintain both the physical and parametric representation, we will not have to communicate with the physical domain when we introduce more control points.

We have discussed three types of basis functions in the FEM approach, namely Lagrange basis functions, Bezier decomposition basis functions and Hierarchical basis functions. In the AI approach have we discussed NURBS basis functions and B-splines basis functions, where B-splines are a special case of NURBS where all the weights are equal. If we use NURBS basis functions we can represent conic sections precisely, this is not possible with the FEM basis functions. All the basis functions in both approaches have compact support, and all the basis functions, except the Hierarchical basis functions, constitute a partition of unity.

The stiffness matrix and the mass matrix will be different in the FEM approach and the IA approach. Internally, the basis functions in the FEM approach produce different matrices, but for problems with homogeneous Dirichlet boundary conditions the solutions are the same. For the Bezier basis functions both matrices are SPD and the band widths are consistent with the support of the basis functions. In addition, all the entries in the mass matrix for the Bezier basis functions are positive and sum to one. All these qualities are shared with the matrices in the IA approach, but since all basis functions in the IA approach, except for a small number near the boundary of the patch, are equal, the band width differ in the two approaches. This difference turns out to be vital when we solve a eigenvalue problem, and the IA approach produces much more accurate solutions.

In general, the IA approach is more complicated to implement then the FEM approach, because of the parametric space, the knot insertion algorithm, the weights and the recursive nature of the basis functions. However, because of these features, the implementation is more general and more easily adaptable to different geometries and degrees on the basis functions. Because of the weights and the large number of elements, the assembly procedure in the IA approach is more time consuming then the FEM approach.

When we solve Poisson problems in two dimensions, we use three Gauss Legendre points in each spatial direction in order to execute the numerical integration in both approaches. This could have been done more efficiently in the IA approach because of the higher continuity of the basis. On a square domain, the growth rate of the condition number is equal for all three FEM basis functions and the IA NURBS basis functions, but the condition number in the IA approach is the lowest. The IA approach produces more accurate solutions compared with the FEM approach when we compare the relative error versus the number of DOFs. When we compare the relative error versus the total computational time, the IA approach is more efficient for approximately $e < 0.01$. Because the Bezier basis functions and Hierarchical basis functions produce lower condition numbers then the Lagrange basis functions, they will be slightly more efficient for DOF $\gg 1$.

When we use one patch to represent a square domain with a hole, we can either repeat two control points or two knot values. Both representations have disadvantages. If we repeat the control points, the image of the elements when we bisect the knot vectors get ill-shaped and produce very high condition numbers. If we repeat the knot values, we lower the degree of the basis along the diagonal. In addition, the condition number will be higher than in the first problem. However, the IA approach still produces more accurate solutions compared to the FEM approach when we compare the relative error versus the number of DOFs. Because not all of the weights are equal in this problem, the computational time in the IA approach will be higher than in the first problem, and the IA approach is more efficient then the FEM approach for approximately $e < 0.0075$. If we represent this domain with two patches instead, we can derive the same results as in the first problem for the condition number and relative error versus the number of DOFs. However, because of the weights the computational time will still be more time consuming in the IA approach.

In conclusion, both methods are well suited to solve ODEs and PDEs, but it seems like the IA approach has a higher potential. Because of the higher continuity in the basis and the fact that all the basis functions except a few are equal, the IA approach results in more accurate solutions than the FEM approach compared with the number of DOFs. The IA basis functions are better suited then the FEM basis functions to represent various geometries. In addition, the IA refinement scheme does not require communication with the physical domain. Because of the recursive nature of the IA basis functions and refinement schemes, we can implement very general and robust methods in order to solve ODEs and PDEs compared to the FEM approach. Because the IA approach requires more elements then the FEM approach compared with the number of DOFs, and because the assembling time is higher in the IA approach, the FEM approach is in this thesis more efficient for errors approximately > 0.0075 on two-dimensional Poisson problems. We expect that a more efficient numerical integration scheme in the IA approach can improve the computational time significantly. In addition, because the IA approach results in lower condition numbers, we expect the IA approach to be even more efficient for three-dimensional problems compared to the FEM approach.

References

Argyris, J. & Kelsey, S. (1960), *Energy Theorems and Structural Analysis*, Butterworths, London.

Bazilevs, Y., de Veiga, L. B., Cottrell, J., Hughes, T. & Sangalli, G. (2006), ‘Isogeometric analysis: approximation, stability and error estimates for h-refined meshes’, *Mathematical Models and Methods in Applies Sciences* .

Braess, D. (2007), *Finite elements, third edition*, Cambridge univeristy press.

Chen, T. & Raju, I. (2003), ‘A coupled finite element and meshless local petrov-galerkin method for two-dimensional problems’, *Computer Methods in Applied Mechanics and Engineering* .

Cottrell, J., Hughes, T. & Baziliévs, Y. (n.d.), Isogeometric Analysis. Towards Unification of CAD and FEA. Preprint.

Cottrell, J., Hughes, T. & Reali, A. (2007), ‘Studies of refinement and continuity in isogeometric structural analysis’, *Computer Methods in Applied Mechanics and Engineering* .

Cottrell, J., Reali, A., Bazilevs, Y. & Hughes, T. (2006), ‘Isogeometric analysis of structural vibrations’, *Computer Methods in Applied Mechanics and Engineering* .

Hughes, T. (1987), *The Finite Element Method - Linear Static and Dynamic Finite Element Analysis*, Prentice-Hall.

Hughes, T., Cottrell, J. & Baziliévs, Y. (2005), ‘Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement’, *Computer Methods in Applied Mechanics and Engineering* .

Hughes, T., Reali, A. & Sangalli, G. (2008), ‘Efficient Quadrature for NURBS-based Isogeometrical Analysis’, *Computer Methods in Applied Mechanics and Engineering* .

Kagan, P., Fischer, A. & Bar-Yoseph, P. (1998), ‘New B-spline finite element approach for geometrical design and mechanical analysis’, *International Journal of Numerical Methods in Engineering* .

Kreyszig, E. (1999), *Advanced Engineering Mmathematics, eight edition*, John Wiley and Sons, inc.

Lyche, T. & Mørken, K. (2006), Splines and B-splines. Lecture notes. UiO - The University of Oslo.

Patera, A. T. (2003), Discretization of the Poisson Problem in R^1: Theory and Implementation. MIT OpenCourseWare.

Patera, A. T. (2004), Finite element methods for the Poisson Problem in R^2. MIT OpenCourseWare.

Rønquist, E. (2008*a*), Deformed Geometries. Lecture note in MA8502, Departement of Mathematical Sciences NTNU.

Rønquist, E. (2008*b*), Spectra of the continuous and discrete Laplace operator. Lecture note in TMA4220, Departement of Mathematical Sciences NTNU.

Saad, Y. (1999), *Iterative Methods for Sparse Linear Systems*, siam.

Sederberg, T., Cardon, D., Finnigan, G., Zhengs, J. & Lyche, T. (2004), 'T-spline simplification and local refinement', *ACM Transactions on Graphics* .

Sederberg, T., Zhengs, J., Bakenov, A. & Nasiri, A. (2003), 'T-splines and T-NURCCSs', *ACM Transactions on Graphics* .

Silvester, P. & Ferrari, R. (1996), *Finite Element for Electrical Engineers, third edition*, Cambridge University Press.

Young, N. (1988), *An introduction to Hilbert space*, Cambridge univeristy press.

A Coordinates and weights for the square domain with circular hole.

Table 3: Nodal coordinates of the FEM representation of Ω with two elements.

(-1,0)	(-2.5,0)	(-4,0)
-(1+1/$\sqrt{2}$)/2,(1/$\sqrt{2}$)/2	(-2.5,1.25)	(-4,2)
(-1/$\sqrt{2}$,1/$\sqrt{2}$)	(-2.5,2.5)	(-4,4)
(-(1/$\sqrt{2}$)/2,(1+1/$\sqrt{2}$)/2))	(-1.25,2.5)	(-2,4)
(0,1)	(0,2.5)	(0,4)

Table 4: Control net for the IA repeated control point representation of Ω with two elements.

(-1,0)	(-2.5,0)	(-4,0)
(-1,$\sqrt{2}$-1)	(-2.5,0.75)	(-4,4)
(1-$\sqrt{2}$,1)	(-0.75,2.5)	(-4,4)
(0,1)	(0,2.5)	(0,4)

Table 5: Weights for the IA repeated control point representation of Ω with two elements.

1	1	1
(1+1/$\sqrt{2}$)/2)	1	1
(1+1/$\sqrt{2}$)/2)	1	1
1	1	1

Table 6: Control net for the IA repeated knot value representation of Ω with two elements.

(-1,0)	(-2.5,0)	(-4,0)
(-1,$\sqrt{2}$-1)	(-2.375,1.125)	(-4,2)
(-1/$\sqrt{2}$,1/$\sqrt{2}$)	(-1.75,1.75)	(-4,4)
(1-$\sqrt{2}$,1)	(-1.125,2.375)	(-2,4)
(0,1)	(0,2.5)	(0,4)

Table 7: Weights for the IA repeated knot value representation of Ω with two elements.

1	1	1
(1+1/$\sqrt{2}$)/2)	1	1
(1+1/$\sqrt{2}$)/2)	1	1
(1+1/$\sqrt{2}$)/2)	1	1
1	1	1