NTNU – Trondheim
Norwegian University of
Science and Technology

Isogeometric Methods for Computational Fluid Dynamics: Divergence-conforming Discretizations for the 2D Stokes Equations

Finn-Idar Grøtta Giske

Master of Science in Physics and Mathematics
Submission date: June 2014
Supervisor: Trond Kvamsdal, MATH
Co-supervisor: Kjetil Andre Johannessen, IMF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

This thesis is about investigating the use of Isogeometric Analysis with divergence-conforming spline spaces for solving incompressible Stokes flow problems. By means of divergence-conforming spaces one may achieve pointwise fulfillment of the incompressibility condition, and initial work published on this method have shown encouraging results regarding stability and accuracy.

Abstract

In this thesis we look at how boundary value problems for partial differential equations can be solved numerically using B-splines, or more generally NURBS, both to express the geometry of the problem exactly and as a basis for a finite element approximation. This is called isogeometric analysis, and we consider the theory behind the method as well as aspects regarding implementation. We take a close look at the construction of B-spline basis functions and geometries, and how the basis can be refined, leading up to the construction of NURBS basis functions and geometries. The ubiquitous Poisson problem is considered as a model problem, and a numerical solver for this problem is implemented in MATLAB using Galerkin's finite element method. We finally consider a method for numerically solving the Stokes problem for incompressible fluid flow, using divergence-conforming B-splines in an isogeometric setting. This method gives a discrete velocity which is pointwise divergence-free, making the numerical solution satisfy mass conservation in an exact sense. Numerical tests are performed, showing that isogeometric analysis makes it possible to use exact geometry throughout the analysis and provides great flexibility regarding refinement. The convergence properties of the method for the Stokes problem are investigated numerically, with very good results for the numerical velocity solution, but with a reduced convergence rate for the pressure solution that is accounted for. The method is also tested on benchmark problems, the results confirming the stability of the method.

Sammendrag

I denne masteroppgaven ser vi på hvordan randverdiproblemer of partielle differensialligninger kan løses numerisk ved å bruke B-splines, eller mer generelt NURBS, både til å uttrykke problemets geometri eksakt og som basis for en tilnærmet løsning med elementmetoden. Dette blir kalt isogeometrisk analyse, og vi studerer teorien bak metoden i tillegg til implementasjonsmessige aspekter. Vi går grundig gjennom oppbygningen av B-spline basisfunksjoner og geometrier. Det allestedsnærværende Poisson-problemet er brukt som modellproblem, og en numerisk løser for dette problemet er implementert i MATLAB med bruk av Galerkins elementmetode. Vi ser til slutt på en metode for numerisk løsning av Stokes-problemet for inkompressibel væskestrømning, der divergenskonforme B-splines er brukt i en isogeometrisk setting. Denne metoden gir en diskret hastighet som er punktvis divergensfri, noe som gjør at den numeriske løsningen oppfyller massebevarelse eksakt. Numeriske tester er utført, og disse viser at isogeometrisk analyse gjør det mulig å bruke eksakt geometri gjennom hele analyseprosessen og gir stor fleksibilitet hva angår forfining. Konvergensegenskapene til metoden for Stokesproblemet er undersøkt numerisk, med svært gode resultater for den numeriske løsningen for hastighet, men med en redusert konvergensrate for trykkløsningen som blir gjort rede for. Metoden er også testet på referanseproblemer med resultater som bekrefter metodens stabilitet.

Acknowledgements

The process of writing this thesis has been a great way to delve deep into the field of isogeometric analysis and learn more about numerical solution of partial differential equations, and I am grateful for being given this opportunity. I would like to thank my supervisor Trond Kvamsdal for the feedback and guidance he has provided, making this thesis come together. Also great thanks to my co-supervisor Kjetil A. Johannesen for his constant availability, his great inspiration and for the valuable knowledge he has shared. Finally, I owe my wife Ingvild a large thanks for being patient and encouraging, giving motivation to perform at my best.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Isogeometric analysis

Computer aided design (CAD) is the use of computers to construct and modify geometric objects. CAD is widely used in industry to give a description and a visualization of the objects being designed. Computer aided engineering (CAE) is the use of computers to perform engineering analysis on geometries. Some examples are stress analysis, thermal and fluid flow analysis, dynamics and kinematics analysis and product optimization. The physical phenomena being analysed are often modelled mathematically by boundary value problems of partial differential equations. These problems must usually be solved using a numerical method, especially in the case of complex geometries, and a widely used method is the finite element method.

Finite element analysis (FEA) is often used as a part of the design process to control and verify the design, and based on the analysis the design may be improved. However, the technologies of FEA and CAD have been developed quite separately, and in order to perform analysis on a CAD geometry it must be translated into a geometry suitable for analysis. This translation process has proven to be time consuming [7, p. 2]. If the design is improved multiple times on the basis of the analysis, it is obvious that a more seamless integration of CAD and FEA would be preferable. In addition, the geometry used for analysis is often an approximation to the CAD geometry which is viewed as the exact geometry. This may introduce errors, e.g. [9] shows an example where using an approximated geometry introduces unwanted numerical oscillations in the solution.

Pursuing the goal of bringing the CAD and FEA communities closer together, the concept of isogeometric analysis was proposed in [18] as a way to perform finite element analysis using the exact CAD geometry. The most widely used CAD technology in engineering design is NURBS (non-uniform rational B-splines) [7, p. 7], and in isogeometric analysis NURBS basis functions are used both to represent the geometry and as basis functions for finite element analysis. The methods and ideas of isogeometric

analysis are described in the book "Isogeometric Analysis - Toward Integration of CAD and FEA" [7] and much of the theory and implementation in this thesis is based on this book.

1.2 The Stokes problem

The Stokes equations give a model for the flow of an incompressible fluid with constant viscosity, in the case where the viscous forces dominate over the inertial forces [20]. This is characterized by a very small Reynolds number, for instance if the velocity or the length scale of the flow is very small, or the viscosity is very large. A huge class of numerical approximations of this problem is obtained using the framework of mixed finite element methods [14]. Using this framework must, however, be done with some care in order to guarantee stability and convergence of the method. Furthermore, a lot of these methods give discrete solutions that are only approximately divergence-free, which can be an issue [15, 21]. A method satisfying the stability requirements while at the same time giving exact mass conservation is proposed in [11] in the context on isogeometric analysis, with very promising results.

1.3 This thesis

Some familiarity with the finite element method is recommended before reading this thesis, even though we go in some detail explaining the methods. In Chapter 2 the theory of B-splines and NURBS will be treated in quite some depth, with a focus on the properties that will be important for their use as finite element basis functions. We will take a close look at spline spaces, as well as methods for refining the basis of a given spline space. In Chapter 3 we use the NURBS basis functions in a finite element implementation for the Poisson problem, according to the principles of isogeometric analysis. In Chapter 4 we consider a method for numerically solving the Stokes problem for incompressible fluid flow, using divergence-conforming spline spaces. Chapter 5 summarizes the results and contains the conclusion of this thesis.

This thesis is written as an extension of the work done in the specialization project in numerical analysis [16], so much of what is presented here comes from this project.

Chapter 2

B-splines and NURBS

In isogeometric analysis B-spline or NURBS basis functions are used both to define the geometry and as the basis for the solution space in which we search for the numerical solution of the problem at hand. This idea of using the same basis for the geometry and the analysis is referred to as the isoparametric concept [7, p. 69]. The NURBS basis functions are defined using the B-spline basis functions, and the former is actually a generalization of the latter. Thus in order to use the NURBS basis functions we must understand how the B-spline basis functions behave. In this chapter we take a thorough look into the theory of B-splines and NURBS. We start by defining the B-spline basis functions and explore some of their most important properties. We then consider spline spaces, which the B-splines give a basis for, and spaces of spline derivatives. We also consider the ways that the basis of a spline space can be refined, before we see how the NURBS are constructed using B-splines.

2.1 B-splines

2.1.1 Definition

Spline functions and spline geometries are built from basis functions called B-splines. We start by defining these basis functions.

Definition 2.1. *Let p be a nonnegative integer and $\Xi = [\xi_1, \xi_2, \ldots, \xi_{n+p+1}]$ a vector of real numbers that are nondecreasing with increasing index. The i^{th} of the n B-splines of degree p, denoted $N_{i,p,\Xi}$ or simply $N_{i,p}$, is defined by the recursion*

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi), \qquad \forall \xi \in \mathbb{R}, \quad (2.1.1)$$

using the convention "$0/0 = 0$*" and the base case*

$$N_{i,0}(\xi) = \begin{cases} 1, & \textit{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0, & \textit{otherwise.} \end{cases} \tag{2.1.2}$$

If a knot value is repeated r times in the knot vector we say that it has *multiplicity* r. We say that a knot vector is *open* or $p+1$*-regular* if the multiplicity of both the first and the last knot is $p+1$ and no knot has multiplicity larger than $p+1$. In this thesis we will only encounter open knot vectors.

Obviously division by zero may occur in Definition 2.1 in the case of repeated knots. However, the zero convention "$0/0 = 0$" prevents this from creating any trouble, since every time division by zero occurs we also multiply by zero. Indeed we can note the following property:

Proposition 1. *For* $p \geq 1$ *we have that*

$$\xi_i = \xi_{i+p} \quad \Rightarrow \quad N_{i,p-1}(\xi) = 0, \forall \xi \in \mathbb{R}. \tag{2.1.3}$$

Proof. The property is easily proved by induction. For $p = 1$ the result is obvious from (2.1.2) since there is no $\xi \in \mathbb{R}$ such that $\xi_i \leq \xi < \xi_{i+1}$ if we have $\xi_i = \xi_{i+1}$. Now assume that (2.1.3) holds for $p = k$. From the monotonicity of the knot vector we observe that if $\xi_i = \xi_{i+k+1}$ then $\xi_i = \xi_{i+k}$ and $\xi_{i+1} = \xi_{i+k+1}$. Thus (2.1.1) gives that (2.1.3) holds for $k+1$ by the following calculation for $\xi \in \mathbb{R}$:

$$\begin{aligned} N_{i,k}(\xi) &= \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} N_{i,k-1}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} N_{i+1,k-1}(\xi) \\ &= (\xi - \xi_i)\frac{0}{0} + (\xi_{i+k+1} - \xi)\frac{0}{0} = 0. \end{aligned}$$

Here we have used the induction hypothesis and the zero convention. □

B-spline evaluation can of course be implemented using the recursion in Definition 2.1 with the zero convention. However, a much more efficient way of evaluating the B-splines can be obtained using a matrix representation of the B-splines [22, Ch. 2] and this is what we use in practice.

2.1.2 Basic properties of B-splines

We now consider some of the properties of B-splines that are important for their use as finite element basis functions. The first property to note is the local support, which is important in order to end up with a sparse system of linear equations when the B-splines are used as a finite element basis.

Proposition 2. *A B-spline $N_{i,p}$ satisfies*

$$N_{i,p}(\xi) > 0, \qquad \xi \in (\xi_i, \xi_{i+p+1}), \tag{2.1.4}$$

$$N_{i,p}(\xi) = 0, \qquad \xi \notin [\xi_i, \xi_{i+p+1}), \tag{2.1.5}$$

so the support of $N_{i,p}$ is $[\xi_i, \xi_{i+p+1}]$ and it is actually strictly positive on the interior of the support.

Proposition 2 can be proved straightforwardly by induction, using the recurrence relation in Definition 2.1 along with Proposition 1. An immediate consequence of Proposition 2 is that given a knot interval $[\xi_\mu, \xi_{\mu+1})$, the $p+1$ nonzero B-splines of degree p on the interval are $\{N_{i,p}\}_{i=\mu-p}^{\mu}$.

Another well known property of the B-splines is that they are piecewise polynomial functions.

Proposition 3. *A B-spline can be expressed by*

$$N_{i,p}(\xi) = \sum_{j=i}^{i+p} P_{i,p}^{j}(\xi)\, N_{j,0}(\xi), \tag{2.1.6}$$

where the $P_{i,p}^{j}$ are polynomials of degree at most p and $N_{j,0}$ are the piecewise constant B-splines, i.e. the characteristic functions of the knot intervals.

Proof. To see that we can express the B-splines this way we use induction. It is trivial that (2.1.6) holds for $p = 0$, since we can use $P_{i,0}^{i} = 1$ which is a polynomial of degree 0. Now if we assume that (2.1.6) holds for $p = k-1$ we can write

$$\begin{aligned}
N_{i,k}(\xi) &= \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} N_{i,k-1}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} N_{i+1,k-1}(\xi) \\
&= \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} \sum_{j=i}^{i+k-1} P_{i,k-1}^{j}(\xi)\, N_{j,0}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} \sum_{j=i+1}^{i+k} P_{i+1,k-1}^{j}(\xi)\, N_{j,0}(\xi) \\
&= \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} P_{i,k-1}^{i}(\xi)\, N_{i,0}(\xi) \\
&\quad + \sum_{j=i+1}^{i+k-1} \left(\frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} P_{i,k-1}^{j}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} P_{i+1,k-1}^{j}(\xi) \right) N_{j,0}(\xi) \\
&\quad + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} P_{i+1,k-1}^{i+k}(\xi)\, N_{i+k,0}(\xi),
\end{aligned}$$

and we observe that the functions in front of the $N_{j,0}$ for all $j \in \{i, \ldots, i+k\}$ are the $(k-1)^{th}$ degree polynomials $P_{i,k-1}^{j}$ multiplied by first degree polynomials, or sums of such products. Hence we have k^{th} degree polynomials $P_{i,k}^{j}$ such that (2.1.6) holds for $p = k$. □

Under the assumption of open knot vectors the B-spline basis $\{N_{i,p}\}_{i=1}^{n}$ is a partition of unity, which is a useful property when used in finite element analysis.

Proposition 4. *If the B-splines $\{N_{i,p}\}_{i=1}^{n}$ are constructed from an open knot vector, then*

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1, \qquad \forall \xi \in [\xi_1, \xi_{n+p+1}). \tag{2.1.7}$$

Proof. We use the recurrence relation (2.1.1) and compute

$$\begin{aligned}\sum_{i=1}^{n} N_{i,p}(\xi) &= \sum_{i=1}^{n} \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \sum_{i=1}^{n} \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \\ &= \frac{\xi - \xi_1}{\xi_{1+p} - \xi_1} N_{1,p-1}(\xi) + \sum_{i=2}^{n} N_{i,p-1}(\xi) + \frac{\xi_{n+p+1} - \xi}{\xi_{n+p+1} - \xi_{n+1}} N_{n+1,p-1}(\xi).\end{aligned}$$

If the knot vector is open we have that $\xi_1 = \xi_{p+1}$ and $\xi_{n+1} = \xi_{n+p+1}$, and by Proposition 1 this gives $N_{1,p-1}(\xi) = N_{n+1,p-1}(\xi) = 0$. Hence

$$\sum_{i=1}^{n} N_{i,p}(\xi) = \sum_{i=2}^{n} N_{i,p-1}(\xi).$$

Now we can proceed in a similar fashion until we reach $\sum_{i=p+1}^{n} N_{i,0}(\xi)$. For every $\xi \in [\xi_1, \xi_{n+p+1})$ there is exactly one index i such that $\xi_i \leq \xi < \xi_{i+1}$, and since the knot vector is open this index will satisfy $p+1 \leq i \leq n$, so clearly

$$\sum_{i=1}^{n} N_{i,p}(\xi) = \sum_{i=p+1}^{n} N_{i,0}(\xi) = 1, \qquad \forall \xi \in [\xi_1, \xi_{n+p+1}). \tag{2.1.8}$$

□

Finally we state the important smoothness result which lets us know the smoothness of the B-splines simply by looking at the knot vector.

Proposition 5. *If a knot ξ_i has multiplicity r then all B-splines $N_{i,p}$ are C^{p-r}-continuous at the knot ξ_i.*

In the intervals between the knots the B-splines will be C^∞, a consequence of being represented by polynomials in these intervals as seen in Proposition 3. Figure 2.1.1 and Figure 2.1.2 give an example of this continuity result. We see that with a double knot at $\xi = 1$ as in Figure 2.1.1 the basis is $C^{2-2} = C^0$-continuous at $\xi = 1$, while with a single knot at $\xi = 1$ as in Figure 2.1.2 the basis is $C^{2-1} = C^1$-continuous at $\xi = 1$.

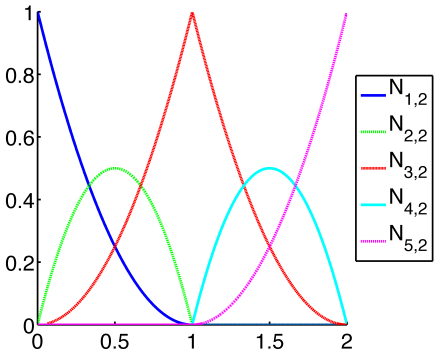

Figure 2.1.1: The quadratic B-splines for the knot vector $\Xi = [0, 0, 0, 1, 1, 2, 2, 2]$.

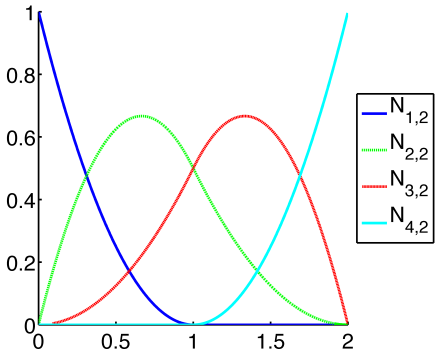

Figure 2.1.2: The quadratic B-splines for the knot vector $\Xi = [0, 0, 0, 1, 2, 2, 2]$.

2.1.3 The derivatives of the B-splines

The first order derivative of a B-spline $N_{i,p}$ when $p \geq 1$ is given by the formula

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{2.1.9}$$

The B-splines $N_{i,0}$ are piecewise constant, so their derivatives will be identically zero. Note that even though a B-spline might not be differentiable in the ordinary sense at the knots, we define the derivative at the knots to be the limit of the derivative from the right. We prove the formula (2.1.9), again using induction.

Proof. We know that the function $N_{i,1}$ is the piecewise linear "hat function" which has the value 1 at ξ_{i+1} and is 0 at all other knots. With this in mind it is clear that we have

$$\frac{d}{d\xi} N_{i,1}(\xi) = \frac{1}{\xi_{i+1} - \xi_i} N_{i,0}(\xi) - \frac{1}{\xi_{i+2} - \xi_{i+1}} N_{i+1,0}(\xi),$$

when the derivative at the knots is defined as the limit of the derivative from the right. Thus (2.1.9) holds for $p = 1$. We now consider $p = k$ and assume that (2.1.9) holds for $p = k - 1$. If we differentiate on each side of the recursion formula (2.1.1) we obtain

$$\begin{aligned}\frac{d}{d\xi} N_{i,k}(\xi) = {} & \frac{1}{\xi_{i+k} - \xi_i} N_{i,k-1}(\xi) - \frac{1}{\xi_{i+k+1} - \xi_{i+1}} N_{i+1,k-1}(\xi) \\ & + \frac{\xi - \xi_i}{\xi_{i+k} - \xi_i} \frac{d}{d\xi} N_{i,k-1}(\xi) + \frac{\xi_{i+k+1} - \xi}{\xi_{i+k+1} - \xi_{i+1}} \frac{d}{d\xi} N_{i+1,k-1}(\xi).\end{aligned}$$

Using the induction hypothesis this gives that

$$\begin{aligned}
\frac{dN_{i,k}}{d\xi}(\xi) &= \frac{1}{\xi_{i+k}-\xi_i}N_{i,k-1}(\xi) - \frac{1}{\xi_{i+k+1}-\xi_{i+1}}N_{i+1,k-1}(\xi) \\
&+ \frac{\xi-\xi_i}{\xi_{i+k}-\xi_i}\left(\frac{k-1}{\xi_{i+k-1}-\xi_i}N_{i,k-2}(\xi) - \frac{k-1}{\xi_{i+k}-\xi_{i+1}}N_{i+1,k-2}(\xi)\right) \\
&+ \frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+1}}\left(\frac{k-1}{\xi_{i+k}-\xi_{i+1}}N_{i+1,k-2}(\xi) - \frac{k-1}{\xi_{i+k+1}-\xi_{i+2}}N_{i+2,k-2}(\xi)\right) \\
\frac{dN_{i,k}}{d\xi}(\xi) &= \frac{1}{\xi_{i+k}-\xi_i}N_{i,k-1}(\xi) - \frac{1}{\xi_{i+k+1}-\xi_{i+1}}N_{i+1,k-1}(\xi) \\
&+ \frac{k-1}{\xi_{i+k}-\xi_i}\frac{\xi-\xi_i}{\xi_{i+k-1}-\xi_i}N_{i,k-2}(\xi) - \frac{k-1}{\xi_{i+k+1}-\xi_{i+1}}\frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+2}}N_{i+2,k-2}(\xi) \\
&+ \frac{k-1}{\xi_{i+k}-\xi_{i+1}}\left(\frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+1}} - \frac{\xi-\xi_i}{\xi_{i+k}-\xi_i}\right)N_{i+1,k-2}(\xi).
\end{aligned}$$

We would now like to add $\frac{\xi_{i+k}-\xi_i}{\xi_{i+k}-\xi_i} - \frac{\xi_{i+k+1}-\xi_{i+1}}{\xi_{i+k+1}-\xi_{i+1}}$ in the above parentheses. This must, however, be done with some care. Because of the zero convention "$0/0 = 0$", this expression might not be zero if $\xi_i = \xi_{i+k}$ or $\xi_{i+1} = \xi_{i+k+1}$. Luckily in both those cases we have $\xi_{i+1} = \xi_{i+k}$ from the monotonicity of the knot vector, and by Proposition 1 this implies that $N_{i+1,k-2}(\xi) = 0$. Hence equality still holds if we add $\frac{\xi_{i+k}-\xi_i}{\xi_{i+k}-\xi_i} - \frac{\xi_{i+k+1}-\xi_{i+1}}{\xi_{i+k+1}-\xi_{i+1}}$ in the parentheses. Thus we obtain

$$\begin{aligned}
&\frac{d}{d\xi}N_{i,k}(\xi) \\
&= \frac{1}{\xi_{i+k}-\xi_i}N_{i,k-1}(\xi) - \frac{1}{\xi_{i+k+1}-\xi_{i+1}}N_{i+1,k-1}(\xi) \\
&+ \frac{k-1}{\xi_{i+k}-\xi_i}\frac{\xi-\xi_i}{\xi_{i+k-1}-\xi_i}N_{i,k-2}(\xi) - \frac{k-1}{\xi_{i+k+1}-\xi_{i+1}}\frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+2}}N_{i+2,k-2}(\xi) \\
&+ \frac{k-1}{\xi_{i+k}-\xi_{i+1}}\left(\frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+1}} - \frac{\xi_{i+k+1}-\xi_{i+1}}{\xi_{i+k+1}-\xi_{i+1}} + \frac{\xi_{i+k}-\xi_i}{\xi_{i+k}-\xi_i} - \frac{\xi-\xi_i}{\xi_{i+k}-\xi_i}\right)N_{i+1,k-2}(\xi) \\
&= \frac{1}{\xi_{i+k}-\xi_i}N_{i,k-1}(\xi) - \frac{1}{\xi_{i+k+1}-\xi_{i+1}}N_{i+1,k-1}(\xi) \\
&+ \frac{k-1}{\xi_{i+k}-\xi_i}\left(\frac{\xi-\xi_i}{\xi_{i+k-1}-\xi_i}N_{i,k-2}(\xi) + \frac{\xi_{i+k}-\xi}{\xi_{i+k}-\xi_{i+1}}N_{i+1,k-2}(\xi)\right) \\
&- \frac{k-1}{\xi_{i+k+1}-\xi_{i+1}}\left(\frac{\xi-\xi_{i+1}}{\xi_{i+k}-\xi_{i+1}}N_{i+1,k-2}(\xi) + \frac{\xi_{i+k+1}-\xi}{\xi_{i+k+1}-\xi_{i+2}}N_{i+2,k-2}(\xi)\right).
\end{aligned}$$

Finally we recognize the expressions in the parentheses above from the recursion for-

mula (2.1.1) as $N_{i,k-1}(\xi)$ and $N_{i+1,k-1}(\xi)$ and we obtain

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

This shows that the formula (2.1.9) holds for all $p \geq 1$. □

2.2 Linear combinations of B-splines

Spline geometries can be defined by taking linear combinations of the B-splines. In this thesis we will only consider one- and two-dimensional geometries, but the generalization from two to three dimensions is fairly straightforward. We will throughout the rest of this thesis assume that all knot vectors are open. This assumption was needed for the B-spline basis to be a partition of unity as seen in Proposition 4, a property which is needed when we use these functions as a basis for the finite element analysis. Assuming open knot vectors also guarantees linear independence of the B-splines [22, Ch. 3].

2.2.1 Univariate spline functions

A linear combination of the B-spline basis functions $v = \sum_{i=1}^{n} c_i N_{i,p}$, $c_i \in \mathbb{R}$, is called a spline function, or simply a spline. We now define the function space containing all such linear combinations of a given B-spline basis.

Definition 2.2. *Let p be a nonnegative integer and $\Xi = [\xi_1, \ldots, \xi_{n+p+1}]$ an open knot vector. We define the spline space $S_{p,\Xi}$ on the interval $[\xi_1, \xi_{n+p+1})$ by*

$$S_{p,\Xi} := \operatorname{span}\left\{N_{i,p}|_{[\xi_1,\xi_{n+p+1})}\right\}_{i=1}^{n},$$

where $N_{i,p}|_{[\xi_1,\xi_{n+p+1})}$ means the B-spline $N_{i,p}$ restricted to the interval $[\xi_1, \xi_{n+p+1})$.

The B-splines are piecewise polynomial functions of degree at most p (Property 3), so it is clear that $S_{p,\Xi}$ contains piecewise polynomial functions of degree at most p. As the knot vector is assumed to be open, the B-splines $\{N_{i,p}\}_{i=1}^{n}$ are linearly independent on the interval $[\xi_1, \xi_{n+p+1})$ [22, Ch. 3], so $\{N_{i,p}\}_{i=1}^{n}$ is in fact a basis for $S_{p,\Xi}$. Hence the dimension of the space $S_{p,\Xi}$ is simply

$$\dim S_{p,\Xi} = n = \#\Xi - p - 1. \tag{2.2.1}$$

Indeed, since $n+p+1$ is the last index into Ξ, n is calculated by subtracting $p+1$ from the number of elements in Ξ, denoted by $\#\Xi$.

In general we can define spaces of piecewise polynomial functions with prescribed smoothness in the following way:

Definition 2.3. *Let p be a nonnegative integer, $\boldsymbol{\zeta} = \{\zeta_i\}_{i=1}^N$ a strictly increasing sequence of real numbers and $\boldsymbol{\alpha} = \{\alpha_i\}_{i=1}^N$ a sequence of integers such that $\alpha_i \geq -1$ with equality for $i = 1$ and $i = N$, i.e. $\boldsymbol{\alpha} = \{-1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_{N-1}, -1\}$. We then define $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ to be the linear space of piecewise polynomials of degree p on $[\zeta_1, \zeta_N)$ with α_i continuous derivatives at ζ_i. The elements of $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ will belong to $C^{|\boldsymbol{\alpha}|}([\zeta_1, \zeta_N))$ where*

$$|\boldsymbol{\alpha}| := \min_{2 \leq i \leq N-1} \{\alpha_i\}.$$

The sequence $\boldsymbol{\alpha}$ is referred to as the regularity sequence, or the regularity vector. Having $\alpha_i = -1$ in the definition above means that the elements of $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ can have a discontinuity at ζ_i, so there is no restriction on the smoothness at this point. Assuming $\alpha_i \leq p$ for all $i \in \{2, \ldots, N-1\}$ the dimension of the space $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ is given by the following formula [22, Ch. 3]

$$\dim S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta}) = (N-1)p + 1 - \sum_{i=2}^{N-1} \alpha_i. \tag{2.2.2}$$

For each subinterval $[\zeta_i, \zeta_{i+1})$ we have a polynomial of degree p contributing with $p+1$ degrees of freedom, and the total number of smoothness conditions is $\sum_{i=2}^{N-1}(\alpha_i + 1)$. Hence it is reasonable that we end up with $(N-1)(p+1) - \sum_{i=2}^{N-1}(\alpha_i + 1) = (N-1)p + 1 - \sum_{i=2}^{N-1} \alpha_i$ degrees of freedom in total. That the assumption $\alpha_i \leq p$ is needed for the formula (2.2.2) to hold is clear from the following proposition.

Proposition 6. *Let $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ be a given space as defined in Definition 2.3, and let $\bar{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\zeta}}$ be the subsequences obtained by removing all elements $\alpha_i \geq p$ from $\boldsymbol{\alpha}$ and the corresponding elements from $\boldsymbol{\zeta}$. Then we have that*

$$S_{\bar{\boldsymbol{\alpha}}}^p(\bar{\boldsymbol{\zeta}}) = S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta}).$$

Hence requiring C^p-continuity or more at a point ζ_i will actually give C^∞-continuity for the elements of $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ at ζ_i.

Proof. We first consider the case where $\alpha_i \leq p$ for all $i \in \{2, \ldots, N-1\}$. In this case we can apply (2.2.2) for both spaces $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ and $S_{\bar{\boldsymbol{\alpha}}}^p(\bar{\boldsymbol{\zeta}})$ and note that they have the same dimension. Clearly $S_{\bar{\boldsymbol{\alpha}}}^p(\bar{\boldsymbol{\zeta}}) \subseteq S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ since elements of $S_{\bar{\boldsymbol{\alpha}}}^p(\bar{\boldsymbol{\zeta}})$ will in fact be C^∞-continuous at ζ_j. Thus $S_{\bar{\boldsymbol{\alpha}}}^p(\bar{\boldsymbol{\zeta}})$ is a subspace of $S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta})$ with the same dimension, which means that they must be the same space.

For the case where $\alpha_i > p$ we let $\boldsymbol{\beta}$ be the sequence obtained by replacing with p all elements $\alpha_i > p$ in $\boldsymbol{\alpha}$, then clearly $\bar{\boldsymbol{\alpha}} = \bar{\boldsymbol{\beta}}$. Now observe that

$$S_{\bar{\boldsymbol{\alpha}}}^p(\bar{\boldsymbol{\zeta}}) \subseteq S_{\boldsymbol{\alpha}}^p(\boldsymbol{\zeta}) \subseteq S_{\boldsymbol{\beta}}^p(\boldsymbol{\zeta}),$$

since we only remove continuity conditions as we go from left to right. However, using the previous case and the fact that $\bar{\boldsymbol{\beta}} = \bar{\boldsymbol{\alpha}}$, we find that

$$S^p_{\boldsymbol{\beta}}(\boldsymbol{\zeta}) = S^p_{\bar{\boldsymbol{\beta}}}(\bar{\boldsymbol{\zeta}}) = S^p_{\bar{\boldsymbol{\alpha}}}(\bar{\boldsymbol{\zeta}}).$$

Hence $S^p_{\bar{\boldsymbol{\alpha}}}(\bar{\boldsymbol{\zeta}}) \subseteq S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta}) \subseteq S^p_{\bar{\boldsymbol{\alpha}}}(\bar{\boldsymbol{\zeta}})$ so the spaces must be equal. □

One of the main features of B-splines is that they allow us to construct a basis for any given space $S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta})$ by choosing an appropriate open knot vector. From Proposition 6 we see that we can assume $\alpha_i \leq p-1$ without any loss of generality.

Theorem 1 (Curry-Schoenberg [8])**.** *If we construct a knot vector* $\Xi = [\xi_1, \ldots, \xi_{n+p+1}]$ *by*

$$\Xi = \left[\overbrace{\zeta_1, \ldots, \zeta_1}^{p+1}, \overbrace{\zeta_2, \ldots, \zeta_2}^{p-\alpha_2}, \ldots, \overbrace{\zeta_i, \ldots, \zeta_i}^{p-\alpha_i}, \ldots, \overbrace{\zeta_{N-1}, \ldots, \zeta_{N-1}}^{p-\alpha_{N-1}}, \overbrace{\zeta_N, \ldots, \zeta_N}^{p+1} \right], \quad (2.2.3)$$

then

$$S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta}) = S_{p,\Xi}.$$

Proof. Indeed, $S_{p,\Xi} \subseteq S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta})$ since the B-splines spanning $S_{p,\Xi}$ satisfy the smoothness conditions of the space $S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta})$ (Proposition 5) and are piecewise polynomials of degree p (Proposition 3). But from (2.2.1) the dimension of $S_{p,\Xi}$ is $n = 2(p+1) + \sum_{i=2}^{N-1}(p-\alpha_i) - (p+1) = 1 + (N-1)p - \sum_{i=2}^{N-1}\alpha_i$, which is the same as the dimension of $S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta})$ by (2.2.2). Thus $S_{p,\Xi}$ is a subspace of $S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta})$ with the same dimension, which means that they must be the same space. □

2.2.2 Bivariate spline functions

Let $S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta})$ and $S^q_{\boldsymbol{\beta}}(\boldsymbol{\gamma})$ be given spline spaces. Using Theorem 1 we construct the open knot vectors $\Xi = [\xi_1, \ldots, \xi_{n+p+1}]$ and $\mathcal{H} = [\eta_1, \ldots, \eta_{m+q+1}]$ and we denote by $\{N_{i,p}\}_{i=1}^n$ and $\{M_{j,q}\}_{j=1}^m$ the B-splines of Ξ and $\mathcal{H}$ respectively. We can now define the tensor product B-splines by

$$N_{i,j;p,q} := N_{i,p} \otimes M_{j,q}, \qquad i = 1, \ldots, n \text{ and } j = 1, \ldots, m,$$

so for $\xi, \eta \in \mathbb{R}$ we have that $N_{i,j;p,q}(\xi, \eta) = N_{i,p}(\xi) M_{j,q}(\eta)$. In the same way as for the one dimensional case, a tensor product spline function v is a linear combination of the

basis functions, i.e. $v = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j} N_{i,j;p,q}$ where $c_{i,j} \in \mathbb{R}$. The space of all such linear combinations is the tensor product spline space defined by

$$S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma}) := S^{p}_{\boldsymbol{\alpha}}(\boldsymbol{\zeta}) \otimes S^{p}_{\boldsymbol{\beta}}(\boldsymbol{\gamma}) = \operatorname{span}\{N_{i,j;p,q}|_{\hat{\Omega}}\}_{i,j=1}^{n,m}, \tag{2.2.4}$$

where $\hat{\Omega} = [\xi_1, \xi_{n+p+1}) \times [\eta_1, \eta_{m+q+1})$. Such a tensor product construction will make the functions $\{N_{i,j;p,q}\}_{i,j=1}^{n,m}$ linearly independent on $\hat{\Omega}$, so we can note that

$$\dim S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma}) = \dim S^{p}_{\boldsymbol{\alpha}}(\boldsymbol{\zeta}) \dim S^{p}_{\boldsymbol{\beta}}(\boldsymbol{\gamma}).$$

The most important properties of the tensor product B-splines $\{N_{i,j;p,q}\}_{i,j=1}^{n,m}$ follow directly from the univariate case. Using Proposition 2 we can observe that the support of the basis function $N_{i,j;p,q}$ is the rectangle $[\xi_i, \xi_{i+p+1}] \times [\eta_j, \eta_{j+q+1}]$, and also that it is strictly positive on the interior of the support. It is also clear that the bivariate basis is a partition of unity as well. Indeed, for any $(\xi, \eta) \in [\xi_1, \xi_{n+p+1}) \times [\eta_1, \eta_{m+q+1})$ we have that

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

The continuity properties of the basis functions and their partial derivatives are also easy to deduce from the univariate case.

2.2.3 B-spline curves

For an integer $d \geq 2$ a B-spline curve, or a spline curve, in $\mathbb{R}^d$ is defined by the B-spline basis functions $\{N_{i,p}\}_{i=1}^{n}$, and corresponding *control points* $\{\mathbf{B}_i\}_{i=1}^{n}$, where $\mathbf{B}_i \in \mathbb{R}^d$. The spline curve is given parametrically by

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\,\mathbf{B}_i, \qquad \xi \in [\xi_1, \xi_{n+p+1}). \tag{2.2.5}$$

In other words, a spline curve is an element of the space $(S^{p}_{\boldsymbol{\alpha}}(\boldsymbol{\zeta}))^d$. It is clear form this definition that the spline curves are piecewise polynomial curves, and also the continuity properties follow directly from those of the B-spline basis functions. Linear interpolation of the control points gives the *control polygon*. In the context of FEA, the images of the knot intervals $[\xi_i, \xi_{i+1})$, $i = 1, \ldots, n+p$, are referred to as elements, so the knot locations define the element boundaries. We note that some elements might have zero measure in the case of repeated knots.

The quadratic spline curve of the knot vector $\Xi = [0, 0, 0, 1, 2, 3, 3, 4, 4, 4]$ with the control points

$$[\mathbf{B}_1, \ldots, \mathbf{B}_7] = \begin{bmatrix} 0 & 1 & 2 & 1.5 & 3.5 & 4.5 & 3 \\ 1 & -1 & -0.5 & 1.5 & 0 & 3 & 2 \end{bmatrix}$$

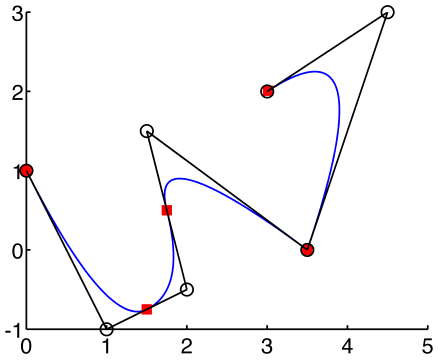

Figure 2.2.1: An example of a quadratic spline curve. The control polygon is shown in black and the knot locations are the red squares.

is shown in Figure 2.2.1, along with the control polygon. Since the knot vector is open, the curve is interpolatory at the first and the last knot. We also notice that the curve is C^1-continuous at the knots with multiplicity 1 and only C^0-continuous at the knot $\xi = 3$ which has multiplicity 2. This is in accordance with Proposition 5 which states that the basis functions (and hence any linear combination of these) is C^{p-r}-continuous at a knot with multiplicity r.

2.2.4 B-spline surfaces

For an integer $d \geq 2$ a tensor product B-spline surface in $\mathbb{R}^d$ is defined by the tensor product B-splines $\{N_{i,j;p,q}\}_{i,j=1}^{n,m}$, and a *control net* $\{\mathbf{B}_{i,j}\}_{i,j=1}^{n,m}$, where $\mathbf{B}_{i,j} \in \mathbb{R}^d$. The spline surface is given parametrically by

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} N_{i,j;p,q}(\xi,\eta)\,\mathbf{B}_{i,j}, \qquad (\xi,\eta) \in \hat{\Omega}, \tag{2.2.6}$$

where $\hat{\Omega} = [\xi_1, \xi_{n+p+1}) \times [\eta_1, \eta_{m+q+1})$. Thus a spline surface is an element of the space $(S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma}))^d$. What we now refer to as elements in a FEA setting are the images of the rectangles $[\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1})$ for $i = 1, \ldots, n+p$ and $j = 1, \ldots, m+q$, the elements boundaries are thus defined by the knot vectors. Figure 2.2.2 gives an example with polynomial orders $p = q = 2$, knot vectors $\Xi = [0,0,0,0.5,1,1,1]$ and $\mathcal{H} = [0,0,0,1,1,1]$, and a control net given by

$$[\mathbf{B}_1, \ldots, \mathbf{B}_{12}] = \begin{bmatrix} -1 & -1 & 1 & 1 & -1.5 & -1.5 & 1.5 & 1.5 & -2 & -2 & 2 & 2 \\ 0 & 1 & 1 & 0 & 0 & 2 & 2 & 0 & 0 & 2 & 2 & 0 \end{bmatrix},$$

where the control points are numbered by the scheme $A = n(j-1)+i$, so that $\mathbf{B}_A = \mathbf{B}_{i,j}$.

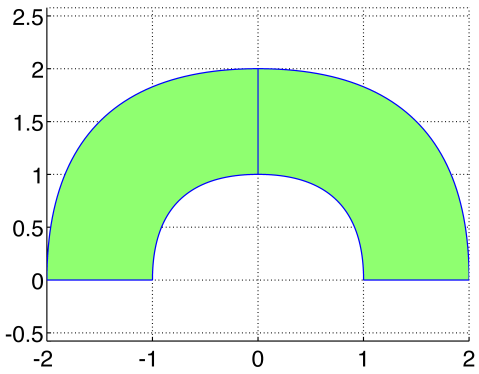

Figure 2.2.2: An example of a B-spline surface along with the grid defined by the knot vectors.

Finally we make the observation that we can consider a B-spline surface given by (2.2.6) as a linear combination of spline curves in one parameter as given by (2.2.5), where the coefficients depend on the other parameter. What we mean by this is that we can write

$$\mathbf{S}(\xi,\eta) = \sum_{j=1}^{m} M_{j,q}(\eta) \sum_{i=1}^{n} N_{i,p}(\xi)\,\mathbf{B}_{i,j} = \sum_{j=1}^{m} M_{j,q}(\eta)\,\mathbf{C}_j(\xi), \tag{2.2.7}$$

where each spline curve $\mathbf{C}_j$, $j = 1,\dots,m$, has control points $\{\mathbf{B}_{i,j}\}_{i=1}^{n}$, and similarly with the roles of ξ and η switched. This observation will be useful when we perform refinement on the B-spline geometry, because it means that we can use the univariate algorithms one direction at a time by refining each of the curves $\mathbf{C}_j$.

2.3 Spaces of spline derivatives

Before the construction of divergence-conforming B-spline spaces for the Stokes problem in Chapter 4, we need to consider the spaces that the derivatives of splines form. More specifically we consider the derivative operator for univariate splines and the divergence operator for spline vector fields.

2.3.1 The derivative operator

Let the spline spaces $S^p_{\boldsymbol{\alpha}} = S^p_{\boldsymbol{\alpha}}(\boldsymbol{\zeta})$ and $S^{p+1}_{\boldsymbol{\alpha}+1} = S^{p+1}_{\boldsymbol{\alpha}+1}(\boldsymbol{\zeta})$ constructed on the same partition $\boldsymbol{\zeta} = \{\zeta_i\}_{i=1}^{N}$ of the interval $[\zeta_1, \zeta_N)$ be given, where $\boldsymbol{\alpha}+1$ is defined by

$$\boldsymbol{\alpha}+1 = \{-1, \alpha_2+1, \dots, \alpha_i+1, \dots, \alpha_{N-1}+1, -1\}.$$

If we look at the knot vectors giving these spaces, it becomes clear from (2.2.3) that the only difference between them is that the first and the last knot are repeated one more time in the knot vector of $S^{p+1}_{\boldsymbol{\alpha}+1}$. Since $S^{p+1}_{\boldsymbol{\alpha}+1}$ has a knot vector with two more elements and a polynomial degree one more than $S^p_{\boldsymbol{\alpha}}$, it follows that $S^{p+1}_{\boldsymbol{\alpha}+1}$ will have one more basis function than $S^p_{\boldsymbol{\alpha}}$, so

$$\dim S^{p+1}_{\boldsymbol{\alpha}+1} = \dim S^p_{\boldsymbol{\alpha}} + 1. \tag{2.3.1}$$

This can also be seen by using the formula (2.2.2) for the two spaces.

Now if we let v be a spline $v \in S^{p+1}_{\boldsymbol{\alpha}+1}$ it is clear that the derivative of v will be an element of $S^p_{\boldsymbol{\alpha}}$, since $\frac{dv}{d\xi}$ will be a piecewise polynomial function of one degree less than v and with one less continuous derivative at each point ζ_i where $2 \leq i \leq N-1$. Thus the derivative operator is a linear map $\frac{d}{d\xi} : S^{p+1}_{\boldsymbol{\alpha}+1} \to S^p_{\boldsymbol{\alpha}}$, and it is in fact surjective as stated in the following proposition.

Proposition 7. *The image of the derivative operator $\frac{d}{d\xi} : S^{p+1}_{\boldsymbol{\alpha}+1} \to S^p_{\boldsymbol{\alpha}}$ is the entire codomain $S^p_{\boldsymbol{\alpha}}$, i.e.*

$$\frac{d}{d\xi} S^{p+1}_{\boldsymbol{\alpha}+1} := \left\{ \frac{dv}{d\xi} : v \in S^{p+1}_{\boldsymbol{\alpha}+1} \right\} = S^p_{\boldsymbol{\alpha}}.$$

Proof. We have already established that $\frac{dv}{d\xi} \in S^p_{\boldsymbol{\alpha}}$ for any $v \in S^{p+1}_{\boldsymbol{\alpha}+1}$, so $\frac{d}{d\xi} S^{p+1}_{\boldsymbol{\alpha}+1} \subseteq S^p_{\boldsymbol{\alpha}}$. Thus the result can be shown by comparing the dimensions of the spaces. From linear algebra we know that the dimension of the space $S^{p+1}_{\boldsymbol{\alpha}+1}$ must be equal to the sum of the dimensions of the kernel and the image of the derivative operator, so we have the relationship

$$\dim S^{p+1}_{\boldsymbol{\alpha}+1} = \dim \left\{ v \in S^{p+1}_{\boldsymbol{\alpha}+1} : \frac{dv}{d\xi} = 0 \right\} + \dim \left\{ \frac{d}{d\xi} S^{p+1}_{\boldsymbol{\alpha}+1} \right\}.$$

The only splines v that satisfy $\frac{dv}{d\xi} = 0$ are the piecewise constants, but $v \in S^{p+1}_{\boldsymbol{\alpha}+1}$ implies that v is C^0-continuous because $\alpha_i \geq -1$ gives $\alpha_i + 1 \geq 0$. Hence the kernel $\left\{ v \in S^{p+1}_{\boldsymbol{\alpha}+1} : \frac{dv}{d\xi} = 0 \right\}$ consists of the constants on the interval $[\zeta_1, \zeta_N)$ and has thus dimension one, which gives

$$\dim S^{p+1}_{\boldsymbol{\alpha}+1} = 1 + \dim \left\{ \frac{d}{d\xi} S^{p+1}_{\boldsymbol{\alpha}+1} \right\}.$$

Comparing this to the result (2.3.1), it is clear that $\frac{d}{d\xi} S^{p+1}_{\boldsymbol{\alpha}+1}$ has the same dimension as $S^p_{\boldsymbol{\alpha}}$ and they must therefore be the same space. □

Let the B-spline bases of the spaces $S^p_{\boldsymbol{\alpha}}$ and $S^{p+1}_{\boldsymbol{\alpha}+1}$ be denoted by $\left\{ N^{\boldsymbol{\alpha}+1}_{i,p+1} \right\}_{i=1}^{n+1}$ and $\left\{ N^{\boldsymbol{\alpha}}_{i,p} \right\}_{i=1}^{n}$ respectively. If we have a spline $v \in S^{p+1}_{\boldsymbol{\alpha}+1}$ given by the coefficients $\{c_i\}_{i=1}^{n+1}$,

we might be interested in the explicit representation of $\frac{dv}{d\xi}$ as a spline in $S^p_{\boldsymbol{\alpha}}$ and find coefficients $\{\tilde{c}_j\}_{j=1}^n$ such that for any $\xi \in \mathbb{R}$ we have that

$$\sum_{j=1}^{n} \tilde{c}_j N^{\boldsymbol{\alpha}}_{j,p}(\xi) = \sum_{j=1}^{n+1} c_j \frac{d}{d\xi} N^{\boldsymbol{\alpha}+1}_{j,p+1}(\xi). \tag{2.3.2}$$

By requiring that (2.3.2) holds at n points ξ_i^*, $i = 1, \ldots, n$, we obtain the following linear system of equations for the coefficients $\tilde{c}_j$:

$$\sum_{j=1}^{n} \tilde{c}_j N^{\boldsymbol{\alpha}}_{j,p}(\xi_i^*) = \sum_{j=1}^{n+1} c_j \frac{d}{d\xi} N^{\boldsymbol{\alpha}+1}_{j,p+1}(\xi_i^*), i = 1, \ldots, n. \tag{2.3.3}$$

Defining the matrices $\mathbf{A} = \left\{N^{\boldsymbol{\alpha}}_{j,p}(\xi_i^*)\right\}_{i,j=1}^{n,n}$ and $\mathbf{B} = \left\{\frac{d}{d\xi} N^{\boldsymbol{\alpha}+1}_{j,p+1}(\xi_i^*)\right\}_{i,j=1}^{n,n+1}$, and the coefficient vectors $\tilde{\mathbf{c}} = \{\tilde{c}_j\}_{j=1}^n$ and $\mathbf{c} = \{c_j\}_{j=1}^{n+1}$, we can write (2.3.3) in matrix form as

$$\mathbf{A}\tilde{\mathbf{c}} = \mathbf{B}\mathbf{c}. \tag{2.3.4}$$

In order for the matrix $\mathbf{A}$ to be nonsingular we have to choose the interpolation points ξ_i^* in such a way that $\xi_i < \xi_i^* < \xi_{i+p+1}$ for $i = 1, \ldots, n$, however $\xi_i^* = \xi_i$ is allowed whenever $\xi_i = \xi_{i+p}$ [22, Chap. 10]. Here the ξ_i are the knots of the knot vector $\Xi^{p+1}_{\boldsymbol{\alpha}+1}$ for the space $S^{p+1}_{\boldsymbol{\alpha}+1}$. This can be achieved by letting the ξ_i^* be the n first knot averages of $\Xi^{p+1}_{\boldsymbol{\alpha}+1}$ defined by $\xi_i^* = (\xi_{i+1} + \ldots + \xi_{i+p+1})/(p+1)$ for $i = 1, \ldots, n$. With this choice we obtain from (2.3.4) that

$$\tilde{\mathbf{c}} = \mathbf{A}^{-1}\mathbf{B}\mathbf{c}.$$

This means that the matrix $\mathbf{D} \in \mathbb{R}^{n\times(n+1)}$ defined by

$$\mathbf{D} = \mathbf{A}^{-1}\mathbf{B} \tag{2.3.5}$$

is the matrix representation of the differentiation operator $\frac{d}{d\xi} : S^{p+1}_{\boldsymbol{\alpha}+1} \to S^p_{\boldsymbol{\alpha}}$.

2.3.2 The divergence operator

Let $S^p_{\boldsymbol{\alpha}}$ and $S^{p+1}_{\boldsymbol{\alpha}+1}$ be the same as above, and let $S^q_{\boldsymbol{\beta}} = S^p_{\boldsymbol{\beta}}(\boldsymbol{\gamma})$ and $S^{q+1}_{\boldsymbol{\beta}+1} = S^{p+1}_{\boldsymbol{\beta}+1}(\boldsymbol{\gamma})$ be constructed in the same manner dependent on the regularity sequence $\boldsymbol{\beta}$, the polynomial degree q and the partition $\boldsymbol{\gamma} = \{\gamma_i\}_{i=1}^M$ of the interval $[\gamma_1, \gamma_M)$. We denote by $\left\{N^{\boldsymbol{\alpha}}_{i,p}\right\}_{i=1}^n$, $\left\{N^{\boldsymbol{\alpha}+1}_{i,p+1}\right\}_{i=1}^{n+1}$, $\left\{M^{\boldsymbol{\beta}}_{j,q}\right\}_{j=1}^m$ and $\left\{M^{\boldsymbol{\beta}+1}_{j,q+1}\right\}_{j=1}^{m+1}$ the bases of $S^p_{\boldsymbol{\alpha}}$, $S^{p+1}_{\boldsymbol{\alpha}+1}$, $S^q_{\boldsymbol{\beta}}$ and $S^{q+1}_{\boldsymbol{\beta}+1}$ respectively. Now consider the linear space of spline vector fields given by

$$S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1} := S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma}) \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}(\boldsymbol{\zeta},\boldsymbol{\gamma}),$$

where the bivariate spline spaces are defined by (2.2.4). Like for the derivative operator we determine the image of the divergence operator in the following proposition.

Proposition 8. *The image of the divergence operator* $\operatorname{div} : S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1} \to S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ *is the entire codomain* $S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$*, i.e.*

$$\operatorname{div}\left(S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}\right) := \left\{\operatorname{div} \mathbf{v} : \mathbf{v} \in S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}\right\} = S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}.$$

Proof. That $\operatorname{div}\left(S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}\right) \subseteq S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is clear by the same reasoning we did in the one dimensional case. We can therefore prove the statement by showing that for any $w \in S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ there exists a $\mathbf{v} \in S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}$ such that $\operatorname{div} \mathbf{v} = w$. Indeed if $w \in S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ we can express w in terms of the basis functions by

$$\begin{aligned} w(\xi,\eta) &= \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij} N^{\boldsymbol{\alpha}}_{i,p}(\xi)\, M^{\beta}_{j,q}(\eta) \\ &= \sum_{j=1}^{m} M^{\beta}_{j,q}(\eta) \sum_{i=1}^{n} \frac{c_{ij}}{2} N^{\boldsymbol{\alpha}}_{i,p}(\xi) + \sum_{i=1}^{n} N^{\boldsymbol{\alpha}}_{i,p}(\xi) \sum_{j=1}^{m} \frac{c_{ij}}{2} M^{\beta}_{j,q}(\eta). \end{aligned}$$

Now $\sum_{i=1}^{n} \frac{c_{ij}}{2} N^{\boldsymbol{\alpha}}_{i,p}(\xi)$ will be a spline in $S^{p}_{\boldsymbol{\alpha}}$ for all indices j, and $\sum_{j=1}^{m} \frac{c_{ij}}{2} M^{\beta}_{j,q}(\eta)$ will be a spline in S^{q}_{β} for all i. Thus applying Proposition 7 in both parametric directions and we obtain

$$\begin{aligned} w(\xi,\eta) &= \sum_{j=1}^{m} M^{\boldsymbol{\beta}}_{j,q}(\eta) \frac{\partial}{\partial \xi} \sum_{i=1}^{n+1} \hat{c}_{ij} N^{\boldsymbol{\alpha}+1}_{i,p+1}(\xi) + \sum_{i=1}^{n} N^{\boldsymbol{\alpha}}_{i,p}(\xi) \frac{\partial}{\partial \eta} \sum_{j=1}^{m+1} \tilde{c}_{ij} M^{\boldsymbol{\beta}+1}_{j,q+1}(\eta) \\ &= \frac{\partial}{\partial \xi}\left(\sum_{i=1}^{n+1}\sum_{j=1}^{m} \hat{c}_{ij} N^{\boldsymbol{\alpha}+1}_{i,p+1}(\xi)\, M^{\boldsymbol{\beta}}_{j,q}(\eta)\right) + \frac{\partial}{\partial \eta}\left(\sum_{i=1}^{n}\sum_{j=1}^{m+1} \tilde{c}_{ij} N^{\boldsymbol{\alpha}}_{i,p}(\xi)\, M^{\boldsymbol{\beta}+1}_{j,q+1}(\eta)\right) \\ &= \operatorname{div}\left(\sum_{i=1}^{n+1}\sum_{j=1}^{m} \hat{c}_{ij} N^{\boldsymbol{\alpha}+1}_{i,p+1}(\xi)\, M^{\boldsymbol{\beta}}_{j,q}(\eta)\, \mathbf{e}_1 + \sum_{i=1}^{n}\sum_{j=1}^{m+1} \tilde{c}_{ij} N^{\boldsymbol{\alpha}}_{i,p}(\xi)\, M^{\boldsymbol{\beta}+1}_{j,q+1}(\eta)\, \mathbf{e}_2\right). \end{aligned}$$

Here $\mathbf{e}_1$ and $\mathbf{e}_2$ are the usual basis vectors of $\mathbb{R}^2$ so that

$$\left\{N^{\boldsymbol{\alpha}+1}_{i,p+1} M^{\boldsymbol{\beta}}_{j,q} \mathbf{e}_1\right\}_{i,j=1}^{n+1,m} \cup \left\{N^{\boldsymbol{\alpha}}_{i,p} M^{\boldsymbol{\beta}+1}_{j,q+1} \mathbf{e}_2\right\}_{i,j=1}^{n,m+1}$$

is a basis for the space $S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}$, and we have found a $\mathbf{v} \in S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}$ such that $\operatorname{div} \mathbf{v} = w$ which completes the proof as w was arbitrary. □

We can also for any $\mathbf{v} \in S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}} \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}$ find an explicit representation of $\operatorname{div} \mathbf{v}$ as an element of $S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ given by the coefficients $\{w_{ij}\}_{i,j=1}^{n,m}$. This is done by writing $\mathbf{v}$ in terms of coefficients $\{u_{ij}\}_{i,j=1}^{n+1,m}$ and $\{v_{ij}\}_{i,j=1}^{n,m+1}$ as

$$\mathbf{v}(\xi,\eta) = \sum_{i=1}^{n+1}\sum_{j=1}^{m} u_{ij} N^{\boldsymbol{\alpha}+1}_{i,p+1}(\xi)\, M^{\boldsymbol{\beta}}_{j,q}(\eta)\, \mathbf{e}_1 + \sum_{i=1}^{n}\sum_{j=1}^{m+1} v_{ij} N^{\boldsymbol{\alpha}}_{i,p}(\xi)\, M^{\boldsymbol{\beta}+1}_{j,q+1}(\eta)\, \mathbf{e}_2,$$

which gives

$$\begin{aligned}\mathrm{div}\mathbf{v} &= \sum_{j=1}^{m} M_{j,q}^{\boldsymbol{\beta}}(\eta)\frac{\partial}{\partial\xi}\sum_{i=1}^{n+1} u_{ij}N_{i,p+1}^{\boldsymbol{\alpha}+1}(\xi) + \sum_{i=1}^{n} N_{i,p}^{\boldsymbol{\alpha}}(\xi)\frac{\partial}{\partial\eta}\sum_{j=1}^{m+1} v_{ij}M_{j,q+1}^{\boldsymbol{\beta}+1}(\eta)\\ &= \sum_{j=1}^{m} M_{j,q}^{\boldsymbol{\beta}}(\eta)\sum_{i=1}^{n}\tilde{u}_{ij}N_{i,p}^{\boldsymbol{\alpha}}(\xi) + \sum_{i=1}^{n} N_{i,p}^{\boldsymbol{\alpha}}(\xi)\sum_{j=1}^{m}\tilde{v}_{ij}M_{j,q}^{\boldsymbol{\beta}}(\eta)\\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}(\tilde{u}_{ij}+\tilde{v}_{ij})N_{i,p}^{\boldsymbol{\alpha}}(\xi)M_{j,q}^{\boldsymbol{\beta}}(\eta).\end{aligned}$$

Here we have used the method of Section 2.3.1 for each j in the ξ-direction and for each i in the η-direction. This means that if $\mathbf{D}_\xi$ and $\mathbf{D}_\eta$ are the differentiation matrices defined by (2.3.5) for the spaces $S_{\boldsymbol{\alpha}+1}^{p+1}$ and $S_{\boldsymbol{\beta}+1}^{q+1}$ respectively, and we organize the coefficients in the vectors

$$\begin{aligned}\mathbf{W} &= \begin{bmatrix} w_{1,1} & \cdots & w_{n,1} & \cdots & w_{1,m} & \cdots & w_{n,m}\end{bmatrix}^T,\\ \mathbf{U} &= \begin{bmatrix} u_{1,1} & \cdots & u_{n+1,1} & \cdots & u_{1,m} & \cdots & u_{n+1,m}\end{bmatrix}^T,\\ \mathbf{V} &= \begin{bmatrix} v_{1,1} & \cdots & v_{n,1} & \cdots & v_{1,m+1} & \cdots & v_{n,m+1}\end{bmatrix}^T,\end{aligned}$$

then we have the relation

$$\mathbf{W} = \begin{bmatrix}\mathbf{I}_m\otimes\mathbf{D}_\xi & \mathbf{D}_\eta\otimes\mathbf{I}_n\end{bmatrix}\begin{bmatrix}\mathbf{U}\\ \mathbf{V}\end{bmatrix}.$$

Here $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ and $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ are identity matrices, and $\otimes$ denotes the Kronecker tensor product. Thus the matrix $\mathbf{D} \in \mathbb{R}^{(nm)\times((n+1)m+n(m+1))}$ defined by

$$\mathbf{D} = \begin{bmatrix}\mathbf{I}_m\otimes\mathbf{D}_\xi & \mathbf{D}_\eta\otimes\mathbf{I}_n\end{bmatrix} \tag{2.3.6}$$

is the matrix representation of the divergence operator $\mathrm{div} : S_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}}^{p+1,q} \times S_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}^{p,q+1} \to S_{\boldsymbol{\alpha},\boldsymbol{\beta}}^{p,q}$.

2.4 Refinement

In isogeometric analysis the B-spline or NURBS basis functions are used both to represent the geometry and as a basis for the solution space in which we seek an approximation to the solution. In order to get good approximations it is essential that we are able to refine the basis so that the finite dimensional solution space approaches the actual solution space. One of the strengths and key features of isogeometric analysis is that refinement can be performed while keeping the geometry exact. In ordinary finite element analysis the refinement methods are h-refinement where the size of the elements is

decreased, and p-refinement where the order of the basis is increased. In isogeometric analysis the tools for performing refinement is knot insertion and order elevation. These operations provide more flexibility than simple h- and p-refinement, we can for example control the continuity of the basis at the knots.

2.4.1 Knot insertion for B-spline curves

Let $\mathbf{C}$ be a spline curve given by (2.2.5) defined by the knot vector $\Xi = [\xi_1, \ldots, \xi_{n+p+1}]$. One way to enrich the basis by which the curve is represented is to insert new knots into Ξ obtaining a new knot vector $\bar{\Xi} = \left[\bar{\xi}_1, \ldots, \bar{\xi}_{m+p+1}\right]$, $m \geq n$, which contains all elements found in Ξ. We assume as before that Ξ is open, and for our purpose we can also assume that the new knot vector $\bar{\Xi}$ contains the same $p+1$ knots as Ξ at each end. We now want to represent the curve using the new knot vector $\bar{\Xi}$, and thereby the new basis functions $\bar{N}_{j,p}$, $j = 1, \ldots, m$. This is achieved by writing the original B-splines as elements of the space spanned by the new basis functions, i.e.

$$N_{i,p} = \sum_{j=1}^{m} \alpha_{i,p}(j)\, \bar{N}_{j,p}, \tag{2.4.1}$$

where the $\alpha_{i,p}(j)$, $i = 1, \ldots, n$, $j = 1, \ldots, m$, are the spline coefficients. This can be done because the new spline space contains the original space when the new knot vector contains all elements of the original knot vector. Inserting (2.4.1) into (2.2.5) we have for any $\xi \in [\xi_1, \xi_{n+p+1}) = \left[\bar{\xi}_1, \bar{\xi}_{m+p+1}\right)$ that

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} \alpha_{j,p}(i)\, \bar{N}_{j,p}(\xi) \right) \mathbf{B}_i = \sum_{j=1}^{m} \bar{N}_{j,p}(\xi) \sum_{i=1}^{n} \alpha_{i,p}(j)\, \mathbf{B}_i = \sum_{j=1}^{m} \bar{N}_{j,p}(\xi)\, \bar{\mathbf{B}}_j.$$

Thus, choosing new control points

$$\bar{\mathbf{B}}_j = \sum_{i=1}^{n} \alpha_{i,p}(j)\, \mathbf{B}_i, \tag{2.4.2}$$

we have a representation of the curve by the B-splines of new knot vector $\bar{\Xi}$. Since the parameter ξ is the same in both representations and the parameter space is the same due to the common knots at the ends, we have that the curve remains unchanged both geometrically and parametrically.

The algorithm for computing the new control points $\bar{\mathbf{B}}_j$ is referred to as the Oslo algorithm [22, Ch. 4.2]. This algorithm is based on a recurrence relation for the coefficients $\alpha_{j,p}(i)$ similar to the recurrence relation in Definition 2.1 of the B-splines. We have that

$$\alpha_{i,p}(j) = \frac{\bar{\xi}_{j+p} - \xi_i}{\xi_{i+p} - \xi_i} \alpha_{i,p-1}(j) + \frac{\xi_{i+p+1} - \bar{\xi}_{j+p}}{\xi_{i+p+1} - \xi_{i+1}} \alpha_{i+1,p-1}(j),$$

starting with $\alpha_{i,0}(j) = N_{i,0}\left(\bar{\xi}_j\right)$. A proof of this relation can be found in [25]. In [22, Chap 4.2] the Oslo algorithm is obtained by adapting the B-spline evaluation algorithms to knot insertion. With this algorithm we can compute the knot insertion matrix $\mathbf{A} = \{\alpha_{i,p}(j)\}_{j,i=1}^{m,n} \in \mathbb{R}^{m\times n}$ from the knot vector Ξ to the new knot vector $\bar{\Xi}$. Then if we organize the control points in matrices $\mathcal{B} = [\mathbf{B}_1, \ldots, \mathbf{B}_n]^T \in \mathbb{R}^{n\times d}$ and $\bar{\mathcal{B}} = \left[\bar{\mathbf{B}}_1, \ldots, \bar{\mathbf{B}}_m\right]^T \in \mathbb{R}^{m\times d}$, where d is the dimension, we obtain from (2.4.2) the relation

$$\bar{\mathcal{B}} = \mathbf{A}\mathcal{B}. \tag{2.4.3}$$

Figure 2.4.1 and Figure 2.4.2 give an example of knot insertion performed on the spline curve from Figure 2.2.1. The knots 1.5 and 2.5 are inserted with multiplicity one, extending the original knot vector $\Xi = [0, 0, 0, 1, 2, 3, 3, 4, 4, 4]$ to the new knot vector $\bar{\Xi} = [0, 0, 0, 1, 1.5, 2, 2.5, 3, 3, 4, 4, 4]$. Figure 2.4.1 shows the original spline curve along with the knot locations and the control polygon, and the original basis functions are also shown. Figure 2.4.2 gives the same plots after the knot insertion. We observe that the new basis is $C^{p-1} = C^1$-continuous at the inserted knots, as expected when the knots are inserted with multiplicity one.

2.4.2 Knot insertion for B-spline surfaces

For knot insertion on a tensor product B-spline surface we use the observation made at the end of Section 2.2.4, that we can view the spline surface as a linear combination of spline curves. If we for instance want to change the knot vector Ξ in the ξ-direction to $\bar{\Xi}$ by inserting knots, we can express the B-spline surface by (2.2.7) and perform knot insertion for each of the curves $\mathbf{C}_j$. This gives

$$\begin{aligned}\mathbf{S}(\xi,\eta) &= \sum_{j=1}^{m} M_{j,q}(\eta) \sum_{i=1}^{n} N_{i,p}(\xi)\,\mathbf{B}_{i,j} = \sum_{j=1}^{m} M_{j,q}(\eta) \sum_{\hat{i}=1}^{\hat{n}} \bar{N}_{\hat{i},p}(\xi)\,\bar{\mathbf{B}}_{\hat{i},j} \\ &= \sum_{\hat{i}=1}^{\hat{n}} \sum_{j=1}^{m} \bar{N}_{\hat{i},p}(\xi)\, M_{j,q}(\eta)\,\bar{\mathbf{B}}_{\hat{i},j},\end{aligned}$$

where the new control net $\left\{\bar{\mathbf{B}}_{\hat{i},j}\right\}$ is obtained from (2.4.3) as

$$\left[\bar{\mathbf{B}}_{1,j}, \ldots, \bar{\mathbf{B}}_{\hat{n},j}\right]^T = \mathbf{A}_\Xi [\mathbf{B}_{1,j}, \ldots, \mathbf{B}_{n,j}]^T, \quad j = 1, \ldots, m. \tag{2.4.4}$$

Here $\mathbf{A}_\Xi$ is the knot insertion matrix from Ξ to $\bar{\Xi}$. In a similar fashion we can insert knots into $\mathcal{H}$ getting the new knot vector $\bar{\mathcal{H}}$ and the representation

$$\mathbf{S}(\xi,\eta) = \sum_{\hat{i}=1}^{\hat{n}} \sum_{\hat{j}=1}^{\hat{m}} \bar{N}_{\hat{i},p}(\xi)\,\bar{M}_{\hat{j},q}(\eta)\,\hat{\mathbf{B}}_{\hat{i},\hat{j}}, \tag{2.4.5}$$

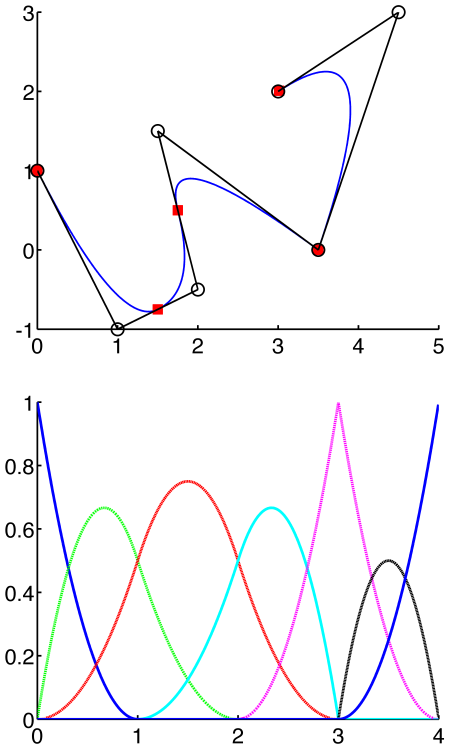

Figure 2.4.1: Knot insertion: The original spline curve with knot locations (squares) and control points (circles). The original basis functions are shown beneath.

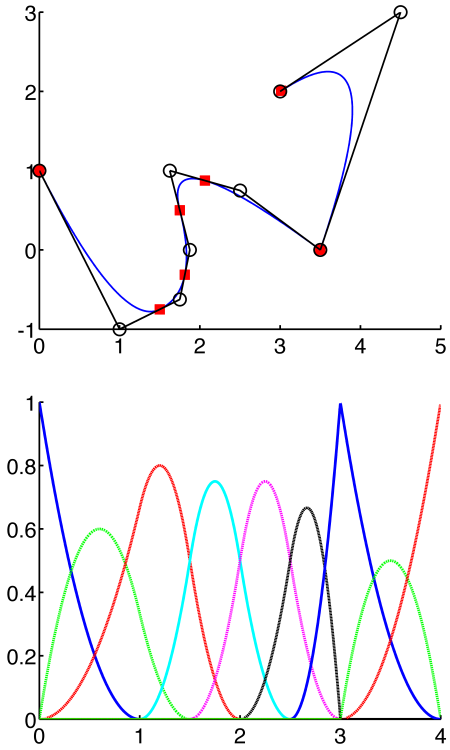

Figure 2.4.2: Knot insertion: The refined spline curve with knot locations (squares) and control points (circles). The refined basis functions are shown beneath.

$$\left[\hat{\mathbf{B}}_{\hat{i},1},\ldots,\hat{\mathbf{B}}_{\hat{i},\hat{m}}\right]^T = \mathbf{A}_{\mathcal{H}}\left[\bar{\mathbf{B}}_{\hat{i},1},\ldots,\bar{\mathbf{B}}_{\hat{i},m}\right]^T, \quad i = 1,\ldots,\hat{n}. \tag{2.4.6}$$

Here we have performed the knot insertion first in the ξ-direction and then in the η-direction, but if we did it the other way around we would get the same result, a consequence of the linear independence of the B-splines. Indeed, if we started in the η-direction we would get a representation of the same surface by

$$\mathbf{S}(\xi,\eta) = \sum_{\hat{i}=1}^{\hat{n}}\sum_{\hat{j}=1}^{\hat{m}} \bar{N}_{\hat{i},p}(\xi)\,\bar{M}_{\hat{j},q}(\eta)\,\tilde{\mathbf{B}}_{\hat{i},\hat{j}},$$

where the B-spline basis functions would be the same as in (2.4.5) because the knot vectors and polynomial orders are the same. Hence

$$\sum_{\hat{i}=1}^{\hat{n}} \bar{N}_{\hat{i},p}(\xi)\sum_{\hat{j}=1}^{\hat{m}} \bar{M}_{\hat{j},q}(\eta)\left(\hat{\mathbf{B}}_{\hat{i},\hat{j}} - \tilde{\mathbf{B}}_{\hat{i},\hat{j}}\right) = 0,$$

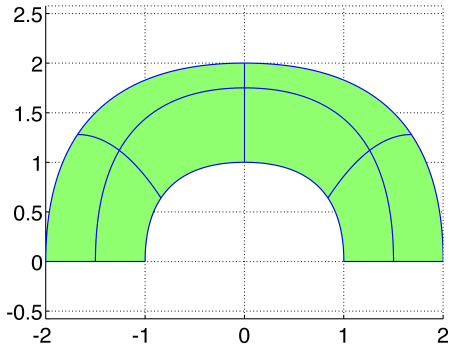

Figure 2.4.3: The B-spline surface from Figure 2.2.2 after inserting simple knots at 0.2 and 0.8 in Ξ, and at 0.5 in $\mathcal{H}$.

and the linear independence of the $\bar{N}_{\hat{i},p}$ gives that $\sum_{\hat{j}=1}^{\hat{m}} \bar{M}_{\hat{j},q}(\eta)\left(\hat{\mathbf{B}}_{\hat{i},\hat{j}} - \tilde{\mathbf{B}}_{\hat{i},\hat{j}}\right) = 0$ for all $\hat{i}$, which by the linear independence of the $\bar{M}_{\hat{j},q}$ yields $\hat{\mathbf{B}}_{\hat{i},\hat{j}} - \tilde{\mathbf{B}}_{\hat{i},\hat{j}} = 0$ for all $\hat{i}$ and $\hat{j}$. This shows that the resulting control net would be the same in both cases.

Figure 2.4.3 shows the grid of the B-spline surface from Figure 2.2.2 after knot insertion, the surface is the same as before, but the basis used to express it is refined.

2.4.3 Order elevation for B-spline curves

Another way to enrich the basis by which a curve given by (2.2.5) is represented is to elevate the order or degree of the basis functions. This is done by finding a new knot vector $\bar{\Xi}$ and new control points $\mathbf{B}_j$, $j = 1, \ldots, \bar{n}$, such that we can write

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi)\,\mathbf{B}_i = \sum_{j=1}^{\bar{n}} N_{j,p+t}(\xi)\,\bar{\mathbf{B}}_j = \bar{\mathbf{C}}(\xi), \tag{2.4.7}$$

and thus obtain the curve $\bar{\mathbf{C}}$ which is geometrically and parametrically identical to $\mathbf{C}$ but represented by B-spline basis functions of degree $p + t$, $t \in \mathbb{N}$. The new knot vector contains no new knot values, but the existing knot values will have to be repeated in order to keep the curve at the same level of continuity. The theory behind order elevation, or degree elevation, is explained in [24] and an algorithm for performing order elevation is given. The algorithm uses an approach where the spline curve is decomposed into several Bézier curves which are all order elevated, and finally excess knots are removed.

Figure 2.4.4 and Figure 2.4.5 give an example of order elevation performed on the spline curve from Figure 2.2.1. The order is increased from $p = 2$ to $p = 3$. Figure 2.4.1 shows the original spline curve along with the knot locations and the control polygon, and the original basis functions are also shown. Figure 2.4.5 gives the same plots after

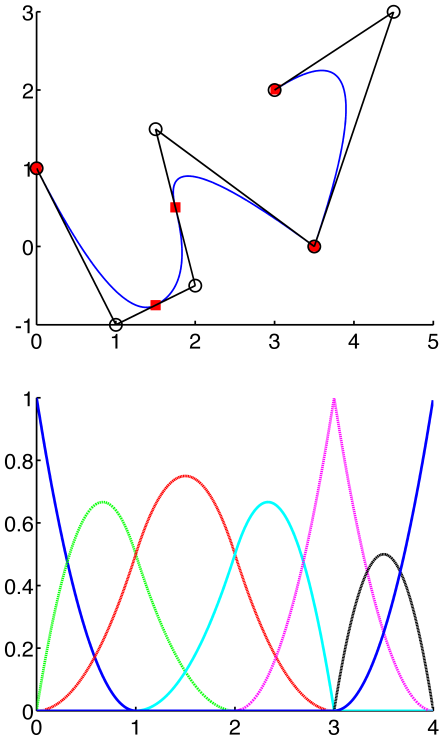

Figure 2.4.4: Order elevation: The original spline curve with knot locations (squares) and control points (circles). The original basis functions are shown beneath.

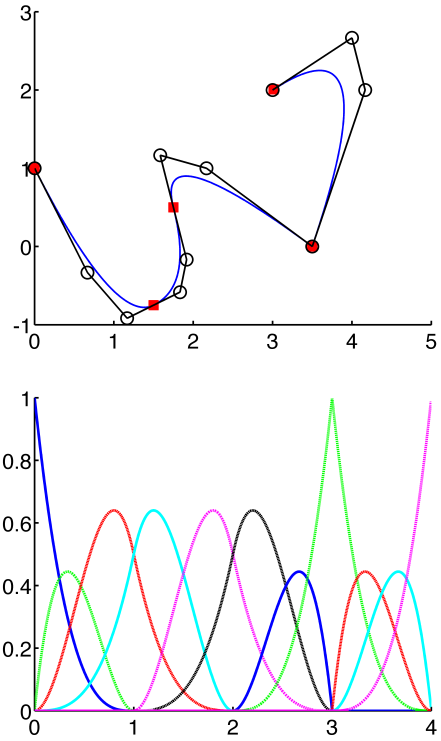

Figure 2.4.5: Order elevation: The refined spline curve with knot locations (squares) and control points (circles). The refined basis functions are shown beneath.

the order has been increased. We see from these figures that the continuity properties of the basis are preserved.

2.4.4 Order elevation for B-spline surfaces

The order elevation of a tensor product B-spline surface is derived in a similar fashion as the knot insertion for the surface, using the observation at the end of Section 2.2.4. We express the B-spline surface by (2.2.7) and perform order elevation for each of the curves $\mathbf{C}_j$. Using (2.4.7) we obtain

$$\begin{aligned}\mathbf{S}(\xi,\eta) &= \sum_{j=1}^{m} M_{j,q}(\eta) \sum_{i=1}^{n} N_{i,p}(\xi)\,\mathbf{B}_{i,j} = \sum_{j=1}^{m} M_{j,q}(\eta) \sum_{\hat{i}=1}^{\hat{n}} N_{\hat{i},p+t}(\xi)\,\bar{\mathbf{B}}_{\hat{i},j} \\ &= \sum_{\hat{i}=1}^{\hat{n}} \sum_{j=1}^{m} N_{\hat{i},p+t}(\xi)\, M_{j,q}(\eta)\,\bar{\mathbf{B}}_{\hat{i},j},\end{aligned}$$

where $t \geq 0$ is the increase in degree, and if we increase the degree in the η-direction by $s \geq 0$ in the same way, we get

$$\mathbf{S}(\xi, \eta) = \sum_{\hat{i}=1}^{\hat{n}} \sum_{\hat{j}=1}^{\hat{m}} N_{\hat{i},p+t}(\xi) M_{\hat{j},q+s}(\eta) \hat{\mathbf{B}}_{\hat{i},\hat{j}}.$$

Again the order in which we perform the order elevation does not matter. We end up with the same knot vectors and polynomial orders, hence the same B-spline basis functions, and by the uniqueness of the tensor product B-spline representation the control net is the same, as was shown when dealing with knot insertion.

2.5 NURBS

Using only spline geometries we are not able to represent important geometric objects such as circles and other conic sections exactly. A generalization of the B-spline basis functions that solves this problem is given by the Non-Uniform Rational B-Spline (NURBS) basis functions.

2.5.1 NURBS basis functions

The NURBS basis functions are defined as follows:

Definition 2.4. *Given a polynomial degree p, a knot vector $\Xi = [\xi_1, \ldots, \xi_{n+p+1}]$ and a sequence of positive real numbers $\{w_i\}_{i=1}^n$, we define the i^{th} Non-Uniform Rational B-Spline (NURBS) by*

$$R_i^p(\xi) := \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}}, \qquad \forall \xi \in [\xi_1, \xi_{n+p+1}) \tag{2.5.1}$$

where the $N_{i,p}$ is the i^{th} B-spline of degree p as defined in Definition 2.1. A number w_i is referred to as the i^{th} weight.

Since the B-splines are piecewise polynomials, it is clear that the NURBS basis functions are piecewise rational functions. Choosing all the weights equal to w, and assuming the knot vector to be open, we note that for any $\xi \in [\xi_1, \xi_{n+p+1})$

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

since the B-spline basis is a partition of unity (Proposition 4). The NURBS basis functions are thus a generalization of the B-splines. Like the B-spline basis, the NURBS basis is also a partition of unity:

$$\sum_{i=1}^n R_i^p(\xi) = \sum_{i=1}^n \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}} = \frac{\sum_{i=1}^n N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}} = 1, \qquad \forall \xi \in [\xi_1, \xi_{n+p+1}).$$

Finally we note that the derivatives of the NURBS basis functions can be calculated from the derivatives of the B-splines. Differentiating (2.5.1) using the quotient rule we obtain

$$\frac{d}{d\xi}R_i^p(\xi) = w_i \frac{\frac{d}{d\xi}N_{i,p}(\xi)\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi)\, w_{\hat{i}} - N_{i,p}(\xi)\sum_{\hat{i}=1}^{n}\frac{d}{d\xi}N_{\hat{i},p}(\xi)\, w_{\hat{i}}}{\left(\sum_{\hat{i}=1}^{n} N_{\hat{i},p}(\xi)\, w_{\hat{i}}\right)^2},$$

and the derivatives of the B-splines can be calculated using (2.1.9).

In the same way as we did for B-splines we can define tensor product NURBS basis functions by

$$R_{i,j}^{p,q}(\xi,\eta) := R_i^p(\xi)\, R_j^q(\eta) = \frac{N_{i,p}(\xi)\, M_{j,q}(\eta)\, w_{i,j}}{W(\xi,\eta)}, \qquad \forall(\xi,\eta)\in\hat{\Omega}, \quad (2.5.2)$$

where $\hat{\Omega} = [\xi_1, \xi_{n+p+1}) \times [\eta_1, \eta_{m+q+1})$, and the *weighting function* $W(\xi,\eta)$ is defined by

$$W(\xi,\eta) = \sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi)\, M_{\hat{j},q}(\eta)\, w_{\hat{i},\hat{j}}, \qquad \forall(\xi,\eta)\in\hat{\Omega}.$$

2.5.2 NURBS geometries

We can write the expressions for NURBS geometries in the same way as for B-spline geometries using the NURBS basis functions (2.5.1) and (2.5.2). NURBS curves are thus given by

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi)\,\mathbf{B}_i, \qquad \xi\in[\xi_1,\xi_{n+p+1}), \quad (2.5.3)$$

and NURBS surfaces by

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\,\mathbf{B}_{i,j}, \qquad (\xi,\eta)\in\hat{\Omega}, \quad (2.5.4)$$

where $\hat{\Omega} = [\xi_1, \xi_{n+p+1}) \times [\eta_1, \eta_{m+q+1})$. Figure 2.5.1 shows an example of a NURBS surface in $\mathbb{R}^2$ along with the grid lines defined by the knot vectors. The polynomial degrees are $p = 2$ and $q = 1$, the knot vectors $\Xi = [0,0,0,1,1,2,2,2]$ and $\mathcal{H} = [0,0,1,1]$, and the control points and weights are given by

$$[\mathbf{B}_1,\ldots,\mathbf{B}_{10}] = \begin{bmatrix} 1 & 1 & 0 & -1 & -1 & 2 & 2 & 0 & -2 & -2 \\ 0 & 1 & 1 & 1 & 0 & 0 & 2 & 2 & 2 & 0 \end{bmatrix},$$
$$[w_1,\ldots,w_{10}] = \begin{bmatrix} 1 & \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} & 1 & 1 & \frac{1}{\sqrt{2}} & 1 & \frac{1}{\sqrt{2}} & 1 \end{bmatrix},$$

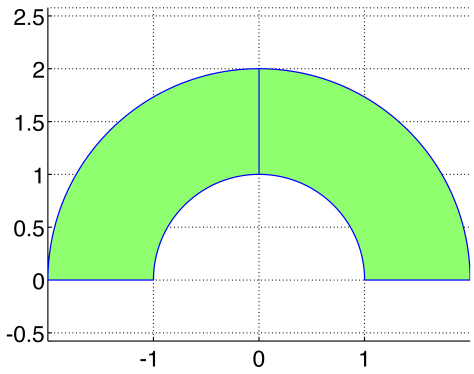

Figure 2.5.1: An example of a NURBS surface along with the grid defined by the knot vectors.

where we have used the numbering scheme $A = n(j-1)+i$. The data of this geometry is obtained without too much trouble using the templates and techniques described in [7, Ch. 2.4]. One of the advantages of using NURBS is that we can represent conic sections exactly. The geometry in Figure 2.5.1 is an example of this with its circular edges.

The NURBS geometries can also be obtained via projective transformations of B-spline geometries [7, Ch. 2.2]. If we have a B-spline geometry in $\mathbb{R}^{d+1}$, this can be projected in a certain way to get a NURBS geometry in $\mathbb{R}^d$, the $\mathbb{R}^{d+1}$-geometry is referred to as the *projective geometry*. If we for instance have a projective B-spline curve $\mathbf{C}^w(\xi)$ in $\mathbb{R}^{d+1}$ defined by the *projective control points* $\mathbf{B}_i^w \in \mathbb{R}^{d+1}$, the control points for the $\mathbb{R}^d$ NURBS curve are given by

$$(\mathbf{B}_i)_j = (\mathbf{B}_i^w)_j / w_i, \quad j = 1, \ldots, d, \tag{2.5.5a}$$

$$w_i = (\mathbf{B}_i^w)_{d+1}, \tag{2.5.5b}$$

where $(\mathbf{B}_i)_j$ denotes the j^{th} component of the control point $\mathbf{B}_i$. In $\mathbb{R}^3$ this corresponds to projecting the control points onto the plane $z = 1$ along rays through the origin. The same transformation can be applied to every point on the projective curve to obtain the NURBS curve, which is then given by

$$(\mathbf{C}(\xi))_j = (\mathbf{C}^w(\xi))_j \Big/ \sum_{i=1}^{n} N_{i,p}(\xi)\, w_i, \quad j = 1, \ldots, d.$$

This formulation is equivalent to the definition (2.5.3).

Even though we use only the definition (2.5.3) to compute points on NURBS geometries, the geometric viewpoint comes in handy when we want to refine the NURBS basis. We can actually use the refinement methods for B-spline geometries discussed in

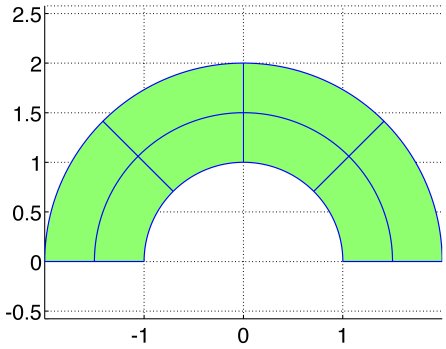

Figure 2.5.2: The NURBS surface from Figure 2.5.1 after inserting simple knots at 0.5 and 1.5 in Ξ, and at 0.5 in $\mathcal{H}$.

Section 2.4 to refine the NURBS basis by applying the methods to the projective geometry. If we have for instance a NURBS curve defined by control points $\mathbf{B}_i$ and weights w_i we can find the projective control points using (2.5.5) by

$$(\mathbf{B}_i^w)_j = w_i(\mathbf{B}_i)_j, \quad j = 1, \ldots, d, \tag{2.5.6a}$$

$$(\mathbf{B}_i^w)_{d+1} = w_i. \tag{2.5.6b}$$

We can then apply the refinement method on the projective curve and get the control points and weights of the refined NURBS curve using (2.5.5). This procedure is the same for NURBS surfaces, we simply transform each control point in the same way. Figure 2.5.2 shows the grid of the NURBS surface from Figure 2.5.1 after knot insertion, the surface is the same as before, but the basis used to express it is refined.

Chapter 3

The Poisson Problem

In this thesis the Poisson problem is primarily used as a model problem to describe the method of isogeometric analysis. However, the Poisson problem is an important problem in itself, and it has important applications in e.g. electrostatics and fluid mechanics where it is used to find potential functions. In fact it follows from the Stokes equations that the pressure of the flow satisfies a special case of the Poisson equation called the Laplace equation, which is useful if boundary conditions are given in terms of pressure [20]. Also solvers for the Poisson problem often generalizes quite easily to solving the biharmonic equation. This equation can be used to find the streamfunction or the vorticity of a flow [20].

3.1 Discretizing the problem

3.1.1 Strong form

We consider the Poisson problem on a domain $\Omega \subset \mathbb{R}^d$, where d is the dimension and the *physical domain* Ω is given as a B-spline or NURBS geometry. This means that Ω is the image of the *parametric domain* $\hat{\Omega}$ under a geometrical mapping $\mathbf{x} : \hat{\Omega} \to \Omega$ defined by (2.2.5), (2.2.6), (2.5.3), (2.5.4) or similar expressions for $d \geq 3$. The parametric domain $\hat{\Omega}$ is the d-dimensional hyperrectangle defined by end values of the knot vectors.

For the Poisson problem we seek to find a function $u : \bar{\Omega} \to \mathbb{R}$ such that

$$\Delta u + f = 0 \qquad \text{in } \Omega, \tag{3.1.1a}$$

$$u = g \qquad \text{on } \Gamma_D, \tag{3.1.1b}$$

$$\nabla u \cdot \mathbf{n} = h \qquad \text{on } \Gamma_N, \tag{3.1.1c}$$

where $\Gamma_D \cup \Gamma_N = \Gamma = \partial\Omega$, $\Gamma_D \cap \Gamma_N = \emptyset$, $\Gamma_D \neq \emptyset$ and $\mathbf{n}$ is the outward unit normal vector of $\partial\Omega$. The Dirichlet boundary condition (3.1.1b) is specified by a given function

$g : \Gamma_D \to \mathbb{R}$, and the Neumann boundary condition (3.1.1c) by the given function $h : \Gamma_N \to \mathbb{R}$. For simplicity we will assume $g = 0$ in this report, but the general case can be handled without too much trouble using a so called lifting function. Another type of boundary condition that is worth mentioning is the Robin condition, which gives the value of $\beta u + \nabla u \cdot \mathbf{n}$, for some constant β, on a part Γ_R of the boundary. This type of boundary condition is handled in a quite similar fashion as the Neumann condition.

The standard way of solving the Poisson problem numerically using finite element analysis is by use of the Galerkin method. We first find the weak formulation of (3.1.1), then approximate the infinite-dimensional function spaces by finite-dimensional ones and finally a system of linear equations is obtained and solved.

3.1.2 Weak form

To find the weak formulation of (3.1.1), we multiply both sides of (3.1.1a) with an arbitrary test function v, and integrate over Ω. This gives

$$\int_\Omega v \Delta u d\Omega = -\int_\Omega v f d\Omega,$$

and integrating by parts using Green's identity yields

$$\int_\Omega \nabla v \cdot \nabla u d\Omega = \int_\Omega v f d\Omega + \int_{\Gamma_D} v \left(\nabla u \cdot \mathbf{n}\right) d\Gamma + \int_{\Gamma_N} v \left(\nabla u \cdot \mathbf{n}\right) d\Gamma.$$

If we assume that $v|_{\Gamma_D} = 0$ and use the Neumann boundary condition (3.1.1c) we end up with

$$\int_\Omega \nabla v \cdot \nabla u d\Omega = \int_\Omega v f d\Omega + \int_{\Gamma_N} v h d\Gamma.$$

In order for these integrals to make sense, v and its first order derivatives must be square-integrable. We have also used that $v|_{\Gamma_D} = 0$. We will therefore require that

$$v \in \mathcal{V} = \mathrm{H}^1_{\Gamma_D}(\Omega) = \left\{ v \in \mathrm{H}^1(\Omega) : v|_{\Gamma_D} = 0 \right\},$$

where $\mathrm{H}^1(\Omega)$ is the Sobolev space $\mathrm{H}^1(\Omega) = \{v \in \mathrm{L}^2(\Omega) : D^{\boldsymbol{\alpha}} v \in \mathrm{L}^2(\Omega), \forall\, |\boldsymbol{\alpha}| \leq 1\}$. We will also be searching for the solution u in the same space $\mathcal{V}$. Hence the weak formulation of (3.1.1) can be stated as follows: Find $u \in \mathcal{V}$ such that

$$a(v, u) = L(v), \quad \forall v \in \mathcal{V}, \tag{3.1.2}$$

where $a(v, u) = \int_\Omega \nabla v \cdot \nabla u d\Omega$ and $L(v) = \int_\Omega v f d\Omega + \int_{\Gamma_N} v h d\Gamma$. From the linearity of the integral and the differential operators and the commutativity of the dot product it

follows that $a(\cdot,\cdot)$ is bilinear and symmetric and that $L(\cdot)$ is linear. It can be shown that $a(\cdot,\cdot)$ is also continuous and coercive, and $L(\cdot)$ is also continuous [26, Ch. 3.4]. With $\mathcal{V}$ being a closed subspace of the Hilbert space $\mathrm{H}^1(\Omega)$ and thus a Hilbert space itself, the Lax-Milgram Lemma [26, Ch. 3.5] applies and guarantees the existence of a unique solution to (3.1.2).

3.1.3 Galerkin's method

The idea of Galerkin's method is to use a finite-dimensional subspace $\mathcal{V}^h$ to approximate the search and test space $\mathcal{V}$. Isogeometric analysis follows the isoparametric concept where the same basis is used for both geometry and analysis. Thus the solution space $\mathcal{V}^h$ is chosen to be the space spanned by the NURBS or B-spline basis functions N_A for $A = 1, \ldots, n_{np}$, where n_{np} is the number of basis functions. Note that we use the notation N_A for the basis functions whether they are the B-spline basis funcitons $N_{i,p}(\xi)$ or $N_{i,j;p,q}(\xi,\eta)$, or the NURBS basis functions $R_i^p(\xi)$ or $R_{i,j}^{p,q}(\xi,\eta)$. Strictly speaking the basis functions N_A should be functions of the physical parameter $\mathbf{x}$ defined as compositions $N_A \circ \mathbf{x}^{-1}$, where $\mathbf{x}^{-1}$ is the inverse of the geometrical mapping, but we commit the notational crime of denoting both functions by N_A. Now the homogeneous Dirichlet boundary condition (3.1.1b) where $g = 0$ can be incorporated into the solution space simply by reducing the basis to the functions for which $N_A|_{\Gamma_D} = 0$. If we reorder these basis functions using the indices $\tilde{A}$ such that $N_{\tilde{A}}|_{\Gamma_D} = 0$ for $\tilde{A} = 1, \ldots, n_{eq}$, we have that any function $v^h \in \mathcal{V}^h$ can be represented by

$$v^h = \sum_{\tilde{A}=1}^{n_{eq}} N_{\tilde{A}} c_{\tilde{A}}$$

for some constants $c_{\tilde{A}}$.

We can now make an approximation to the weak formulation (3.1.2), called the Galerkin problem: Find $u^h \in \mathcal{V}^h$ such that

$$a\left(v^h, u^h\right) = L\left(v^h\right), \quad \forall v^h \in \mathcal{V}^h. \tag{3.1.3}$$

Now $\mathcal{V}^h$ is in turn a closed subspace of the Hilbert space $\mathcal{V}$, and the Lax-Milgram Lemma guarantees the existence of a unique solution to the problem (3.1.3) as well. Writing u^h as $u^h = \sum_{\tilde{B}=1}^{n_{eq}} N_{\tilde{B}} d_{\tilde{B}}$, where $d_{\tilde{B}}$ are the unknown control variables, and using the linearity of a we require that

$$a\left(v^h, \sum_{\tilde{B}=1}^{n_{eq}} N_{\tilde{B}} d_{\tilde{B}}\right) = \sum_{\tilde{B}=1}^{n_{eq}} a\left(v^h, N_{\tilde{B}}\right) d_{\tilde{B}} = L\left(v^h\right), \quad \forall v^h \in \mathcal{V}^h.$$

From the linearity of $a(\cdot,\cdot)$ and $L(\cdot)$ it follows that the above equation will be satisfied by any linear combination of elements of $\mathcal{V}^h$ that satisfy the equation. Since the basis

functions $N_{\tilde{A}}$ span $\mathcal{V}^h$ and are linearly independent, we have that for the equality to hold for all $v \in \mathcal{V}^h$ it is sufficient and necessary that it holds for all the basis functions. Hence the Galerkin problem (3.1.3) is equivalent to

$$\sum_{\tilde{B}=1}^{n_{eq}} a\left(N_{\tilde{A}}, N_{\tilde{B}}\right) d_{\tilde{B}} = L\left(N_{\tilde{A}}\right), \qquad \tilde{A} = 1, \ldots, n_{eq}.$$

This is a system of linear equations, and if we define $\mathbf{K} = \left[K_{\tilde{A},\tilde{B}}\right] = \left[a\left(N_{\tilde{A}}, N_{\tilde{B}}\right)\right]$, $\mathbf{d} = [d_{\tilde{B}}]$ and $\mathbf{F} = [F_{\tilde{A}}] = \left[L\left(N_{\tilde{A}}\right)\right]$ we can write the system in matrix form by $\mathbf{Kd} = \mathbf{F}$. Thus the Galerkin problem (3.1.3) is equivalent to finding $\mathbf{d} \in \mathbb{R}^{n_{eq}}$ such that

$$\mathbf{Kd} = \mathbf{F}. \tag{3.1.4}$$

The matrix $\mathbf{K}$ is often referred to as the stiffness matrix, the vector $\mathbf{F}$ as the force vector and the vector $\mathbf{d}$ as the displacement vector. With $a(\cdot, \cdot)$ being bilinear, coercive and symmetric, it is not difficult to show that the matrix $\mathbf{K}$ is symmetric, positive definite [26, Ch. 4.1]. Solving the system (3.1.4) for $\mathbf{d}$ we find the unknown control variables, and the numerical solution is finally given by

$$u^h = \sum_{\tilde{A}=1}^{n_{eq}} N_{\tilde{A}} d_{\tilde{A}}. \tag{3.1.5}$$

It should be noted that since the NURBS-basis is not interpolatory at the knots, the control variables $d_{\tilde{A}}$ have no direct interpretation, as opposed to ordinary FEA where the $d_{\tilde{A}}$ would be the values of the numerical solution at the nodes.

3.2 Implementation of numerical solver

We now want to implement a numerical solver for the Poisson problem (3.1.1) in the two-dimensional case by assembling and solving the linear system (3.1.4). Implementing a solver for the case $d = 1$ is easy in comparison, because both the indexing and computing the integrals simplify a lot, and generalizing a 2D solver to higher dimensions is not very hard but gives more indices to handle. Thus considering the two-dimensional case is reasonable.

Assume the domain Ω is given as a NURBS surface defined by the knot vectors $\Xi = [\xi_1, \ldots, \xi_{n+p+1}]$ and $\mathcal{H} = [\eta_1, \ldots, \eta_{m+q+1}]$, with corresponding polynomial orders p and q, a set of control points $\{\mathbf{B}_{i,j}\}_{i,j=1}^{n,m}$ and a set of weights $\{w_{i,j}\}_{i,j=1}^{n,m}$. In principle we could let Ω be a three-dimensional NURBS surface by letting the control points be elements of $\mathbb{R}^3$ and compute the relevant integrals as surface integrals, however for simplicity and visualization purposes we will assume $\mathbf{B}_{i,j} \in \mathbb{R}^2$. We have then that

$$\Omega = \left\{\mathbf{x} \in \mathbb{R}^2 : \mathbf{x} = \mathbf{S}\left(\xi, \eta\right), \xi \in \left(\xi_1, \xi_{n+p+1}\right), \eta \in \left(\eta_1, \eta_{m+q+1}\right)\right\},$$

where $\mathbf{S}(\xi, \eta)$ is given by (2.5.4). Thus Ω is given as the image of the rectangle $(\xi_1, \xi_{n+p+1}) \times (\eta_1, \eta_{m+q+1})$ under the mapping $(\xi, \eta) \mapsto \mathbf{x} = \mathbf{S}(\xi, \eta)$. The rectangle $(\xi_1, \xi_{n+p+1}) \times (\eta_1, \eta_{m+q+1})$ is referred to as the *parametric domain*, while Ω is referred to as the *physical domain*. The elements, or the *physical elements*, are defined as the images of *parametric elements* $[\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1})$ under the geometrical mapping.

3.2.1 Index handling by connectivity arrays

The basis functions we consider are the $R_{i,j}^{p,q}$ given by (2.5.2), where $i = 1, \ldots, n$ and $j = 1, \ldots, m$. These basis functions are tensor products of the univariate basis functions $N_{i,p}(\xi)$ and $M_{j,q}(\eta)$ multiplied by the weight and divided by a weighting function. The pair of indices (i, j) is called the NURBS coordinates, and they refer to the univariate basis functions that correspond to the bivariate basis function. If we introduce the global numbering

$$A = n(j-1) + i, \tag{3.2.1}$$

we can denote the basis functions by $N_A = R_{i,j}^{p,q}$. To keep track of the NURBS coordinates of a given basis function N_A, we use the NURBS coordinates array INC. It is constructed such that $\text{INC}(A, 1) = i$ and $\text{INC}(A, 2) = j$. A useful observation is that the NURBS coordinates of a basis function give the indices of the knots where the support of the function begins in each parametric direction, this is due to Property 2 and the tensor product nature of the basis.

Since we are assuming open knot vectors, we know that there will be elements of zero measure at the ends of the knot vectors, so these are disregarded. Thus we are only numbering the elements Ω^e that correspond to a parametric element $[\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1})$ for which $p+1 \leq i \leq n$ and $q+1 \leq j \leq m$. These are numbered by

$$e = (j-q-1)(n-p) + (i-p).$$

Given a parametric element $[\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1})$ we know from Proposition 2 that a tensor product $N_{\alpha,p}(\xi) M_{\beta,q}(\eta)$ has support on the element if and only if $i - p \leq \alpha \leq i$ and $j - q \leq \beta \leq j$. Thus for each element the only basis functions $R_{\alpha,\beta}^{p,q}(\xi, \eta) = \frac{N_{\alpha,p}(\xi) M_{\beta,q}(\eta) w_{i,j}}{W(\xi,\eta)}$ that have support on the element are the $n_{en} = (p+1)(q+1)$ basis functions for which $i - p \leq \alpha \leq i$ and $j - q \leq \beta \leq j$. These are referred to as local basis functions and they are given local numbers b according to the numbering scheme

$$b = (p+1)(j-\beta) + (i-\alpha) + 1 = (p+1) j_{loc} + i_{loc} + 1. \tag{3.2.2}$$

Here we have defined the local indices $i_{loc} = i - \alpha$ and $j_{loc} = j - \beta$ such that $0 \leq i_{loc} \leq p$ and $0 \leq j_{loc} \leq q$. The local basis function numbers are thus incremented starting at $\alpha = i, \beta = j$ and counting backwards first in α, then in β. With the global numbering

given by (3.2.1), we note that the global basis function number of the function with NURBS coordinates (α, β) is given by

$$B = n\,(\beta - 1) + \alpha = n\,(j-1) + i - n\,(j - \beta) - (i - \alpha) = A - j_{loc}n - i_{loc},$$

where A is the global basis function number of $N_A = R^{p,q}_{i,j}$. The element nodes array IEN is used to keep track of the global basis function numbers of the local basis functions on a given element. Given an element e and a local number b, the global number is given by $B = \text{IEN}(b, e)$.

The above formulas and an algorithm for setting up the connectivity arrays INC and IEN are presented in [7, App. A].

3.2.2 Assembling the linear system

The stiffness matrix

If we denote the elements by Ω^e, $e = 1, \ldots, n_{el}$, where n_{el} is the number of elements, the entries of the stiffness matrix $\mathbf{K} \in \mathbb{R}^{n_{np} \times n_{np}}$ can be written

$$K_{A,B} = a\,(N_A, N_B) = \int_\Omega \nabla N_A \cdot \nabla N_B d\Omega = \sum_{e=1}^{n_{el}} \int_{\Omega^e} \nabla N_A \cdot \nabla N_B d\Omega,$$

splitting the integral over Ω into a sum of integrals over the elements Ω^e. This means that we can loop through the elements adding contributions to the integral as we go. Because of the local support of the basis functions, the element integrals $\int_{\Omega^e} \nabla N_A \cdot \nabla N_B d\Omega$ will be zero for a lot of functions N_A and N_B. From Section 3.2.1 we have that the basis functions with support on element e are the N_A with $A = \text{IEN}(a, e)$ for $a = 1, \ldots, n_{en}$, where $n_{en} = (p+1)(q+1)$ is the number of local basis functions. Thus we have that

$$K^e_{a,b} := \int_{\Omega^e} \nabla N_{\text{IEN}(a,e)} \cdot \nabla N_{\text{IEN}(b,e)} d\Omega, \quad a, b = 1, \ldots, n_{en} \tag{3.2.3}$$

are the only contributions we have from element e to the stiffness matrix $\mathbf{K}$. Hence the integrals can be computed and stored in the *element stiffness matrix* $\mathbf{K}^e = \left[K^e_{a,b}\right]_{a,b=1}^{n_{en}}$, and then the contributions of this matrix is added to the global stiffness matrix. We also note that if the element Ω^e has zero measure, any integral over the element is zero and the matrix $\mathbf{K}^e = \mathbf{0}$, this will happen whenever $\xi_i = \xi_{i+1}$ or $\eta_j = \eta_{j+1}$ for the corresponding parametric element $[\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1})$.

The force vector

In the same fashion as for the stiffness matrix we can write the elements of the force vector $\mathbf{F}$ as

$$F_A = L(N_A) = \int_\Omega N_A f d\Omega + \int_{\Gamma_N} N_A h d\Gamma = \sum_{e=1}^{n_{el}} \left[\int_{\Omega^e} N_A f d\Omega + I^e_{\Gamma_N}(N_A) \right].$$

Here $I^e_{\Gamma_N}(N_A)$ denotes the contribution from the element Ω^e to the integral $\int_{\Gamma_N} N_A h d\Gamma$. Using the same argument as for the stiffness matrix regarding the support of the basis functions, we can define an *element force vector* $\mathbf{F}^e \in \mathbb{R}^{n_{en}}$ with entries given by

$$F^e_a := \int_{\Omega^e} N_{\mathrm{IEN}(a,e)} f d\Omega + I^e_{\Gamma_N}\left(N_{\mathrm{IEN}(a,e)}\right). \tag{3.2.4}$$

The contribution $I^e_{\Gamma_N}\left(N_{\mathrm{IEN}(a,e)}\right)$ to the integral $\int_{\Gamma_N} N_{\mathrm{IEN}(a,e)} h d\Gamma$ is zero for most of the elements Ω^e. With open knot vectors we need only compute these contributions for the elements whose corresponding parametric element $[\xi_i, \xi_{i+1}) \times [\eta_j, \eta_{j+1})$ is such that $i = p+1$, $i = n$, $j = q+1$ or $j = m$ so that a part of the element boundary is a part of the boundary $\partial\Omega$ of the physical domain, we must also check if this part is a part of the Neumann boundary Γ_N.

Gaussian quadrature

The n_q-point Gaussian quadrature rule can be stated as

$$\int_{-1}^{1} F\left(\tilde{\xi}\right) d\tilde{\xi} \approx \sum_{i=1}^{n_q} W_i F\left(\tilde{\xi}_i\right) \tag{3.2.5}$$

with the remainder

$$R_{n_q} = \int_{-1}^{1} F\left(\tilde{\xi}\right) d\tilde{\xi} - \sum_{i=1}^{n_q} w_i F\left(\tilde{\xi}_i\right) = \frac{2^{2n_q+1}(n_q!)^4}{(2n_q+1)\left[(2n_q)!\right]^3} F^{(2n_q)}(\tau), \quad \tau \in (-1,1). \tag{3.2.6}$$

These formulas are produced using the Legendre polynomials $L_{n_q}(\tilde{\xi})$ which are orthogonal with respect to the inner product $(F,G) = \int_{-1}^{1} F\left(\tilde{\xi}\right) G\left(\tilde{\xi}\right) d\tilde{\xi}$, and normalized by $L_{n_q}(1) = 1$. The quadrature point $\tilde{\xi}_i$ is the i-th zero of the polynomial L_{n_q}, and the weights are given by $W_i = \frac{2}{\left(1-\tilde{\xi}_i^2\right)\left[L_{n_q}{}'\left(\tilde{\xi}_i\right)\right]^2}$ [1, p. 887]. Using (3.2.6) we note that the Gaussian quadrature gives the exact value of the integral for polynomials of degree not exceeding $2n_q - 1$, since the $(2n_q)^{th}$ derivative will then be identically zero. For two-dimensional integrals we can use (3.2.5) in each direction and obtain

$$\int_{-1}^{1}\int_{-1}^{1} F\left(\tilde{\xi},\tilde{\eta}\right) d\tilde{\xi} d\tilde{\eta} \approx \sum_{i=1}^{n_q}\sum_{j=1}^{m_q} W_i W_j F\left(\tilde{\xi}_i, \tilde{\eta}_j\right). \tag{3.2.7}$$

This formula will be exact if $F(\tilde{\xi}, \tilde{\eta})$ is a tensor product of polynomials of degree not exceeding $2n_q - 1$ and $2m_q - 1$ respectively.

We now want to apply Gaussian quadrature to compute integrals over arbitrary physical elements. In order to do this the physical element Ω^e is pulled back first to the parametric element $\hat{\Omega}^e = (\xi_i, \xi_{i+1}) \times (\eta_j, \eta_{j+1})$ and then to the bi-unit *parent element* $\tilde{\Omega}^e = (-1, 1) \times (-1, 1)$. More specifically if we have an integral over a physical element, we can perform a change of variables using the geometrical mapping $\mathbf{x}(\boldsymbol{\xi})$, $\boldsymbol{\xi} = [\xi, \eta]^T$, and obtain

$$\int_{\Omega^e} F(\mathbf{x})\, d\mathbf{x} = \int_{\hat{\Omega}^e} F(\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right) \right| d\boldsymbol{\xi},$$

where $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$ is the Jacobian matrix of the geometrical mapping. Now we can transform this integral to an integral over the parent element using the affine mapping given by

$$\boldsymbol{\xi} = \frac{1}{2} \begin{bmatrix} \xi_{i+1} - \xi_i & 0 \\ 0 & \eta_{i+1} - \eta_i \end{bmatrix} \tilde{\boldsymbol{\xi}} + \frac{1}{2} \begin{bmatrix} \xi_{i+1} + \xi_i \\ \eta_{i+1} + \eta_i \end{bmatrix}, \tag{3.2.8}$$

which gives

$$\int_{\Omega^e} F(\mathbf{x})\, d\mathbf{x} = \int_{\tilde{\Omega}^e} F(\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \tilde{\boldsymbol{\xi}}} \right) \right| d\tilde{\boldsymbol{\xi}},$$

where the integrand varies with $\tilde{\boldsymbol{\xi}} = \left[\tilde{\xi}, \tilde{\eta}\right]^T$ through the geometrical mapping and the affine mapping (3.2.8). Such an integral can be approximated using the Gaussian quadrature formula given by (3.2.7).

Shape function routine

The integrals $\int_{\Omega^e} N_{\mathrm{IEN}(a,e)}(\mathbf{x})\, f(\mathbf{x})\, d\mathbf{x}$ and $\int_{\Omega^e} \nabla N_{\mathrm{IEN}(a,e)}(\mathbf{x}) \cdot \nabla N_{\mathrm{IEN}(b,e)}(\mathbf{x})\, d\mathbf{x}$ will be computed using Gaussian quadrature, so we need to compute the values of the relevant basis functions, their derivatives and also the determinant of the Jacobian $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \tilde{\boldsymbol{\xi}}}$, at given quadrature points $(\tilde{\xi}, \tilde{\eta})$. We also need to compute $\mathbf{x}$ for evaluation of $f(\mathbf{x})$. This is done by a *shape function routine*. The algorithm for such a routine is given in [7, App. 3.A] for the three-dimensional case, and can with little effort be adapted to the two-dimensional case. In this section we will, however, develop the formulas from scratch in order to obtain a matrix-based algorithm which avoids for-loops for use in MATLAB, and also to give a proof of the algorithm.

If we use the numbering given in (3.2.1) for the control points and apply what we know about the support of the basis functions, the geometrical mapping can be expressed locally for element number e as

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{i=1}^{n} \sum_{j=1}^{m} R_{i,j}^{p,q}(\boldsymbol{\xi})\, \mathbf{B}_{i,j} = \sum_{A=1}^{n_{np}} N_A(\boldsymbol{\xi})\, \mathbf{B}_A = \sum_{a=1}^{n_{en}} N_{\mathrm{IEN}(a,e)}(\boldsymbol{\xi})\, \mathbf{B}_{\mathrm{IEN}(a,e)}, \quad \boldsymbol{\xi} \in \hat{\Omega}^e.$$

Defining the matrix $\mathcal{B}^e = \left[\mathbf{B}_{\mathrm{IEN}(1,e)}, \ldots, \mathbf{B}_{\mathrm{IEN}(n_{en},e)}\right]^T \in \mathbb{R}^{n_{en}\times 2}$ and the vector $\mathbf{N}^e = \left[N_{\mathrm{IEN}(1,e)}, \ldots, N_{\mathrm{IEN}(n_{en},e)}\right]^T \in \mathbb{R}^{n_{en}}$, which contain the relevant control points and basis functions respectively, this can be written

$$\mathbf{x}\left(\boldsymbol{\xi}\right) = \left(\mathcal{B}^e\right)^T \mathbf{N}^e\left(\boldsymbol{\xi}\right), \tag{3.2.9}$$

Differentiating with respect to $\boldsymbol{\xi}$ gives us the Jacobian matrix of the geometrical mapping:

$$\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} = \left(\mathcal{B}^e\right)^T \frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}}. \tag{3.2.10}$$

The chain rule gives that

$$\frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}} = \frac{\partial \mathbf{N}^e}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}},$$

and if the geometrical mapping is invertible at $\boldsymbol{\xi}$, the derivatives with respect to the physical variables are given by

$$\frac{\partial \mathbf{N}^e}{\partial \mathbf{x}} = \frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}} \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right)^{-1}. \tag{3.2.11}$$

Now we only need formulas for $\mathbf{N}^e$ and $\frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}}$. To obtain these we take a similar approach as for the geometrical mapping above. If the NURBS coordinates of the element Ω^e are (i,j), the functions $N_{A,p}\left(\xi\right) M_{B,q}\left(\eta\right)$ with support on the element Ω^e are the $N_{\alpha,p}\left(\xi\right) M_{\beta,q}\left(\eta\right)$ for which $i-p \leq \alpha \leq i$ and $j-q \leq \beta \leq j$. Using the local numbering $b = (p+1)(j-\beta) + (i-\alpha) + 1$ from (3.2.2) the elements of $\mathbf{N}^e$ can be written

$$N_{\mathrm{IEN}(b,e)}\left(\boldsymbol{\xi}\right) = \frac{N_{\alpha,p}\left(\xi\right) M_{\beta,q}\left(\eta\right) w_{\alpha,\beta}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m} N_{\hat{i},p}\left(\xi\right) M_{\hat{j},q}\left(\eta\right) w_{\hat{i},\hat{j}}} = \frac{N_{\alpha,p}\left(\xi\right) M_{\beta,q}\left(\eta\right) w_{\mathrm{IEN}(b,e)}}{\sum_{\hat{i}=i-p}^{i}\sum_{\hat{j}=j-q}^{j} N_{\hat{i},p}\left(\xi\right) M_{\hat{j},q}\left(\eta\right) w_{\mathrm{IEN}(b,e)}}. \tag{3.2.12}$$

If we now define the vectors $\mathcal{N}(\boldsymbol{\xi}), \mathbf{W} \in \mathbb{R}^{n_{en}}$ by

$$\mathcal{N}(\boldsymbol{\xi})^T = \tag{3.2.13}$$
$$[N_{i,p}\left(\xi\right) M_{j,q}\left(\eta\right) \cdots N_{i-p,p}\left(\xi\right) M_{j,q}\left(\eta\right) \cdots N_{i,p}\left(\xi\right) M_{j-q,q}\left(\eta\right) \cdots N_{i-p,p}\left(\xi\right) M_{j-q,q}\left(\eta\right)],$$

$$\mathbf{W} = [w_{i,j}, \ldots, w_{i-p,j}, \ldots, w_{i,j-q}, \ldots, w_{i-p,j-q}] = \left[w_{\mathrm{IEN}(1,e)}, \ldots, w_{\mathrm{IEN}(n_{en},e)}\right]^T,$$

we use (3.2.12) to observe that we can write

$$\mathbf{N}^e = \frac{\mathrm{diag}\left(\mathbf{W}\right)\mathcal{N}}{\mathbf{W}^T\mathcal{N}}, \tag{3.2.14}$$

where $\operatorname{diag}(\mathbf{W}) \in \mathbb{R}^{n_{en} \times n_{en}}$ is a matrix with the vector $\mathbf{W}$ along the main diagonal. Differentiating this with respect to $\boldsymbol{\xi}$ using the quotient rule yields

$$\begin{aligned}\frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}} &= \frac{\operatorname{diag}(\mathbf{W}) \frac{\partial \mathcal{N}}{\partial \boldsymbol{\xi}} \left(\mathbf{W}^T \mathcal{N}\right) - \operatorname{diag}(\mathbf{W}) \mathcal{N} \mathbf{W}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{\xi}}}{\left(\mathbf{W}^T \mathcal{N}\right)^2} \\ &= \operatorname{diag}(\mathbf{W}) \frac{\left(\mathbf{W}^T \mathcal{N}\right) I - \mathcal{N} \mathbf{W}^T}{\left(\mathbf{W}^T \mathcal{N}\right)^2} \frac{\partial \mathcal{N}}{\partial \boldsymbol{\xi}}. \end{aligned} \tag{3.2.15}$$

We can see from the definition (3.2.13) that $\frac{\partial \mathcal{N}}{\partial \boldsymbol{\xi}} \in \mathbb{R}^{n_{en} \times 2}$ is given by

$$\frac{\partial \mathcal{N}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} {N_{i,p}}'(\xi) M_{j,q}(\eta) & N_{i,p}(\xi) {M_{j,q}}'(\eta) \\ \vdots & \vdots \\ {N_{i-p,p}}'(\xi) M_{j,q}(\eta) & N_{i-p,p}(\xi) {M_{j,q}}'(\eta) \\ \vdots & \vdots \\ {N_{i,p}}'(\xi) M_{j-q,q}(\eta) & N_{i,p}(\xi) {M_{j-q,q}}'(\eta) \\ \vdots & \vdots \\ {N_{i-p,p}}'(\xi) M_{j-q,q}(\eta) & N_{i-p,p}(\xi) {M_{j-q,q}}'(\eta) \end{bmatrix}. \tag{3.2.16}$$

Now $\mathcal{N}$ and $\frac{\partial \mathcal{N}}{\partial \boldsymbol{\xi}}$ can be computed by evaluation of univariate B-splines and their derivatives according to the formulas in Section 2.1.

The computations of the shape function routine can now be summarized in the following steps:

- Given a quadrature point $\tilde{\boldsymbol{\xi}} = (\tilde{\xi}, \tilde{\eta}) \in \tilde{\Omega}^e$ compute the corresponding point $\boldsymbol{\xi} = (\xi, \eta) \in \Omega^e$ by (3.2.8).
- Evaluate the non-zero univariate B-splines and their derivatives at ξ and η according to the formulas in Section 2.1, and multiply them together forming the matrices $\mathcal{N}$ and $\frac{\partial \mathcal{N}}{\partial \boldsymbol{\xi}}$ defined by (3.2.13) and (3.2.16).
- Compute the matrices $\mathbf{N}^e$ and $\frac{\partial \mathbf{N}^e}{\partial \boldsymbol{\xi}}$ by (3.2.14) and (3.2.15) respectively.
- Compute the point $\mathbf{x} \in \Omega^e$ corresponding to the quadrature point by (3.2.9), and the Jacobian $\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}$ of the geometrical mapping by (3.2.10).
- Compute the derivatives with respect to the physical variables using (3.2.11).
- Compute the Jacobian determinant by $\det\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \tilde{\boldsymbol{\xi}}}\right) = \frac{(\xi_{i+1} - \xi_i)(\eta_{j+1} - \eta_j)}{4} \det\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right)$.

After doing these computations the shape function routine returns the point $\mathbf{x}$ in physical space corresponding to the quadrature point, the vector $\mathbf{N}^e$ containing the values of the basis functions at $\mathbf{x}$, the matrix $\frac{\partial \mathbf{N}^e}{\partial \mathbf{x}}$ whose rows are the gradients of the basis functions at $\mathbf{x}$ and the Jacobian determinant $\det\left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \frac{\partial \boldsymbol{\xi}}{\partial \tilde{\boldsymbol{\xi}}}\right)$.

Neumann boundary conditions

In Section 3.2.2 we saw that if the NURBS coordinates (i, j) of an element Ω^e is such that $i = p + 1$, $i = n$, $j = q + 1$ or $j = m$, then we have to compute a contribution $I^e_{\Gamma_N}\left(N_{\mathrm{IEN}(a,e)}\right)$ from element e to the integral $\int_{\Gamma_N} N_{\mathrm{IEN}(a,e)} h d\Gamma$ if the Neumann boundary Γ_N intersects the element boundary. If for example the parametric element is given by $[\xi_i, \xi_{i+1}) \times [\eta_m, \eta_{m+1})$, then the part of the element boundary $[\xi_i, \xi_{i+1}) \times \eta_{m+1}$ corresponds to a part of the domain boundary $\partial\Omega$, and if this part is contained in the Neumann boundary Γ_N[1], the contribution $I^e_{\Gamma_N}\left(N_{\mathrm{IEN}(a,e)}\right)$ can be computed as a line integral by

$$
\begin{aligned}
I^e_{\Gamma_N}\left(N_{\mathrm{IEN}(a,e)}\right) &= \int_{\xi_i}^{\xi_{i+1}} h\left(\mathbf{x}\left(\xi, \eta_{m+1}\right)\right) N_{\mathrm{IEN}(a,e)}\left(\mathbf{x}\left(\xi, \eta_{m+1}\right)\right) \left\| \frac{d\mathbf{x}}{d\xi}\left(\xi, \eta_{m+1}\right) \right\|_2 d\xi \\
&= \int_{-1}^{1} h\left(\mathbf{x}\left(\boldsymbol{\xi}\left(\tilde{\xi}, 1\right)\right)\right) N_{\mathrm{IEN}(a,e)}\left(\mathbf{x}\left(\boldsymbol{\xi}\left(\tilde{\xi}, 1\right)\right)\right) \frac{\xi_{i+1} - \xi_i}{2} \left\| \frac{d\mathbf{x}}{d\xi}\left(\boldsymbol{\xi}\left(\tilde{\xi}, 1\right)\right) \right\|_2 d\tilde{\xi}.
\end{aligned}
$$

Similar integrals are obtained in the three other cases $i = p + 1$, $i = n$, $j = q + 1$, and for "corner elements" where for example $i = n$ *and* $j = q + 1$, the contribution may consist of two such integrals. These integrals can be computed using one-dimensional Gaussian quadrature as given by (3.2.5). The shape function routine can without much difficulty be adapted to compute the relevant values for these integrals, instead of the Jacobian determinant the 2-norm of the parametrization is computed since they are line integrals.

One way to identify the Neumann part of the boundary is the following: For the lower edge of the rectangular reference domain $[\xi_1, \xi_{n+p+1}) \times [\eta_1, \eta_{m+q+1})$, i.e. the edge where $\eta = \eta_1$, we assign a vector $\Xi^{\mathrm{Neu,lower}}$ of the same size as the knot vector Ξ containing boolean entries, such that if $\xi_i^{\mathrm{Neu,lower}}$ and $\xi_{i+1}^{\mathrm{Neu,lower}}$ both have the value $true$ it means that the segment $[\xi_i, \xi_{i+1}) \times \eta_1$ corresponds to a part of the Neumann boundary Γ_N. We then assign three more such vectors for the three remaining edges.

Assembling the stiffness matrix and the force vector

Since the rows of the matrix $\frac{\partial \mathbf{N}^e}{\partial \mathbf{x}}$ contains the gradients of the $N_{\mathrm{IEN}(a,e)}$ in the order $a = 1, \ldots, n_{en}$ we see from (3.2.3) that we can write the element stiffness matrix as

$$
\mathbf{K}^e = \int_{\Omega^e} \frac{\partial \mathbf{N}^e}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{N}^e}{\partial \mathbf{x}} \right)^T d\mathbf{x}.
$$

[1] We assume for simplicity that the part is a subset of the Neumann boundary whenever the intersection is non-empty.

Similarly we observe from (3.2.4) that the element force vector can be written

$$\mathbf{F}^e = \int_{\Omega^e} f(\mathbf{x})\,\mathbf{N}^e(\mathbf{x})\,d\mathbf{x} + I^e_{\Gamma_N}(\mathbf{N}^e).$$

Thus, using the Gaussian quadrature formulas, we finally obtain

$$\mathbf{K}^e \approx \sum_{i=1}^{n_q}\sum_{j=1}^{m_q} W_i W_j \frac{\partial \mathbf{N}^e}{\partial \mathbf{x}}\left(\tilde{\xi}_i, \tilde{\eta}_j\right)\left(\frac{\partial \mathbf{N}^e}{\partial \mathbf{x}}\left(\tilde{\xi}_i, \tilde{\eta}_j\right)\right)^T,$$

$$\mathbf{F}^e \approx \sum_{i=1}^{n_q}\sum_{j=1}^{m_q} W_i W_j f\left(\mathbf{x}\left(\tilde{\xi}_i, \tilde{\eta}_j\right)\right)\mathbf{N}^e\left(\mathbf{x}\left(\tilde{\xi}_i, \tilde{\eta}_j\right)\right) + I^e_{\Gamma_N}(\mathbf{N}^e),$$

where $I^e_{\Gamma_N}(\mathbf{N}^e)$ is the Neumann contribution which is computed by one-dimensional Gaussian quadrature. With the element stiffness matrix and force vector computed, their contributions to the global stiffness matrix and force vector respectively can be added using the global indices $A = \mathrm{IEN}(a, e)$ and $B = \mathrm{IEN}(b, e)$ for $a, b = 1, \ldots, n_{en}$. It is important to note that the global stiffness matrix $\mathbf{K}$ should be stored as a sparse matrix, since most of its elements are zero due to the local support of the basis functions.

3.2.3 Imposing boundary conditions and solving the linear system

The stiffness matrix $\mathbf{K}$ computed by the assembly process of Section 3.2.2 is an $n_{en} \times n_{en}$-matrix containing $a(N_A, N_B)$ for all basis functions N_A and N_B, $A, B = 1, \ldots, n_{en}$. However the stiffness matrix $\mathbf{K}$ in the linear system (3.1.4), which is the system we need to solve, is an $n_{eq} \times n_{eq}$-matrix containing $a(N_{\tilde{A}}, N_{\tilde{B}})$ for the basis functions $N_{\tilde{A}}$ and $N_{\tilde{B}}$, $\tilde{A}, \tilde{B} = 1, \ldots, n_{eq}$. If we look back to Section 3.1.3 we see that the $N_{\tilde{A}}$ are the basis functions that vanish when restricted to the Dirichlet boundary Γ_D. This means that in order to obtain the stiffness matrix for the linear system, we need to identify the basis function numbers A of the basis functions that are non-zero on the Dirichlet boundary and remove the corresponding rows and columns of the stiffness matrix obtained in the assembly. We also need to remove the corresponding entries of the assembled force vector $\mathbf{F}$.

Since the matrix $\mathbf{K}$ is symmetric positive definite, the linear system (3.1.4) can be solved efficiently both by direct methods, for example using Cholesky factorization, or by iterative methods like the conjugate gradient method.

3.3 Numerical experiments

3.3.1 The Poisson problem on the unit square

We now want to test the numerical solver on problems where we know the exact solution. A simple case is when Ω is the unit square $[0, 1]^2$ given as a NURBS surface by the

knot vectors $\Xi = \mathcal{H} = [0, 0, 1, 1]$, the polynomial orders $p = q = 1$, the control points $[\mathbf{B}_{1,1}, \mathbf{B}_{2,1}, \mathbf{B}_{1,2}, \mathbf{B}_{2,2}] = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ and the weights $w_{i,j} = 1$. The functions f and h are given by

$$f(x, y) = 2\pi^2 \sin(\pi x) \sin(\pi y)$$
$$h(x, y) = -\pi \sin(\pi x) \cos(\pi y)$$

and Γ_N is the part of the boundary where $y = 0$. In this case the exact solution is given by $u(x, y) = \sin \pi x \sin \pi y$. By inserting the knot values i/N for $i = 1, \ldots, N-1$, $N \in \mathbb{N}$, in both Ξ and $\mathcal{H}$ we obtain a grid of N^2 equally sized elements. The numerical solution and the error for $N = 5$ is shown in Figure 3.3.1.

If we now want a better approximation we can use the three refinement techniques of isogeometric analysis, h-, p- and k-refinement. We can move around in the so called hpk-refinement space with h^{-1}, p and k along the axes, where h is a length parameter describing the size of the elements, p is the polynomial order of the basis and k is the number of continuous derivatives at the knots. Note that the level of continuity k is limited by the polynomial order by $k \leq p - 1$. We have actually one hpk-space for each dimension, but we choose to have the same values of h, p and k in both directions. In this case the element size is naturally described by $h = 1/N$, and since we inserted single knots into a C^0-basis of polynomial orders $p = q = 1$, the solution shown in Figure 3.3.1 is at the refinement level $h^{-1} = 5$, $p = 1$ and $k = 0$. We can perform h-refinement by choosing $N = 25$ instead and thus get smaller elements, this results in the solution and the error shown in Figure 3.3.2. In p-refinement we insert knots with $N = 5$ and then use order elevation to increase the polynomial orders, the resulting solution and the error when the orders are increased to $p = q = 3$ is given in Figure 3.3.3, the basis is still C^0 at the knots since the order elevation preserves the lack of continuity. If we instead increase the order to $p = q = 3$ before we insert the knots with $N = 5$ we have k-refinement, in which case the basis becomes C^2-continuous at the knots. The resulting solution and the error after this k-refinement is shown in Figure 3.3.4. In this way we have enforced C^2 continuity of the numerical solution and we notice from the plot of the error that it appears to be smoother than the error plot in Figure 3.3.3.

We note that the number of degrees of freedom, or the number of linear equations, n_{eq} are 20, 600, 210 and 42 in the respective cases, which illustrates the fact that k-refinement increases the number of equations considerably less than pure p-refinement. Since the convergence rate of the method only depends on the polynomial degree and not the continuity [7, Ch. 3.B], this illustrates a great advantage with isogeometric analysis compared to traditional FEA: It can converge at the same rate with a lot less unknowns.

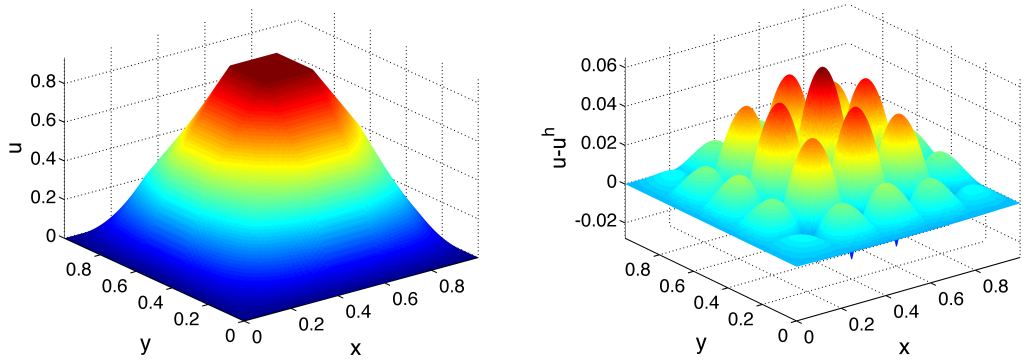

Figure 3.3.1: Poisson problem on the unit square: The numerical solution (left) and the error (right) for the refinement level $h^{-1} = 5$, $p = 1$ and $k = 0$.

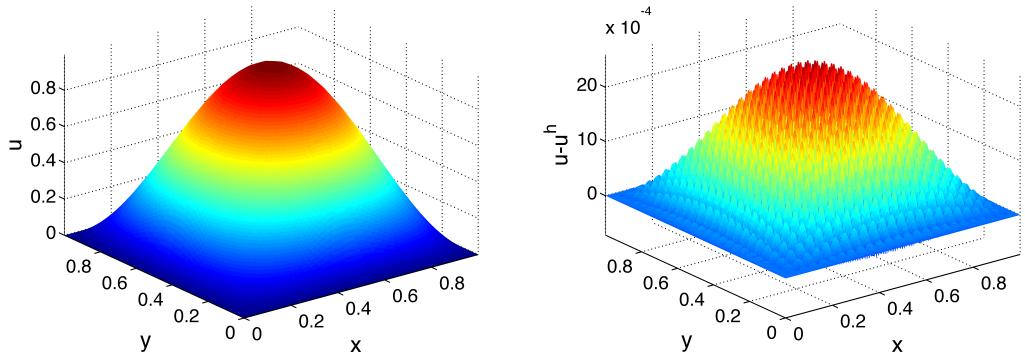

Figure 3.3.2: Poisson problem on the unit square: The numerical solution (left) and the error (right) for the refinement level $h^{-1} = 25$, $p = 1$ and $k = 0$.

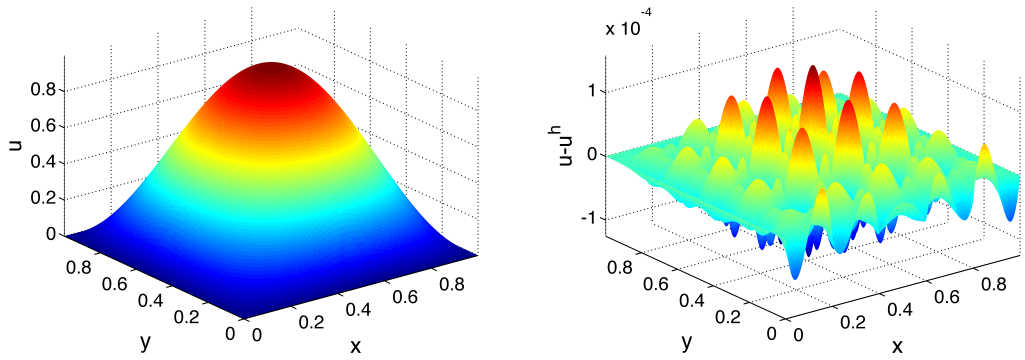

Figure 3.3.3: Poisson problem on the unit square: The numerical solution (left) and the error (right) for the refinement level $h^{-1} = 5$, $p = 3$ and $k = 0$.

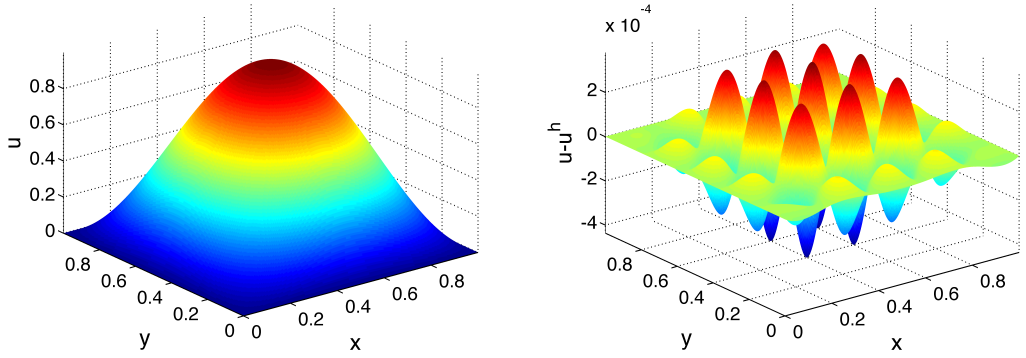

Figure 3.3.4: Poisson problem on the unit square: The numerical solution (left) and the error (right) for the refinement level $h^{-1} = 5$, $p = 3$ and $k = 2$.

3.3.2 The Poisson problem on an annular NURBS surface

We would also like to test the solver on a slightly more exotic domain. To do this we let Ω be given by the NURBS geometry in Figure 2.5.1 from Section 2.5.2. The functions f and h are given by

$$f(x,y) = 4\pi^2\left(x^2+y^2\right)\sin\left(\pi\left(x^2+y^2\right)\right) - 4\pi\cos\left(\pi\left(x^2+y^2\right)\right)$$
$$h(x,y) = 0$$

and Γ_N is the part of the boundary where $y = 0$, i.e. at the straight edges. In this case the exact solution is given by $u(x,y) = \sin\left(\pi\left(x^2+y^2\right)\right)$. We insert knots such that the knot spans are 0.1 in the parameter space for both Ξ and $\mathcal{H}$, and then elevate the order such that $p = q = 2$. The resulting numerical solution and the error are shown in Figure 3.3.5.

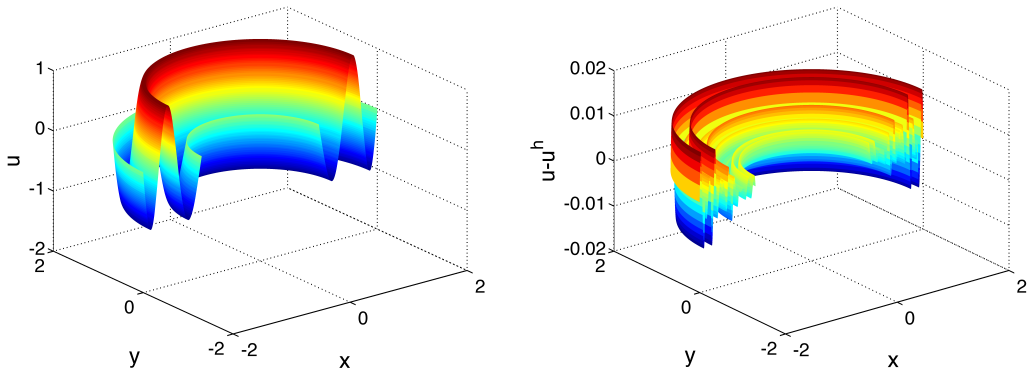

Figure 3.3.5: Poisson problem on a NURBS surface: The numerical solution (left) and the error (right).

Chapter 4

The Stokes Problem

We now consider the two-dimensional Stokes system which gives a model for the flow of an incompressible fluid with constant viscosity μ for which the viscous forces dominate over the inertial forces [20]. It should be noted that the method described in this chapter generalizes to the three-dimensional case [11], however the concepts are very similar so for simplicity we only consider the two-dimensional case in this thesis.

4.1 Discretizing the problem

4.1.1 Strong form

Given a physical domain $\Omega \subset \mathbb{R}^2$ and a function $\mathbf{f} \in (L^2(\Omega))^2$ we want to find a velocity field $\mathbf{u} : \bar{\Omega} \to \mathbb{R}^2$ and a pressure field $p : \bar{\Omega} \to \mathbb{R}$ that satisfy

$$-\mu\nabla^2\mathbf{u} + \nabla p = \mathbf{f} \qquad \text{in } \Omega, \tag{4.1.1a}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega, \tag{4.1.1b}$$

$$\mathbf{u} = \mathbf{0} \qquad \text{on } \partial\Omega. \tag{4.1.1c}$$

We assume that the physical domain Ω is the image of the parametric domain $\hat{\Omega} = (0,1)^2$ under a piecewise smooth geometrical mapping $\mathbf{G} : \hat{\Omega} \to \Omega$ that has a piecewise smooth inverse, typically a B-spline or NURBS mapping. Thus the physical variables $\mathbf{x}$ is related to the parametric variables $\boldsymbol{\xi}$ by $\mathbf{x} = \mathbf{G}(\boldsymbol{\xi})$.

The condition (4.1.1b) is referred to as the incompressibility condition and it models the mass conservation of the fluid. The homogeneous Dirichlet boundary conditions given by (4.1.1c) are called *no-slip* boundary conditions in the context of fluid flow. We note that the pressure p can only be determined up to a constant. A common way to normalize the pressure is to require that the pressure has a zero average over Ω.

4.1.2 Weak form

To obtain the weak form of (4.1.1) we first multiply (4.1.1a) by an arbitrary vector-valued test function $\mathbf{v}$ using the scalar product, and integrate over Ω. This gives

$$-\mu \int_\Omega \mathbf{v} \cdot \nabla^2 \mathbf{u} d\Omega + \int_\Omega \mathbf{v} \cdot \nabla p d\Omega = \int_\Omega \mathbf{v} \cdot \mathbf{f} d\Omega,$$

where $\nabla^2 \mathbf{u}$ is a vector containing the Laplacian of the components of $\mathbf{u}$. Using Green's identity this can be written

$$\mu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} d\Omega - \int_\Omega p \operatorname{div} \mathbf{v} d\Omega = \int_\Omega \mathbf{v} \cdot \mathbf{f} d\Omega + \mu \int_{\partial\Omega} \mathbf{n} \cdot \nabla \mathbf{u} \cdot \mathbf{v} d\Gamma - \int_{\partial\Omega} p \left(\mathbf{v} \cdot \mathbf{n}\right) d\Gamma,$$

where $\nabla \mathbf{u}$ and $\nabla \mathbf{v}$ are second order tensors with columns being the gradients of the components. For a discussion on tensors and tensor algebra see e.g. [19, Ch. 4]. For the incompressibility condition (4.1.1b) we multipy by another test function q, and integrate over Ω, obtaining

$$-\int_\Omega q \operatorname{div} \mathbf{u} d\Omega = 0.$$

Assuming $\mathbf{v}|_{\partial\Omega} = \mathbf{0}$ we end up with

$$\begin{aligned} &\mu \int_\Omega \nabla \mathbf{u} : \nabla \mathbf{v} d\Omega - \int_\Omega p \operatorname{div} \mathbf{v} d\Omega = \int_\Omega \mathbf{v} \cdot \mathbf{f} d\Omega, \\ &-\int_\Omega q \operatorname{div} \mathbf{u} d\Omega = 0. \end{aligned}$$

These calculations will hold for any test functions $\mathbf{v} \in \mathcal{V}_0$ and $q \in \mathcal{Q}_0$, where

$$\begin{aligned} \mathcal{V}_0 = \mathbf{H}_0^1(\Omega) &:= \left\{ \mathbf{v} \in \left(H^1(\Omega)\right)^2 : \mathbf{v}|_{\partial\Omega} = \mathbf{0} \right\} \\ \mathcal{Q}_0 = L_0^2(\Omega) &:= \left\{ q \in L^2(\Omega) : \int_\Omega q d\Omega = 0 \right\}. \end{aligned}$$

Using these test function spaces as search spaces, we want to find $\mathbf{u} \in \mathcal{V}_0$ and $p \in \mathcal{Q}_0$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = f(\mathbf{v}), \qquad \forall \mathbf{v} \in \mathcal{V}_0, \tag{4.1.2a}$$

$$b(\mathbf{u}, q) = 0, \qquad \forall q \in \mathcal{Q}_0, \tag{4.1.2b}$$

where we have defined the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ by

$$a(\mathbf{u},\mathbf{v}) = \mu \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v} d\Omega,$$
$$b(\mathbf{v},q) = -\int_\Omega q \operatorname{div}\mathbf{v} d\Omega.$$

The problem (4.1.2) is a saddle point problem, for which the existence and uniqueness conditions are well known, see e.g. [14] which studies this particular case of the Stokes problem as an example. With $\mathcal{V}_0 = \mathbf{H}_0^1(\Omega)$ and $\mathcal{Q}_0 = L_0^2(\Omega)$ the problem (4.1.2) does indeed have a unique solution $(\mathbf{u}, p)$.

4.1.3 The discrete problem

We finally obtain the discretized problem by letting $\mathcal{V}_0^h$ and $\mathcal{Q}_0^h$ be finite dimensional subspaces of $\mathcal{V}_0$ and $\mathcal{Q}_0$ respectively, depending on a length parameter h, and we want to find $\mathbf{u}^h \in \mathcal{V}_0^h$ and $p^h \in \mathcal{Q}_0^h$ such that

$$a\left(\mathbf{u}^h,\mathbf{v}^h\right) + b\left(\mathbf{v}^h,p^h\right) = f\left(\mathbf{v}^h\right), \qquad \forall \mathbf{v}^h \in \mathcal{V}_0^h, \tag{4.1.3a}$$
$$b\left(\mathbf{u}^h,q^h\right) = 0, \qquad \forall q^h \in \mathcal{Q}_0^h. \tag{4.1.3b}$$

Now if we have bases $\{\mathbf{N}_A\}_{A=1}^{n_{eq}^{\mathbf{u}}}$ and $\{N_C\}_{C=1}^{n_{eq}^{p}}$ for the spaces $\mathcal{V}_0^h$ and $\mathcal{Q}_0^h$ respectively, we can write $\mathbf{u}^h \in \mathcal{V}_0^h$ and $p^h \in \mathcal{Q}_0^h$ uniquely in terms of coefficients U_A and P_C as

$$\mathbf{u}^h = \sum_{A=1}^{n_{eq}^{\mathbf{u}}} \mathbf{N}_A U_A \qquad\qquad p^h = \sum_{C=1}^{n_{eq}^{p}} N_C P_C.$$

Since $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are bilinear we only need (4.1.3) to hold for all the basis functions and we obtain the equivalent system of linear equations

$$\sum_{A=1}^{n_{eq}^{\mathbf{u}}} a(\mathbf{N}_A,\mathbf{N}_{\tilde{A}})U_A + \sum_{C=1}^{n_{eq}^{p}} b(\mathbf{N}_{\tilde{A}},N_C)P_C = f(\mathbf{N}_{\tilde{A}}), \qquad \tilde{A} = 1,\dots,n_{eq}^{\mathbf{u}},$$
$$\sum_{A=1}^{n_{eq}^{\mathbf{u}}} b(\mathbf{N}_A,N_{\tilde{C}})U_A = 0, \qquad \tilde{C} = 1,\dots,n_{eq}^{p}.$$

If we define the matrices $\mathbf{A} = [a(\mathbf{N}_A,\mathbf{N}_{\tilde{A}})]_{\tilde{A},A=1}^{n_{eq}^{\mathbf{u}}}$ and $\mathbf{B} = [b(\mathbf{N}_A,N_{\tilde{C}})]_{\tilde{C},A=1}^{n_{eq}^{p},n_{eq}^{\mathbf{u}}}$, and the vectors $\mathbf{F} = [f(\mathbf{N}_{\tilde{A}})]_{\tilde{A}=1}^{n_{eq}^{\mathbf{u}}}$, $\mathbf{U} = [U_{\tilde{A}}]_{\tilde{A}=1}^{n_{eq}^{\mathbf{u}}}$ and $\mathbf{P} = [P_{\tilde{C}}]_{\tilde{C}=1}^{n_{eq}^{p}}$, we can write this in the compact form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}. \tag{4.1.4}$$

The conditions for existence and uniqueness of the discrete system are discussed in e.g. [14]. An important condition guaranteeing the stability of the discrete system is the *inf-sup condition*, or the *Babuška-Brezzi condition* [2, 5], given by

$$\inf_{q\in\mathcal{Q}_0^h\setminus\{0\}}\sup_{v\in\mathcal{V}_0^h\setminus\{\mathbf{0}\}}\frac{b\left(\mathbf{v},q\right)}{\|q\|_{L^2}\|\mathbf{v}\|_{\mathbf{H}^1}}\geq\beta>0,$$

where β is a constant independent of the parameter h. Finite elements satisfying this conditions are called *stable finite elements*. In general the velocity solution $\mathbf{u}^h$ of (4.1.3) will not satisfy the incompressibility condition div $\mathbf{u}^h = 0$ exactly. This will however be the case if we have that

$$\left\{\text{div }\mathbf{v} : \mathbf{v}\in\mathcal{V}_0^h\right\}\subseteq\mathcal{Q}_0^h.$$

Then div $\mathbf{u}^h$ will always be an element of $\mathcal{Q}_0^h$, which by (4.1.3b) is orthogonal to all other elements of $\mathcal{Q}_0^h$, and hence div $\mathbf{u}^h = 0$ since $\mathcal{Q}_0^h$ is a Hilbert space.

Different choices of velocity/pressure-spaces $(\mathcal{V}_0^h, \mathcal{Q}_0^h)$ give rise to different finite elements. A popular choice is the Taylor-Hood velocity/pressure pair [14]. This will give finite element spaces that are inf-sup stable, but the discrete velocity will not be exactly divergence-free. NURBS generalizations of the Taylor-Hood elements have been studied and tested in [3] and [6] in the context of isogeometric analysis. In this thesis we will consider a family of velocity/pressure-pairs $(\mathcal{V}_0^h, \mathcal{Q}_0^h)$ that will both be inf-sup stable and give velocity approximations that are pointwise divergence-free. These velocity/pressure pairs were proposed in [6] and their stability and convergence analysis is given in [11] for the more general case of the Darcy-Stokes-Brinkman equations or generalized Stokes equations.

4.1.4 Velocity and pressure spaces on the parametric domain

We start the construction of our finite element search and test spaces by defining the velocity/pressure-pair on the parametric domain $\hat{\Omega}$, first without boundary conditions. Being in the context of isogeometric analysis we let the finite element spaces be spline spaces. More specifically, given two partitions $\boldsymbol{\zeta}$ and $\boldsymbol{\gamma}$ of the interval $(0, 1)$, nonnegative integers p, q and continuity sequences $\boldsymbol{\alpha}, \boldsymbol{\beta}$ according to the requirements in Definition 2.3, our choice of parametric velocity/pressure spaces is given by

$$\hat{\mathcal{V}}^h := S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma})\times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}(\boldsymbol{\zeta},\boldsymbol{\gamma}), \tag{4.1.5a}$$

$$\hat{\mathcal{Q}}^h := S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma}). \tag{4.1.5b}$$

The construction of the spline spaces is detailed in Section 2.2. We note that for $p = q$ and $|\boldsymbol{\alpha}| = |\boldsymbol{\beta}| = -1$ the space $\hat{\mathcal{V}}^h$ gives the Raviart-Thomas element [27]. However such a choice of $\hat{\mathcal{V}}^h$ does not give an H^1-conforming approximation, since the

velocity space will be discontinuous, and it is therefore not a popular choice for the Stokes problem. In this thesis we will only consider conforming elements, so we assume $|\boldsymbol{\alpha}|, |\boldsymbol{\beta}| \geq 0$. The pair (4.1.5) can also be referred to as compatible B-spline spaces, because they have the same mathematical structure as the spaces they approximate. In the same way that the divergence operator maps $\mathbf{H}^1(\hat{\Omega})$ continuously onto $L^2(\hat{\Omega})$, it maps $\hat{\mathcal{V}}^h$ continuously onto $\hat{\mathcal{Q}}^h$ as seen in Proposition 8. This preservation of structure is explained thoroughly in [12].

In order to maintain both stability and pointwise mass conservation when imposing boundary conditions, it is essential that we have the relation

$$\left\{\text{div } \mathbf{v} : \mathbf{v} \in \widehat{\mathcal{V}}_0^h\right\} = \widehat{\mathcal{Q}}_0^h$$

for the discrete velocity/pressure pair $(\widehat{\mathcal{V}}_0^h, \widehat{\mathcal{Q}}_0^h)$ satisfying the boundary conditions. We should therefore choose the pressure space to be the image of the velocity space under the divergence operator. In the case of slip boundary conditions, i.e. $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega$, this is no problem. Indeed we have that

$$\left\{\text{div } \mathbf{v} : \mathbf{v} \in \widehat{\mathcal{V}}^h, \mathbf{v} \cdot \mathbf{n}|_{\partial\hat{\Omega}} = 0\right\} = \left\{ q \in \widehat{\mathcal{Q}}^h : \int_{\hat{\Omega}} q d\hat{\Omega} = 0 \right\}.$$

The proof of this relation can be found in [6]. Thus we choose in this case the velocity space $\mathcal{V}_0^h$ to be $\left\{\mathbf{v} \in \widehat{\mathcal{V}}^h : \mathbf{v} \cdot \mathbf{n}|_{\partial\hat{\Omega}} = 0\right\}$ and $\left\{ q \in \widehat{\mathcal{Q}}^h : \int_{\hat{\Omega}} q d\hat{\Omega} = 0 \right\}$ as the pressure space $\mathcal{Q}_0^h$.

The case of no-slip boundary conditions is however more tricky. This is due to the relation

$$\left\{\text{div } \mathbf{v} : \mathbf{v} \in \widehat{\mathcal{V}}^h, \mathbf{v}|_{\partial\hat{\Omega}} = \mathbf{0}\right\} = \left\{ q \in \widehat{\mathcal{Q}}^h : \int_{\hat{\Omega}} q d\hat{\Omega} = 0, q(\mathbf{x}_i) = 0, i = 1, \ldots, 4 \right\}, \tag{4.1.6}$$

where $\mathbf{x}_i$ are the corners of the parametric domain $\hat{\Omega}$, again the proof is given in [6]. This relation suggests that we choose a pressure space of functions that are zero at the corners of the parametric domain. There is of course no reason why the exact pressure solution of (4.1.2) should be zero at the corners of $\hat{\Omega}$, and hence this choice of discrete pressure space will restrict the accuracy of the pressure solution in the L^2-norm to first order. In [6] two different solutions to this problem are proposed, one of which uses T-splines to construct a compatible pressure space. Another way to deal with this problem is to impose the no-slip boundary conditions weakly using Nitsche's penalty method, this is the approach taken in [11]. We then use the discrete velocity space $\left\{\mathbf{v} \in \widehat{\mathcal{V}}^h : \mathbf{v} \cdot \mathbf{n}|_{\partial\hat{\Omega}} = 0\right\}$ as in the case of slip boundary conditions, but the bilinear form $a(\cdot, \cdot)$ is modified using

a penalty parameter in a way that preserves consistency and stability. More on weak imposition of Dirichlet boundary conditions can be found in [4].

In this thesis we have only implemented the straightforward case suggested by the relation (4.1.6). Thus we use the discrete velocity/pressure pair $(\mathcal{V}_0^h, \mathcal{Q}_0^h)$ given by

$$\widehat{\mathcal{V}}_0^h := \left\{ \mathbf{v} \in \widehat{\mathcal{V}}^h : \mathbf{v}|_{\partial\hat{\Omega}} = \mathbf{0} \right\}, \tag{4.1.7a}$$

$$\widehat{\mathcal{Q}}_0^h := \left\{ q \in \widehat{\mathcal{Q}}^h : \int_{\hat{\Omega}} q d\hat{\Omega} = 0, q(\mathbf{x}_i) = 0, i = 1, \ldots, 4 \right\}. \tag{4.1.7b}$$

4.1.5 Velocity and pressure spaces on the physical domain

We now want to construct compatible discrete velocity and pressure spaces $(\mathcal{V}^h, \mathcal{Q}^h)$ on the physical domain Ω in such a way that the divergence operator maps $\mathcal{V}^h$ continuously onto $\mathcal{Q}^h$. This is done by mapping the parametric spaces $\widehat{\mathcal{V}}^h$ and $\widehat{\mathcal{Q}}^h$ in the way that is done in [11]. This means that we define our discrete velocity and pressure spaces by

$$\mathcal{V}^h := \left\{ \left(\frac{(D\mathbf{G})\,\hat{\mathbf{v}}}{\det(D\mathbf{G})} \right) \circ \mathbf{G}^{-1} : \hat{\mathbf{v}} \in \widehat{\mathcal{V}}^h \right\}, \tag{4.1.8a}$$

$$\mathcal{Q}^h := \left\{ \left(\frac{\hat{q}}{\det(D\mathbf{G})} \right) \circ \mathbf{G}^{-1} : \hat{q} \in \widehat{\mathcal{Q}}^h \right\}, \tag{4.1.8b}$$

where $D\mathbf{G}$ denotes the Jacobian matrix of the geometrical mapping $\mathbf{G}$. We map the velocity and pressure spaces with boundary conditions in exactly the same way, so we define

$$\mathcal{V}_0^h := \left\{ \left(\frac{(D\mathbf{G})\,\hat{\mathbf{v}}}{\det(D\mathbf{G})} \right) \circ \mathbf{G}^{-1} : \hat{\mathbf{v}} \in \widehat{\mathcal{V}}_0^h \right\}, \tag{4.1.9a}$$

$$\mathcal{Q}_0^h := \left\{ \left(\frac{\hat{q}}{\det(D\mathbf{G})} \right) \circ \mathbf{G}^{-1} : \hat{q} \in \widehat{\mathcal{Q}}_0^h \right\}. \tag{4.1.9b}$$

These spaces will satisfy the necessary boundary conditions on the physical domain. Indeed for corresponding elements $\mathbf{v} \in \mathcal{V}_0^h$ and $\hat{\mathbf{v}} \in \widehat{\mathcal{V}}_0^h$ we have that $\mathbf{v}|_{\partial\Omega} = 0$ if and only if $\hat{\mathbf{v}}|_{\partial\hat{\Omega}} = 0$ since the geometrical mapping $\mathbf{G}$ is invertible. Also, we have for corresponding elements $q \in \mathcal{Q}_0^h$ and $\hat{q} \in \widehat{\mathcal{Q}}_0^h$ that

$$\int_{\Omega} q d\Omega = \int_{\hat{\Omega}} \frac{\hat{q}}{\det(D\mathbf{G})} \left| \det(D\mathbf{G}) \right| d\hat{\Omega},$$

so assuming the Jacobian determinant has no sign changes in $\hat{\Omega}$ then $\int_\Omega q d\Omega = 0$ if and only if $\int_{\hat{\Omega}} \hat{q} d\hat{\Omega} = 0$.

We note that the mapping used in (4.1.8a) and (4.1.9a) to map the velocity spaces is the Piola transform. The reason for using this Piola mapping is that it preserves the divergence of the vector field in the following way:

Proposition 9. *Let $\hat{\mathbf{v}} : \hat{\Omega} \to \mathbb{R}^2$ be a vector field on the parametric domain and let $\mathbf{v} : \Omega \to \mathbb{R}^2$ be defined by the Piola mapping*

$$\mathbf{v} = \left(\frac{(D\mathbf{G})\,\hat{\mathbf{v}}}{\det(D\mathbf{G})} \right) \circ \mathbf{G}^{-1}, \tag{4.1.10}$$

where $\mathbf{G} : \hat{\Omega} \to \Omega$ is the transformation from the parametric variables $\boldsymbol{\xi} = (\xi, \eta)$ to the physical variables $\mathbf{x} = (x, y)$. We then have that

$$\operatorname{div} \mathbf{v} = \frac{1}{\det(D\mathbf{G})} \widehat{\operatorname{div}}\, \hat{\mathbf{v}}.$$

Here div *and* $\widehat{\operatorname{div}}$ *denote the divergence operators with respect to physical and parametric variables respectively.*

Proof. The relation (4.1.10) can be written

$$\begin{aligned} \hat{\mathbf{v}} &= \det(D\mathbf{G})\,(D\mathbf{G})^{-1}\,(\mathbf{v} \circ \mathbf{G}), \\ \begin{bmatrix} \hat{v}_1 \\ \hat{v}_2 \end{bmatrix} &= \begin{bmatrix} \frac{\partial y}{\partial \eta} v_1 - \frac{\partial x}{\partial \eta} v_2 \\ -\frac{\partial y}{\partial \xi} v_1 + \frac{\partial x}{\partial \xi} v_2 \end{bmatrix}, \end{aligned}$$

and direct computation using the chain rule gives

$$\begin{aligned} \frac{\partial \hat{v}_1}{\partial \xi} &= \frac{\partial^2 y}{\partial \xi \partial \eta} v_1 - \frac{\partial^2 x}{\partial \xi \partial \eta} v_2 + \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial v_1}{\partial x} + \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial v_1}{\partial y} - \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial v_2}{\partial x} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\frac{\partial v_2}{\partial y}, \\ \frac{\partial \hat{v}_2}{\partial \eta} &= -\frac{\partial^2 y}{\partial \eta \partial \xi} v_1 + \frac{\partial^2 x}{\partial \eta \partial \xi} v_2 - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\frac{\partial v_1}{\partial x} - \frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial v_1}{\partial y} + \frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial v_2}{\partial x} + \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial v_2}{\partial y}. \end{aligned}$$

Hence we arrive at

$$\widehat{\operatorname{div}}\, \hat{\mathbf{v}} = \frac{\partial \hat{v}_1}{\partial \xi} + \frac{\partial \hat{v}_2}{\partial \eta} = \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \right) \left(\frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y} \right) = \det(D\mathbf{G})\,(\operatorname{div} \mathbf{v}),$$

which completes the proof. □

As an immediate consequence of Proposition 9 and (4.1.6) we have that

$$\left\{ \operatorname{div} \mathbf{v} : \mathbf{v} \in \mathcal{V}_0^h \right\} = \mathcal{Q}_0^h$$

when the pressure space $\mathcal{Q}_0^h$ is defined as in (4.1.9b). Thus we obtain both stability and pointwise mass conservation on the physical domain Ω.

4.2 Implementation of numerical solver

Recall that our parametric velocity/pressure pair $(\widehat{\mathcal{V}}^h, \widehat{\mathcal{Q}}^h)$ was defined in (4.1.5) by

$$\begin{aligned}\widehat{\mathcal{V}}^h &= S^{p+1,q}_{\boldsymbol{\alpha}+1,\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma}) \times S^{p,q+1}_{\boldsymbol{\alpha},\boldsymbol{\beta}+1}(\boldsymbol{\zeta},\boldsymbol{\gamma}),\\ \widehat{\mathcal{Q}}^h &= S^{p,q}_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\boldsymbol{\zeta},\boldsymbol{\gamma}).\end{aligned}$$

As seen in Section 2.3.2 we have a basis for $\widehat{\mathcal{V}}^h$ given by $\left\{N^{\boldsymbol{\alpha}+1}_{i,p+1}M^{\boldsymbol{\beta}}_{j,q}\mathbf{e}_1\right\}_{i,j=1}^{n+1,m} \cup \left\{N^{\boldsymbol{\alpha}}_{i,p}M^{\boldsymbol{\beta}+1}_{j,q+1}\mathbf{e}_2\right\}_{i,j=1}^{n,m+1}$, and a basis for $\widehat{\mathcal{Q}}^h$ given by $\left\{N^{\boldsymbol{\alpha}}_{i,p}M^{\boldsymbol{\beta}}_{j,q}\right\}_{i,j=1}^{n,m}$, using the notation introduced in Section 2.3.2. We introduce single indices A and C for the bases of $\widehat{\mathcal{V}}^h$ and $\widehat{\mathcal{Q}}^h$ in a similar way that we did in Section 3.2.1 for the Poisson equation by ordering the basis functions in the following way:

$$\begin{aligned}&\left\{\hat{\mathbf{N}}_A\right\}_{A=1}^{n^{\mathbf{u}}_{np}}\\ &\quad= \left\{N^{\alpha+1}_{1,p+1}M^{\beta}_{1,q}\mathbf{e}_1,\ldots,N^{\alpha+1}_{n+1,p+1}M^{\beta}_{1,q}\mathbf{e}_1,\ldots,N^{\alpha+1}_{1,p+1}M^{\beta}_{m,q}\mathbf{e}_1,\ldots,N^{\alpha+1}_{n+1,p+1}M^{\beta}_{m,q}\mathbf{e}_1,\right.\\ &\qquad\left. N^{\alpha}_{1,p}M^{\beta+1}_{1,q+1}\mathbf{e}_2,\ldots,N^{\alpha}_{n,p}M^{\beta+1}_{1,q+1}\mathbf{e}_2,\ldots,N^{\alpha}_{1,p}M^{\beta+1}_{m+1,q+1}\mathbf{e}_2,\ldots,N^{\alpha}_{n,p}M^{\beta+1}_{m+1,q+1}\mathbf{e}_2\right\},\\ &\left\{\hat{N}_C\right\}_{C=1}^{n^{p}_{np}}\\ &\quad= \left\{N^{\alpha}_{1,p}M^{\beta}_{1,q},\ldots,N^{\alpha}_{n,p}M^{\beta}_{1,q},\ldots,N^{\alpha}_{1,p}M^{\beta}_{m,q},\ldots,N^{\alpha}_{n,p}M^{\beta}_{m,q}\right\},\end{aligned}$$

where $n^{\mathbf{u}}_{np} = (n+1)m + n(m+1)$ and $n^{p}_{np} = nm$.

Now the physical velocity and pressure spaces $\mathcal{V}^h$ and $\mathcal{Q}^h$ were defined in (4.1.8) by transforming the spaces $\widehat{\mathcal{V}}^h$ and $\widehat{\mathcal{Q}}^h$ using invertible linear mappings. Hence we obtain bases for these spaces by transforming the parametric bases using the same mappings, i.e. the bases of $\mathcal{V}^h$ and $\mathcal{Q}^h$ are given by

$$\begin{aligned}\{\mathbf{N}_A\}_{A=1}^{n^{\mathbf{u}}_{np}} &= \left\{\left(\frac{(D\mathbf{G})\,\hat{\mathbf{N}}_A}{\det(D\mathbf{G})}\right)\circ\mathbf{G}^{-1}\right\}_{A=1}^{n^{\mathbf{u}}_{np}},\\ \{N_C\}_{C=1}^{n^{p}_{np}} &= \left\{\left(\frac{\hat{N}_C}{\det(D\mathbf{G})}\right)\circ\mathbf{G}^{-1}\right\}_{C=1}^{n^{p}_{np}}.\end{aligned}$$

Thus any $\mathbf{v}\in\mathcal{V}^h$ and any $q\in\mathcal{Q}^h$ can be uniquely represented by coefficients $\left\{V^h_A\right\}_{A=1}^{n^{\mathbf{u}}_{np}}$ and $\left\{Q^h_C\right\}_{C=1}^{n^{p}_{np}}$ as

$$\mathbf{v} = \sum_{A=1}^{n^{\mathbf{u}}_{np}}\mathbf{N}_A V^h_A, \qquad\qquad q = \sum_{C=1}^{n^{p}_{np}} N_C Q^h_C.$$

4.2.1 Assembling the linear system

With bases of the discrete velocity and pressure spaces at hand we can assemble the linear system (4.1.4). Note that even though the system (4.1.4) is defined using the bases of the spaces $\mathcal{V}_0^h$ and $\mathcal{Q}_0^h$ satisfying the boundary conditions, we will instead construct the matrices and vectors for the bases of $\mathcal{V}^h$ and $\mathcal{Q}^h$ and impose the boundary conditions later. This is the same as we did when implementing the Poisson solver. The assembly of the matrices will be done in much the same manner as for the Poisson equation, by creating element matrices using the knowledge of the support of the basis functions, and computing the integrals by Gauss quadrature. Apart from the fact that we here have three different bivariate spline spaces to handle simultaneously, the main difference from what we did for the Poisson equation will be that the basis functions on the physical domain are not mapped using pointwise mappings. We will therefore focus on the consequences of this to our implementation.

First we consider the matrix $\mathbf{B}$ whose entries can be written

$$B_{C,A} = b\left(\mathbf{N}_A, N_C\right) = -\int_\Omega N_C \mathrm{div}\left(\mathbf{N}_A\right) d\Omega = -\int_\Omega \frac{\hat{N}_C \widehat{\mathrm{div}}\left(\hat{\mathbf{N}}_A\right)}{\det\left(D\mathbf{G}\right)^2} d\Omega$$
$$= -\int_{\hat{\Omega}} \frac{\hat{N}_C \widehat{\mathrm{div}}\left(\hat{\mathbf{N}}_A\right)}{\left|\det\left(D\mathbf{G}\right)\right|} d\hat{\Omega},$$

using Proposition 9 and integration by substitution. Now for every A, $\widehat{\mathrm{div}}\left(\hat{\mathbf{N}}_A\right)$ can be represented as an element of $\widehat{\mathcal{Q}}^h$ using the method described in Section 2.3.2. If $\mathbf{D}$ is the matrix representation of the div operator as defined in (2.3.6) we have that the coefficients of $\widehat{\mathrm{div}}\left(\hat{\mathbf{N}}_A\right)$ will be contained in the A^{th} column of $\mathbf{D}$. This gives that

$$B_{\tilde{C},A} = \int_{\hat{\Omega}} \frac{\hat{N}_{\tilde{C}}}{\left|\det\left(D\mathbf{G}\right)\right|} \sum_{C=1}^{n_{np}^p} \hat{N}_C D_{C,A} d\hat{\Omega} = -\sum_{C=1}^{n_{np}^p} D_{C,A} \int_{\hat{\Omega}} \frac{\hat{N}_{\tilde{C}} \hat{N}_C}{\left|\det\left(D\mathbf{G}\right)\right|} d\hat{\Omega}. \tag{4.2.1}$$

Defining the *pressure mass matrix* $\mathbf{M}^p = \left[M^p_{C,\tilde{C}}\right]_{C,\tilde{C}=1}^{n_{np}^p}$ by

$$M^p_{C,\tilde{C}} = \int_\Omega N_C N_{\tilde{C}} d\Omega = \int_\Omega \frac{\hat{N}_C \hat{N}_{\tilde{C}}}{\det\left(D\mathbf{G}\right)^2} d\Omega = \int_{\hat{\Omega}} \frac{\hat{N}_C \hat{N}_{\tilde{C}}}{\left|\det\left(D\mathbf{G}\right)\right|} d\hat{\Omega}, \mathbf{N}_A$$

we can recognize that (4.2.1) describes the matrix product

$$\mathbf{B} = -\mathbf{M}^p \mathbf{D}.$$

Thus $\mathbf{B}$ can be calculated by computing the pressure mass matrix and the matrix representation $\mathbf{D}$ of the div operator.

In order to compute the matrix $\mathbf{A}$ given by the entries

$$A_{\tilde{A},A} = a\left(\mathbf{N}_A, \mathbf{N}_{\tilde{A}}\right) = \int_\Omega \nabla \mathbf{N}_A : \nabla \mathbf{N}_{\tilde{A}} d\Omega$$

we need to express the gradient matrices $\nabla \mathbf{N}_A$ by the parametric basis functions $\hat{\mathbf{N}}_A$ and their derivatives with respect to the parametric variables. Now if $\hat{\mathbf{v}} : \hat{\Omega} \to \mathbb{R}^2$ is a vector field on the parametric domain and $\mathbf{v} : \Omega \to \mathbb{R}^2$ is defined by the Piola mapping (4.1.10), then the chain rule gives

$$\nabla \mathbf{v} = (D\mathbf{G})^{-T} \hat{\nabla} \mathbf{v} = (D\mathbf{G})^{-T} \hat{\nabla} \left(\frac{D\mathbf{G}}{\det (D\mathbf{G})} \hat{\mathbf{v}} \right),$$

where $\hat{\nabla}$ is the gradient operator with respect to parametric variables. Carrying out the differentiation we find that this gives

$$\nabla \mathbf{v} = \frac{(D\mathbf{G})^{-T}}{\det (D\mathbf{G})} \left(\left(\hat{\nabla} \hat{\mathbf{v}} - \frac{\nabla \det (D\mathbf{G})}{\det (D\mathbf{G})} \hat{\mathbf{v}}^T \right) (D\mathbf{G})^T + \begin{bmatrix} (D^2 x)\, \hat{\mathbf{v}} & (D^2 y)\, \hat{\mathbf{v}} \end{bmatrix} \right),$$

where $D^2 x$ and $D^2 y$ are the Hessian matrices of the physical variables x and y with respect to the parametric variables. The gradient of the Jacobian determinant is given by

$$\nabla \det (D\mathbf{G}) = \begin{bmatrix} \frac{\partial^2 x}{\partial \xi^2} \frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \xi} \frac{\partial^2 y}{\partial \xi \partial \eta} - \frac{\partial^2 x}{\partial \xi \partial \eta} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \eta} \frac{\partial^2 y}{\partial \xi^2} \\ \frac{\partial^2 x}{\partial \xi \partial \eta} \frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \xi} \frac{\partial^2 y}{\partial \eta^2} - \frac{\partial^2 x}{\partial \eta^2} \frac{\partial y}{\partial \xi} - \frac{\partial x}{\partial \eta} \frac{\partial^2 y}{\partial \xi \partial \eta} \end{bmatrix}.$$

Thus the gradient matrices $\nabla \mathbf{N}_A$ are given by

$$\nabla \mathbf{N}_A = \frac{(D\mathbf{G})^{-T}}{\det (D\mathbf{G})} \left(\left(\hat{\nabla} \hat{\mathbf{N}}_A - \frac{\nabla \det (D\mathbf{G})}{\det (D\mathbf{G})} \hat{\mathbf{N}}_A^T \right) (D\mathbf{G})^T + \begin{bmatrix} (D^2 x)\, \hat{\mathbf{N}}_A & (D^2 y)\, \hat{\mathbf{N}}_A \end{bmatrix} \right).$$

The basis functions on the parametric domain and their derivatives can be computed in the same way as for the Poisson equation.

Note that if we solve a problem where $\Omega = \hat{\Omega}$, the assembly of $\mathbf{A}$ simplifies a lot, since the geometrical mapping $\mathbf{G}$ will be the identity mapping. Therefore a specialized solver should be implemented in such cases.

4.2.2 Imposing boundary conditions and solving the linear system

Recall that our discrete velocity and pressure spaces satisfying the boundary conditions are defined in (4.1.9) by mapping the parametric spaces given in (4.1.7) as

$$\widehat{\mathcal{V}}_0^h := \left\{ \mathbf{v} \in \widehat{\mathcal{V}}^h : \mathbf{v}|_{\partial\hat{\Omega}} = \mathbf{0} \right\},$$
$$\widehat{\mathcal{Q}}_0^h := \left\{ q \in \widehat{\mathcal{Q}}^h : \int_{\hat{\Omega}} q d\hat{\Omega} = 0, q(\mathbf{x}_i) = 0, i = 1, \ldots, 4 \right\},$$

where the points $\mathbf{x}_i$ are the corners of the parametric domain $\hat{\Omega}$.

Imposing homogeneous Dirichlet boundary conditions for the velocity is done in the same way as we did for the Poisson equation by identifying the basis functions $\hat{\mathbf{N}}_A$ that have non-zero values on the boundary, setting their coefficients equal to zero, and removing in $\mathbf{A}$ the corresponding rows and columns and in $\mathbf{B}$ the corresponding columns. For the pressure space we note that all the basis functions $\hat{N}_C$ will be zero at the corners $\mathbf{x}_i$ except the four that have indices $C = 1$, $C = n$, $C = n(m-1)+1$ and $C = nm$. We will therefore have to remove their corresponding rows in $\mathbf{B}$.

The last condition we need to impose on the discrete system is the zero mean value of the pressure, i.e. we require the discrete pressure solution p^h to satisfy

$$\int_\Omega p^h d\Omega = \int_\Omega \sum_{i=1}^{n_{np}^p} N_C P_C d\Omega = \sum_{i=1}^{n_{np}^p} P_C \int_\Omega N_C d\Omega = 0.$$

Defining the vector $\mathbf{J} = [\int_\Omega N_C d\Omega]_{C=1}^{n_{np}^p}$ this can be written $\mathbf{J}^T\mathbf{P} = 0$. Now if we have the system

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix} \tag{4.2.2}$$

before we have imposed the zero mean condition, this needs to be augmented using the equation $\mathbf{J}^T\mathbf{P} = 0$. Let C^* be any index into $\mathbf{J}$ (except those corresponding to the corner basis functions) and denote by J_{C^*} element number C^* in $\mathbf{J}$ and by $\mathbf{B}_{C^*}$ column number C^* in $\mathbf{B}^T$. Observing that $\mathbf{B}_{C^*}^T\mathbf{U} = 0$ is one of the equations in the system (4.2.2) and using that $\mathbf{J}^T\mathbf{P} = 0$ we have from (4.2.2) that

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T - \frac{1}{J_{C^*}}\mathbf{B}_{C^*}\mathbf{J}^T \\ \mathbf{B} - \frac{1}{J_{C^*}}\mathbf{J}\mathbf{B}_{C^*}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}. \tag{4.2.3}$$

In this system the variable P_{C^*} is eliminated since column number C^* of the matrix $\mathbf{B}^T - \frac{1}{J_{C^*}}\mathbf{B}_{C^*}\mathbf{J}^T$ will consist of only zeros. Hence we solve (4.2.3) for all unknown variables except P_{C^*} and then find P_{C^*} from the equation $\mathbf{J}^T\mathbf{P} = 0$.

After imposing the necessary conditions on our discrete system we end up with the system (4.1.4), i.e.

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}$$

where $\mathbf{A}$ is symmetric positive definite, and knowing that the system has a unique solution, the entire matrix must be symmetric positive definite. Thus the system can be solved efficiently by both direct and iterative methods. An efficient iterative method is the Uzawa algorithm, see e.g. [13], which utilizes the fact that the system can be decoupled into the equation $\left(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T\right)\mathbf{P} = \mathbf{B}\mathbf{A}^{-1}\mathbf{F}$ for the pressure unknowns and the equation $\mathbf{A}\mathbf{U} = \mathbf{F} - \mathbf{B}^T\mathbf{P}$ for the velocity unknowns.

4.3 Numerical experiments

We now want to verify the method by applying our solver to problems where the solution is known. We also solve benchmark problems that are known to cause dramatic failures for unstable discretizations. Recall from Section 4.1.4 that our method is specified by two partitions $\boldsymbol{\zeta}$ and $\boldsymbol{\gamma}$ of the interval $(0, 1)$, polynomial degrees p, q and regularity sequences $\boldsymbol{\alpha}, \boldsymbol{\beta}$. Since discrete solutions of high continuity is a key feature of isogeometric analysis, we will use discrete spaces of maximal continuity in all our tests. Using spline spaces with maximal continuity can be referred to as the k-method, and it has been shown that the k-method has better approximation properties than classical finite element methods using C^0 basis functions [10]. Maximal continuity is obtained using the regularity sequences

$$\begin{aligned} \boldsymbol{\alpha} &= \{-1, p-1, p-1, \ldots, p-1, p-1, -1\}, \\ \boldsymbol{\beta} &= \{-1, q-1, q-1, \ldots, q-1, q-1, -1\}. \end{aligned}$$

Also, unless otherwise specified, we use uniform partitions $\boldsymbol{\zeta} = \boldsymbol{\gamma} = \{i/N\}_{i=0}^{N}$ for some $N \in \mathbb{N}$. A natural length parameter describing the element sizes is then $h = 1/N$.

4.3.1 The Stokes problem on the unit square

We first test our solver on a manufactured solution presented in [6]. Consider the Stokes system (4.1.1) on the parametric domain $\Omega = \hat{\Omega} = (0, 1)^2$ with kinematic viscosity

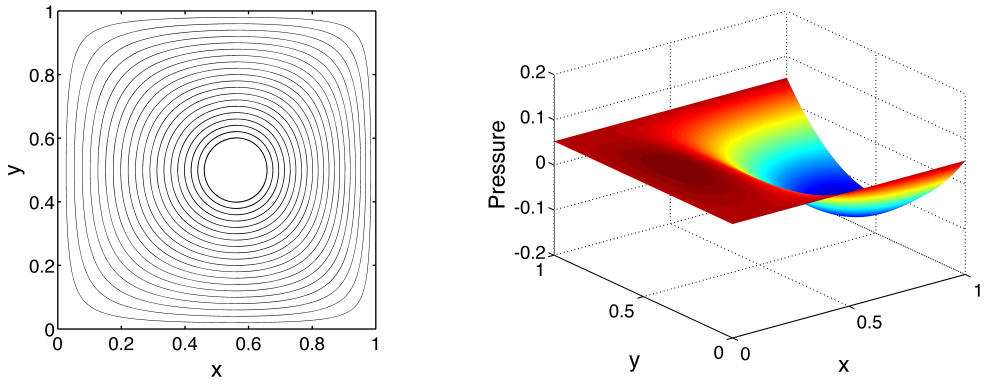

Figure 4.3.1: Stokes problem on the unit square: The streamlines of the exact velocity solution $\mathbf{u}$ (left) and the exact pressure p (right).

$\mu = 1$ and a load $\mathbf{f} = -\nabla^2 \bar{\mathbf{u}} + \nabla \bar{p}$ with

$$\bar{\mathbf{u}} = \begin{bmatrix} 2e^x x^2 y (x-1)^2 (y-1)(2y-1) \\ -e^x x y^2 (x-1)(y-1)^2 (x(x+3)-2) \end{bmatrix},$$

$$\bar{p} = -424 + 156e + \left(y^2 - y\right)\left(-456 + e^x\left(456 + x^2\left(228 - 5\left(y^2 - y\right)\right)\right.\right.$$
$$\left.\left. + 2x\left(y^2 - y - 228\right) + 2x^3\left(y^2 - y - 36\right) + x^4\left(y^2 - y + 12\right)\right)\right).$$

We impose no-slip boundary conditions and normalize the pressure by requiring a zero mean value over Ω. The exact solution is given by $(\mathbf{u}, p) = (\bar{\mathbf{u}}, \bar{p})$, for which the streamlines and pressure is shown in Figure 4.3.1.

In order to investigate the convergence rate of the method we have solved the problem using different values of h and calculated the $\mathbf{H}^1(\Omega)$-seminorm and the $\mathbf{L}^2(\Omega)$-norm of the velocity error $\mathbf{u} - \mathbf{u}^h$, as well as the $L^2(\Omega)$-norm of the pressure error $p - p^h$. We have also measured the $L^2(\Omega)$-norm of div $\mathbf{u}^h$ to verify that it is in fact exactly zero in all of Ω, at least up to floating point truncation errors. Table 4.1 shows the results for $p = q = 1$, $p = q = 2$ and $p = q = 3$ respectively, and as an example where $p \neq q$ it gives the results for $p = 2$ and $q = 3$. Figure 4.3.2 give logarithmic plots of the errors for the four cases, visualizing the results in the tables. We see that the order of convergence for the discrete velocity solution is the same as the polynomial degree of the discretization for the H^1-seminorm, and one more than the polynomial degree for the L^2-norm. Thus the results display optimal convergence rates for the discrete velocity solution. For the discrete pressure solution we observe that the order of convergence is restricted to one, regardless of polynomial degree. This is due to the fact that we enforce zero pressure at the corners when this clearly is not the case for the exact solution. Note however that there are several solutions to this problem, as explained in Section 4.1.4, though these are not implemented here. The tables also confirm that the divergence of

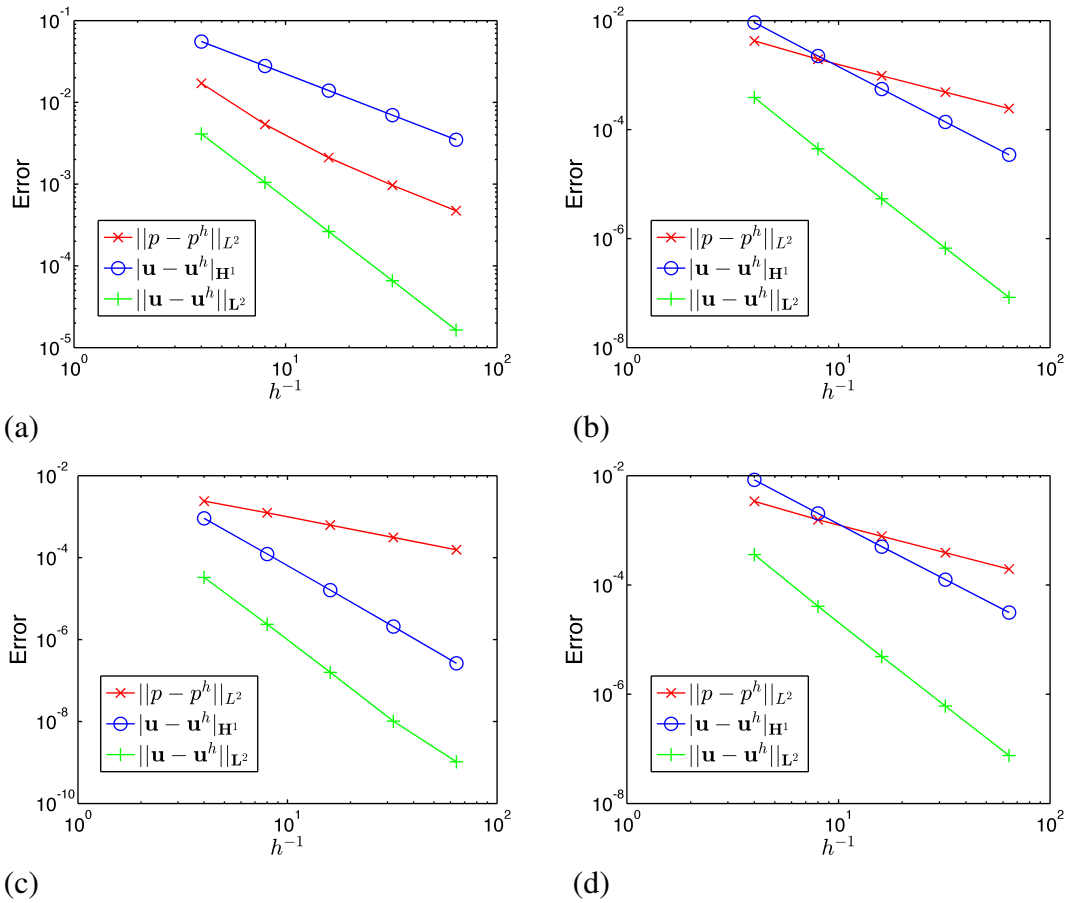

Figure 4.3.2: Stokes problem on the unit square: Convergence plots for different choices of polynomial degrees: (a) $p = q = 1$, (b) $p = q = 2$, (c) $p = q = 3$ and (d) $p = 2$, $q = 3$.

the velocity solution is zero apart from floating point truncations errors. We can observe that the value of $||\text{div } \mathbf{u}^h||_{L^2(\Omega)}$ roughly doubles as the number of degrees of freedom doubles, as would be expected for the error made due to floating point truncation.

4.3.2 The Stokes problem on an annular NURBS surface

We now want to test the method on a physical domain different from the parametric domain. Let Ω be the first quadrant part of an annulus centered at the origin with inner radius $r_i = 1$ and outer radius $r_o = 2$, described in polar coordinates by the inequalities $1 < r < 2$ and $0 < \theta < \pi/2$. The example NURBS mapping given in Section 2.5.2 is easily modified to obtain the geometrical mapping $\mathbf{G} : \hat{\Omega} \to \Omega$. In order to get a manufactured solution of (4.1.1) with no-slip boundary conditions on this domain we

Table 4.1: Stokes problem on the unit square: Convergence rates and norm of the divergence.

Polynomial degrees $p = q = 1$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	5.55e-2	2.79e-2	1.40e-2	6.98e-3	6.49e-3
order	-	0.99	1.00	1.00	1.00
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	4.11e-3	1.05e-3	2.63e-4	6.58e-5	1.65e-5
order	-	1.97	1.99	2.00	2.00
$\|\|p - p^h\|\|_{L^2(\Omega)}$	1.71e-2	5.38e-3	2.11e-3	9.68e-4	4.73e-4
order	-	1.67	1.35	1.12	1.03
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	2.39e-17	1.41e-17	2.42e-17	4.35e-17	8.27e-17

Polynomial degrees $p = q = 2$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	9.24e-3	2.24e-3	5.56e-4	1.39e-4	3.46e-5
order	-	2.04	2.01	2.00	2.00
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	3.87e-4	4.44e-5	5.40e-6	6.69e-7	8.35e-8
order	-	3.12	3.04	3.01	3.00
$\|\|p - p^h\|\|_{L^2(\Omega)}$	4.25e-3	1.95e-3	9.74e-4	4.87e-4	2.43e-4
order	-	1.12	1.00	1.00	1.00
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	6.64e-17	4.45e-17	5.49e-17	8.71e-17	1.72e-16

Polynomial degrees $p = q = 3$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	9.10e-4	1.23e-4	1.62e-5	2.09e-6	2.64e-7
order	-	2.89	2.92	2.96	2.98
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	3.28e-5	2.35e-6	1.59e-7	1.02e-8	1.03e-9
order	-	3.80	3.89	3.95	3.32
$\|\|p - p^h\|\|_{L^2(\Omega)}$	2.39e-3	1.24e-3	6.23e-4	3.11e-4	1.56e-4
order	-	0.95	1.00	1.00	1.00
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	9.00e-17	1.70e-16	8.79e-17	1.35e-16	2.48e-16

Polynomial degrees $p = 2$, $q = 3$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	8.41e-3	2.04e-3	5.03e-4	1.25e-4	3.13e-5
order	-	2.05	2.02	2.00	2.00
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	3.60e-4	4.06e-5	4.90e-6	6.06e-7	7.56e-8
order	-	3.15	3.05	3.01	3.00
$\|\|p - p^h\|\|_{L^2(\Omega)}$	3.40e-3	1.56e-3	7.79e-4	3.89e-4	1.95e-4
order	-	1.12	1.00	1.00	1.00
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	8.50e-17	6.15e-17	5.98e-17	9.61e-17	1.77e-16

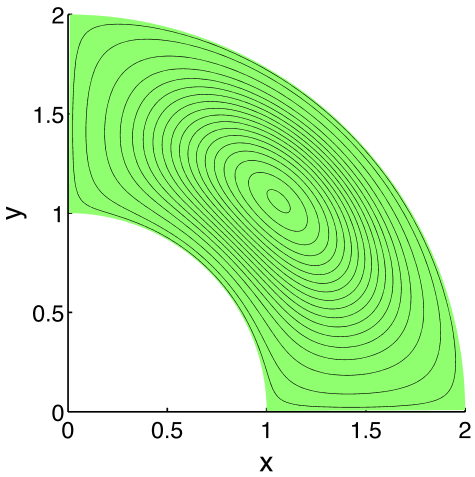

Figure 4.3.3: Stokes problem on a NURBS surface: The streamlines of the exact velocity solution $\mathbf{u}$ on the NURBS surface Ω.

construct a potential function ϕ and choose

$$\bar{\mathbf{u}} = \begin{bmatrix} \frac{\partial \phi}{\partial y} \\ -\frac{\partial \phi}{\partial x} \end{bmatrix}$$

as our exact velocity solution, thus satisfying the incompressibility condition div $\bar{\mathbf{u}} = 0$. The condition that $\bar{\mathbf{u}}|_{\partial\Omega} = \mathbf{0}$ will then mean that the gradient of ϕ must be zero on the boundary, which will be the case if the derivative in two linearly independent directions is zero. We therefore construct a potential ϕ which is constant on $\partial\Omega$ and has a zero normal derivative there. Using polar coordinates this is not too hard, and we choose here the potential

$$\phi\left(r,\theta\right) = (r-1)^2(2-r)^2\left(1-\cos 4\theta\right),$$

which gives in Cartesian coordinates

$$\phi\left(x,y\right) = 8x^2y^2\left(1-\frac{1}{\sqrt{x^2+y^2}}\right)^2\left(\frac{2}{\sqrt{x^2+y^2}}-1\right)^2.$$

Now we can choose any pressure $\bar{p}$ having a zero average on Ω, and obtain our right hand side by $\mathbf{f} = -\nabla^2\bar{\mathbf{u}} + \nabla\bar{p}$, here we have used the zero pressure $\bar{p} = 0$. With this choice of $\mathbf{f}$ the solution to problem (4.1.1) with no-slip boundary conditions is $(\mathbf{u}, p) = (\bar{\mathbf{u}}, \bar{p})$. The domain Ω and the velocity streamlines of the exact solution $\bar{\mathbf{u}}$ is shown in Figure 4.3.3.

In the same way as we did for the Stokes problem on the unit square we have calculated the norms of the errors for decreasing element size h and for different polynomial

degrees p and q. Table 4.2 shows the results for the cases $p = q = 1$, $p = q = 2$, $p = q = 3$ and $p = 2$, $q = 3$. Figure 4.3.4 gives a visualization of the numbers by showing convergence plots for the different cases. For the velocity solutions we observe the same optimal convergence rates. However, for the pressure solution we do not have the same first order convergence as we had in Section 4.3.1, this is of course because the exact solution p happens to be zero at the points corresponding to the four corners of the parametric domain. We observe that the order of convergence for the pressure hints toward the optimal order of one more than the polynomial degree of the discretization, though we get even better approximations due to the rather banal choice of $\bar{p} = 0$ as manufactured solution. We could of course choose another $\bar{p}$, but since "guessing" the correct value at the corner is quite unrealistic anyway we focus more on the obtained velocity solution. As for the divergence of the discrete velocity we observe the same behaviour as before, and conclude that it is zero apart from floating point truncation errors.

4.3.3 Lid-driven cavity flow

We also want to investigate the performance of the method by solving the lid-driven cavity flow benchmark problem which is known to cause dramatic failure in unstable formulations [3]. Consider the Stokes problem (4.1.1) on the unit square, i.e. $\Omega = \hat{\Omega} = (0, 1)^2$, with $\mathbf{f} = \mathbf{0}$ and no-slip boundary conditions on all edges except the top edge where $y = 1$, here we impose that the velocity is directed in the positive x-direction with unitary length. This models a two-dimensional Stokes flow in a rectangular cavity driven only by the motion of the lid of the cavity. The non-homogeneous Dirichlet boundary conditions are imposed in the usual way using a lifting function.

Figure 4.3.5 shows the streamlines of the numerical velocity solution for different levels of refinement. We see that the solutions appear stable, showing no spurious oscillations. The method also captures the characteristic Moffatt eddies [23] in the lower corners already for quite coarse meshes using polynomial degree $p = q = 2$, for the coarsest mesh and lowest polynomial degree we see that these are lost. The Moffatt eddies actually appear in infinite sequences of eddies of rapidly decreasing size [23]. However, taking a closer look at the lower left corner in Figure 4.3.6, we see no traces of the second eddy which would have appeared even closer to the corner. Thus we need an even finer discretization to reveal the second eddy. This is a typical example where methods for local refinement would be ideal, but that is beyond the scope of this thesis.

4.3.4 Two-sided lid-driven cavity flow

We also consider a version of lid-driven cavity flow where two edges are moving. Consider the Stokes problem (4.1.1) with $\Omega = (-1, 1) \times (-3, 3)$ and $\mathbf{f} = \mathbf{0}$ but with no-slip boundary conditions only on the vertical edges of the domain where $x = -1$ and $x = 1$,

Table 4.2: Stokes problem on a NURBS surface: Convergence rates and norm of the divergence.

Polynomial degrees $p = q = 1$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	9.43e-1	4.75e-1	2.38e-1	1.19e-1	5.94e-2
order	-	0.99	1.00	1.00	1.00
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	7.62e-2	1.91e-2	4.79e-3	1.20e-3	2.99e-4
order	-	1.99	2.00	2.00	2.00
$\|\|p - p^h\|\|_{L^2(\Omega)}$	1.76e-1	4.39e-2	1.10e-2	2.72e-3	6.81e-4
order	-	2.00	2.01	2.00	2.00
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	3.96e-16	1.67e-16	3.02e-16	5.34e-16	1.11e-15

Polynomial degrees $p = q = 2$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	1.20e-1	2.81e-2	6.95e-3	1.73e-3	4.33e-4
order	-	2.10	2.02	2.00	2.00
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	7.42e-3	7.03e-4	8.17e-5	1.00e-5	1.25e-6
order	-	3.40	3.11	3.02	3.01
$\|\|p - p^h\|\|_{L^2(\Omega)}$	3.15e-2	8.95e-4	5.77e-5	4.29e-6	3.43e-7
order	-	5.14	3.96	3.75	3.65
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	6.89e-16	3.16e-16	5.41e-16	1.00e-15	1.71e-15

Polynomial degrees $p = q = 3$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	3.11e-2	1.57e-3	1.58e-4	1.87e-5	2.30e-6
order	-	4.31	3.31	3.08	3.03
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	3.70e-3	8.38e-5	3.83e-6	2.20e-7	2.96e-8
order	-	5.46	4.45	4.12	2.90
$\|\|p - p^h\|\|_{L^2(\Omega)}$	1.28e-2	5.07e-5	8.13e-7	2.16e-8	7.66e-10
order	-	7.98	5.96	5.23	4.82
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	8.47e-16	7.12e-16	1.11e-15	1.54e-15	3.06e-15

Polynomial degrees $p = 2$, $q = 3$.

h	1/4	1/8	1/16	1/32	1/64
$\|\mathbf{u} - \mathbf{u}^h\|_{\mathbf{H}^1(\Omega)}$	5.68e-2	9.91e-3	2.26e-3	5.54e-4	1.38e-4
order	-	2.52	2.13	2.03	2.01
$\|\|\mathbf{u} - \mathbf{u}^h\|\|_{\mathbf{L}^2(\Omega)}$	6.16e-3	4.86e-4	5.15e-5	6.17e-6	7.64e-7
order	-	3.66	3.24	3.06	3.01
$\|\|p - p^h\|\|_{L^2(\Omega)}$	3.21e-2	8.46e-4	5.20e-5	3.73e-6	2.89e-7
order	-	5.25	4.02	3.80	3.69
$\|\|\text{div } \mathbf{u}^h\|\|_{L^2(\Omega)}$	6.55e-16	5.27e-16	6.05e-16	1.32e-15	2.10e-15

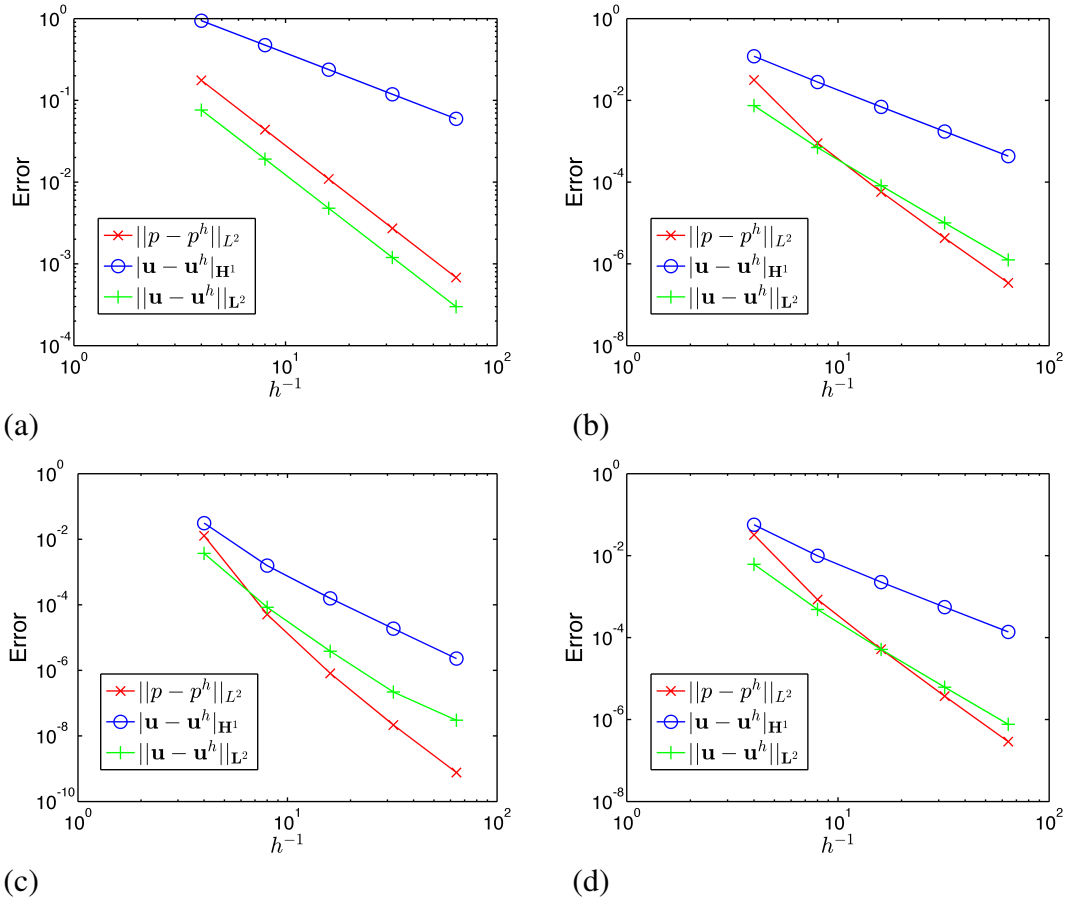

Figure 4.3.4: Stokes problem on a NURBS surface: Convergence plots for different choices of polynomial degrees: (a) $p = q = 1$, (b) $p = q = 2$, (c) $p = q = 3$ and (d) $p = 2$, $q = 3$.

and with prescribed velocities of unitary length moving in opposite tangential directions on the horizontal edges of the domain. This models a two-dimensional Stokes flow in a rectangular cavity driven by the motion of both the lid and bottom of the cavity. For this problem we use partitions of the form $\boldsymbol{\zeta} = \{i/2N\}_{i=0}^{2N}$ and $\boldsymbol{\gamma} = \{i/6N\}_{i=0}^{6N}$ for the ξ-direction and the η-direction of $\hat{\Omega}$ respectively, in order to get square h-by-h elements on Ω, where $h = 1/N$.

Figure 4.3.7 shows the streamlines of the discrete velocity and the discrete pressure obtained when $h = 1/16$, and gives a plot of the divergence. We also found that the L_2-norm of the divergence was $6.27e-16$. Again we conclude that the divergence is indeed zero apart from floating point truncation errors. In [17] analytical solutions for the streamlines are obtained, and comparing these with the streamlines in Figure 4.3.7 we observe that they are practically the same. In [6] the same benchmark problem is solved using a modified pressure space that gives optimal order convergence, comparing

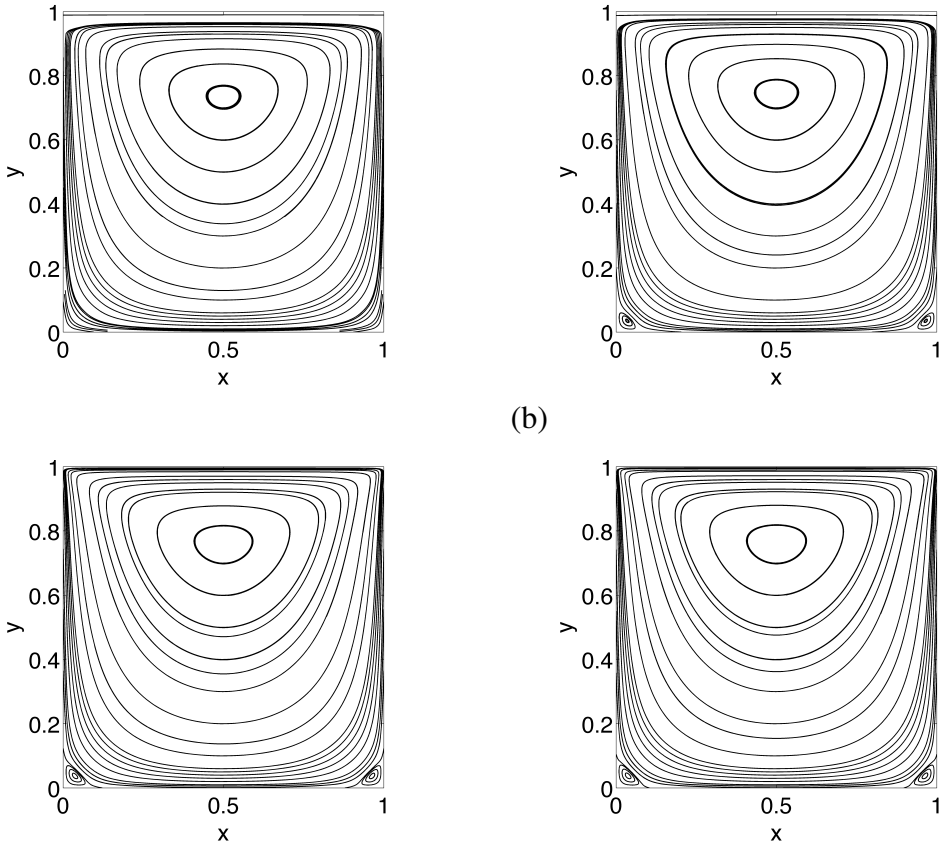

Figure 4.3.5: Lid-driven cavity flow: The streamlines for different levels of refinement: (a) $p = q = 1, h = 1/16$, (b) $p = q = 2, h = 1/16$, (c) $p = q = 1, h = 1/128$ and (d) $p = q = 2, h = 1/128$.

that solution with the solution in Figure 4.3.7 we see that the streamlines are very similar but the pressure is not the same.

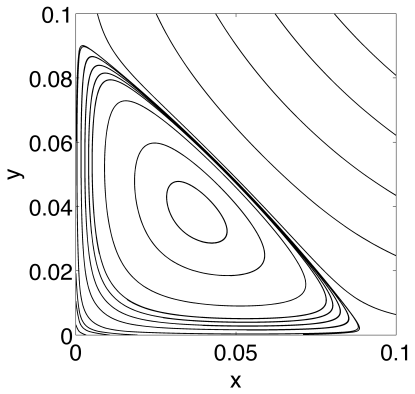

Figure 4.3.6: Lid-driven cavity flow: The first Moffatt eddy in the lower left corner when $p = q = 2$ and $h = 1/128$.

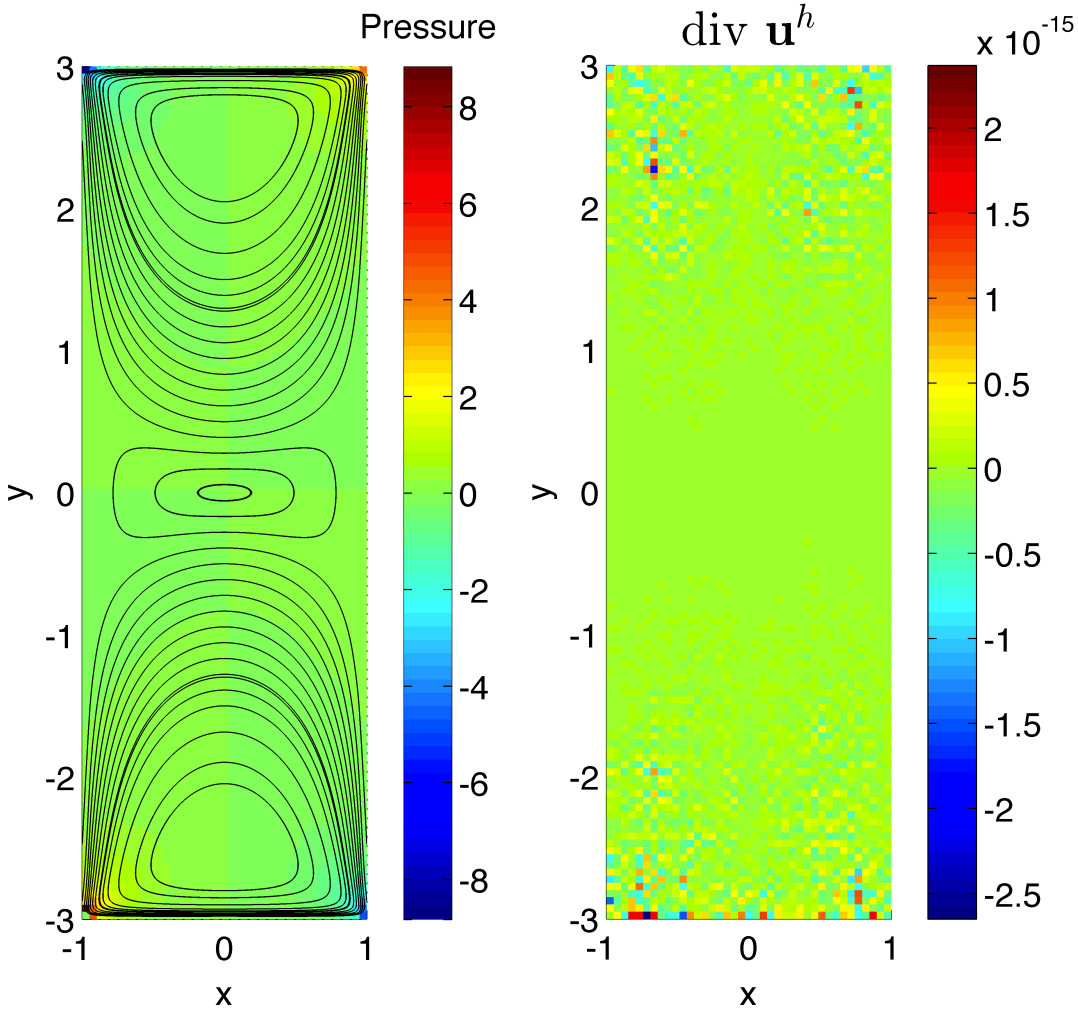

Figure 4.3.7: Two-sided lid-driven cavity flow: Streamlines and pressure of the numerical solution for $p = q = 3$ and $h = 1/16$ (left), and the divergence of the discrete velocity field (right).

Chapter 5

Conclusion

In this thesis we have seen how boundary value problems for partial differential equations can be solved numerically using isogeometric analysis, where we use B-spline basis functions, or more generally NURBS, both to express the geometry of the problem and as a basis for a finite element approximation. We have taken a close look at the construction and theory of splines and NURBS, including methods of basis refinement that make it possible to keep the exact geometry while the finite element spaces are refined. This is in contrast to classical finite element methods, where the geometry of the problem is often approximated using better approximations as the solution space is refined, thus changing the geometry during refinement.

Challenges regarding implementation have also been addressed, and a numerical solver of the two-dimensional Poisson problem has been implemented and tested using MATLAB. The Poisson problem serves as a model problem, and the numerical solver can be extended in many ways. Both more general boundary conditions and the multi-dimensional case can be handled without too much trouble, and the method and implementation is a great foundation for solving other partial differential equations.

Finally we have considered a method for numerically solving the Stokes problem for incompressible fluid flow, using divergence-conforming spline spaces. We have seen that this method gives a discrete velocity which is pointwise divergence-free, so the numerical solution satisfy mass conservation in an exact sense which is important in many applications. The method also displays very good stability and convergence properties.

Our results show that NURBS basis functions appear to work well in a finite element setting, and that they make it possible to use exact geometry throughout the analysis. We have also shown that isogeometric analysis provides great flexibility when it comes to refinement of the basis, in that we can control both polynomial degree and continuity in a very natural way. We have seen that this flexibility allows for the construction of compatible discrete velocity and pressure spaces for the Stokes equations, preserving

the structure of the continuous problem and providing a stable numerical method. The convergence properties of this method have been investigated numerically, with very good results for the numerical velocity solution but a reduced convergence rate for the pressure solution which is accounted for. The method has also been tested on benchmark problems, and the results confirm the stability of the method.

Bibliography

[1] Milton Abramowitz and Irene A Stegun. *Handbook of Mathematical Functions: With Formulars, Graphs, and Mathematical Tables*, volume 55. DoverPublications. com, 1964.

[2] Ivo Babuška. The finite element method with lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.

[3] Y. Bazilevs, L. Beirão da Veiga, J. A. Cottrell, T. J. R. Hughes, and G. Sangalli. Isogeometric analysis: Approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16(07):1031–1090, 2006.

[4] Y. Bazilevs and T.J.R. Hughes. Weak imposition of dirichlet boundary conditions in fluid mechanics. *Computers & Fluids*, 36(1):12 – 26, 2007.

[5] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 8(R2):129–151, 1974.

[6] A. Buffa, C. de Falco, and G. Sangalli. Isogeometric analysis: Stable elements for the 2d stokes equation. *International Journal for Numerical Methods in Fluids*, 65(11-12):1407–1422, 2011.

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

[8] H.B. Curry and I.J. Schoenberg. On pólya frequency functions iv: The fundamental spline functions and their limits. *Journal d'Analyse Mathématique*, 17(1):71–107, 1966.

[9] Claes Eskilsson and Spencer J. Sherwin. Spectral/hp discontinuous galerkin methods for modelling 2d boussinesq equations. *Journal of Computational Physics*, 212(2):566 – 589, 2006.

[10] John A. Evans, Yuri Bazilevs, Ivo Babuška, and Thomas J.R. Hughes. n-widths, sup–infs, and optimality ratios for the k-version of the isogeometric finite element method. *Computer Methods in Applied Mechanics and Engineering*, 198(21–26):1726 – 1741, 2009.

[11] John A. Evans and Thomas J. R. Hughes. Isogeometric divergence-conforming b-splines for the darcy–stokes–brinkman equations. *Mathematical Models and Methods in Applied Sciences*, 23(04):671–741, 2013.

[12] John Andrews Evans. *Divergence-free B-spline discretizations for viscous incompressible flows*. PhD thesis, The University of Texas at Austin, 2012.

[13] M. Fortin and R. Glowinski. Chapter 1 augmented lagrangian methods in quadratic programming. In Michel Fortin and Roland Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, volume 15 of *Studies in Mathematics and Its Applications*, pages 1 – 46. Elsevier, 1983.

[14] Michel Fortin and F Brezzi. *Mixed and hybrid finite element methods*. Springer, 1991.

[15] J.-F. Gerbeau, C. Le Bris, and M. Bercovier. Spurious velocities in the steady flow of an incompressible fluid subjected to external forces. *International Journal for Numerical Methods in Fluids*, 25(6):679–695, 1997.

[16] Finn-Idar Grøtta Giske. *An implementation of NURBS-based isogeometric analysis for the Poisson equation*. Specialization project, Norwegian Unversity of Science and Technology (NTNU), 2013.

[17] Fuat Gürcan. Streamline topologies in stokes flow within lid-driven cavities. *Theoretical and Computational Fluid Dynamics*, 17(1):19–30, 2003.

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

[19] Fridtjov Irgens. *Continuum mechanics*. Springer, 2008.

[20] Brian J. Kirby. *Micro- and nanoscale fluid mechanics: transport in microfluidic devices*. Cambridge University Press, 2010.

[21] Alexander Linke. Collision in a cross-shaped domain – a steady 2d navier–stokes example demonstrating the importance of mass conservation in {CFD}. *Computer Methods in Applied Mechanics and Engineering*, 198(41–44):3278 – 3286, 2009.

[22] Tom Lyche and Knut Mørken. *Spline Methods, Draft*. Department of Informatics, University of Oslo, 2011.

[23] HK Moffatt. Viscous and resistive eddies near a sharp corner. *Journal of Fluid Mechanics*, 18(01):1–18, 1964.

[24] Les Piegl and Wayne Tiller. Software-engineering approach to degree elevation of b-spline curves. *Computer-Aided Design*, 26(1):17 – 28, 1994.

[25] Hartmut Prautzsch. A short proof of the oslo algorithm. *Computer Aided Geometric Design*, 1(1):95 – 96, 1984.

[26] Alfio Quarteroni. *Numerical models for differential problems*, volume 2. Springer, 2009.

[27] P.A. Raviart and J.M. Thomas. A mixed finite element method for 2-nd order elliptic problems. In Ilio Galligani and Enrico Magenes, editors, *Mathematical Aspects of Finite Element Methods*, volume 606 of *Lecture Notes in Mathematics*, pages 292–315. Springer Berlin Heidelberg, 1977.