

**Weierstrass Institute for
Applied Analysis and Stochastics**

Isogeometric Analysis for Navier-Stokes equations

Elena Queirolo

Supervisors:
prof. Luca Dede'
prof. Volker John

Academic Year 2013-2014
École Polytechnique Fédérale de Lausanne
Weierstrass Institute for Applied Analysis and Stochastics

To Johanna

Abstract

In this project, we present some applications of Isogeometric Analysis in stationary fluid dynamics problems. Specially, we focus on the numerical solution of the Stokes and Navier-Stokes equations and the fulfillment of the inf-sup conditions, specifically by computing the Brezzi and Babuška inf-sup constants. The Isogeometric Taylor-Hood spaces are presented and the stability of the Stokes and Navier-Stokes equations has been numerically tested in a variety of settings, including smooth NURBS basis functions and the multiple patches case.

Contents

Index	1
1 Introduction	3
2 Isogeometric Analysis: geometrical representation by NURBS and approximation of PDEs	6
2.1 Introduction to B-splines	6
2.2 From B-splines to NURBS	11
2.3 Refinements	14
2.3.1 Knot insertion	14
2.3.2 Degree elevation	15
2.3.3 k -refinement	16
2.4 Domain and functions with IGA	16
2.5 Solving a partial differential problem with IGA	17
2.5.1 Dirichlet boundary conditions	18
2.6 NURBS patches	19
3 Stokes equations	22
3.1 Setting the problem	22
3.2 Taylor-Hood spaces	26
3.3 Calculating the inf-sup constant	27
3.4 Numerical tests	28
3.4.1 Poiseuille flow	29
3.4.2 Vortex flux	31
3.4.3 Different discretizations of the square	33
3.4.4 Higher order test functions	35
4 Navier-Stokes equations	39
4.1 Equations	40
4.2 Picard iterative method for Navier-Stokes equations	41
4.3 Numerical tests in the unit square	42
4.3.1 Exponential flux	44
4.4 The benchmark problem	46
4.4.1 Babuška inf-sup constant vs iteration number for different geometries	46

Conclusions and outlook	49
Bibliography	51

1

Introduction

Partial Differential Equations (PDE) are widely exploited to describe physical systems. In virtue of their flexibility, they can be applied to the modelization of many physical phenomena, from Mechanics to Chemistry. Most of these applications are not yet solvable analytically, and therefore require a numerical approximation. From the first introduction of computers in the field of numerical calculus, tremendous improvements took place, both in the hardware and in the software components. The concrete possibilities disclosed by such advances in Computer Science and Mathematics are immense and every year new fields are reshaped by them. In the last decades, the Engineering world has been significantly remodelled, thanks to the concrete employment of computers in everyday problems. This evolution has been made possible by two main factors: the availability of faster computational devices and the development of efficient computational methods. One of the main actors of this development can be considered to be one of the most widespread numerical method, the Finite Element method [19].

The strength of the Finite Element Method relies on its versatility, and up to now it has been used in all fields successfully. Researches had been carried out to apply this method to an increasing number of problems, making it one of the most employed numerical methods in applications. However the interface between applications and numerical methods is not always easy. Indeed, in several applications the domain in which the PDEs are defined is often constructed with Computer Aided Design tools. This domain needs then to be reconstructed in such a way to be usable by the Finite Element method. This interfacial step is called mesh generation and might, depending on the applications, take up to 80% of the computational time, as reported in [7]. This single datum underlines how time-consuming may be the construction of the mesh in the Finite Element Method when starting from a geometry defined in Computer Aided Design.

In order to overcome this drawback, in latest years research has been conducted on two sides: firstly, automatic methods to create a usable mesh are investigated [22], secondly,

methods that can be interfaced more easily with Computer Aided Design geometries were studied [13].

This master project examines a relatively new method called Isogeometric Analysis, firstly presented in [13]. As anticipated, Isogeometric Analysis is defined in such a way that it does not require an interface to the geometries created by Computer Aided Design tools, therefore overcoming the most cumbersome step in the application of the Finite Element method. But this is not the only innovation introduced by Isogeometric Analysis. As a matter of fact, the complete approach to the construction of the solution space has been remodelled. Indeed, in the Finite Element method the basis functions used for the discretization of the solution were defined on a reference element before the definition of the geometry. Conversely, the basis functions in Isogeometric Analysis are defined depending on the representation of the domain. Such approach is called Isogeometric concept and it is the point that gives the name to the whole method.

Up to now, the results presented in [4] and in [18] in the application of Isogeometric Analysis have been promising, therefore opening a new field of possible researches. Still, a lot of results and tests available for the Finite Element Method have not yet been carried out for Isogeometric Analysis. The aim of this project is therefore to contribute to reduce this gap. More precisely, in this master thesis Isogeometric Analysis is applied to Stokes and Navier-Stokes equations. These equations are applied in dynamics for modeling the behavior of fluids. Stokes equations just target the flows where the Reynolds number, the ratio between inertial forces and viscous forces, is small. This includes slow motion flows, or highly viscous fluids; practical examples can be honey in a short timescale or glaciers, in a much longer timescale. The second problem considered is the Navier-Stokes equations, that includes several types of fluids and flows, such as air in everyday life and water in the aqueduct pipes. These equations have a larger field of applications with respect to the Stokes equations, but the numerical computations for the solution are longer and more involved, requiring an iterative method to compute the solution. Therefore it is quite usual to couple the study of Navier-Stokes equations with their limit version, the Stokes equations.

One of the issues connected with the numerical solution of the Stokes and Navier-Stokes problems with Galerkin method, of which both Finite Element and Isogeometric Analysis are specific applications, is the so-called inf-sup condition. Its fulfillment is among the necessary conditions for the well-posedness of the problem and in order to satisfy it the discrete spaces for the velocity and pressure need to be suitably coupled. In the Stokes problem, two inf-sup constants can be calculated, called Brezzi and Babuška. Under some conditions on the continuous problem, they are proved to be equivalent and in this project they have been calculated explicitly to test the stability of the spaces used.

In this project, a set of stable spaces called isogeometric Taylor-Hood spaces are studied. These are spaces which satisfy the inf-sup conditions in Isogeometric Analysis. In particular, the discrete Brezzi and Babuška inf-sup constants are calculated for this coupled spaces in the tests considered, in order to show how the inf-sup constants appear to be bounded from below by a positive value independent of the enrichment of the discrete spaces considered. The results presented deals with NURBS basis functions of higher order

and of different smoothness.

The report is organized in the following chapters as follows:

Isogeometric Analysis An introduction to this numerical method is presented, from the definition of the B-spline and NURBS basis functions to the construction of the geometry and the application of the Isogeometric concept. In this chapter, the geometry of a benchmark problem for the Navier-Stokes equations is introduced and possible discretizations are presented.

Stokes equations The Stokes equations are introduced, with particular attention to the statement of the Brezzi and Babuška's inf-sup conditions. A coupled space for velocity and pressure that satisfy the Brezzi's inf-sup condition, called Taylor-Hood space, is introduced and numerical tests are performed on the unit square, using solutions of different order and continuity and varying the parametrization of the domain.

Navier-Stokes equations After the introduction of the Navier-Stokes equations and the Picard iterative method, needed to solve the non-linear problems, this chapter focuses on the numerical solutions of benchmark problems. The behavior of the inf-sup constant is strictly related to the discretization of the geometry used.

Conclusion In this last part, the conclusions of our tests are presented, as well as the future possible development of the research.

2

Isogeometric Analysis: geometrical representation by NURBS and approximation of PDEs

In recent years, the importance of numerical methods to solve PDEs has increased steadily, also thanks to the development of efficient toolboxes for their solution. The most wide spread method is the Finite Element Method (FEM), as presented in [20]. The large use of this method underlined how time-consuming is the construction of a proper geometry for the application of the method itself. Indeed, often this step requires more time than all the rest of the computations. For this reason, a lot of effort has been dedicated to the implementation of efficient mesh generators, such as TetGen [22]. An offspring of such research has been the method that will be presented and applied in this project: Isogeometric Analysis (IGA), that was first introduced in [13].

The difference between Finite Element Method and Isogeometric Analysis for what deals with the discretization of the geometry is huge, since FEM uses triangles or quadrangles to construct a geometry that approximates the physical domain, while IGA takes advantage of the flexibility of Non-Uniform Rational B-Splines (NURBS) to construct a geometry that is exact already at its coarsest level. Even more important than the use of different basis functions is the introduction of the isogeometric concept, that is the idea to construct the basis functions depending on the NURBS representation of the domain. Both these differences, in the construction of the domain and the set of basis functions, will be presented in this chapter. For a deeper understanding of the subject, we refer the interested reader to [7].

2.1 Introduction to B-splines

In this section, B-splines and some of their applications and properties are presented. B-splines will prove to be very useful in the following, both directly and as a first step for the

construction of NURBS. B-splines are piecewise polynomials defined recursively by means of a given knot vector. The concept of knot vector and the basis functions for the B-spline space are defined as follows.

Definition 1. Let m be a positive integer and $\xi_1, \xi_2, \dots, \xi_m$ be an increasing series of real numbers. By convention, $\xi_1 = 0$ and $\xi_m = 1$. Then, a *knot vector* is defined as

$$\Xi = \{\xi_1, \xi_2, \dots, \xi_m\}$$

and it is said to be *open* and of degree (or order) p if

$$\xi_1, \xi_2, \dots, \xi_{p+1} = 0 \quad \text{and} \quad \xi_{m-p-1}, \xi_{m-p}, \dots, \xi_m = 1.$$

For future use, we introduce here some further notations for the knot vectors.

Definition 2. Let us consider a knot vector Ξ , of degree 0 and without repetitions. From such knot vector, it is possible to construct a knot vector of degree p and with each node repeated $r \leq p$ times, by repeating $p + 1$ times the first and last knots and all the others r times. The knot vector so constructed will be denoted by $\Xi(p, r)$.

Definition 3. Given an open knot vector Ξ of degree p and cardinality $m \geq 2(p + 1)$, there exist $m - 1$ *B-spline basis functions* of degree 0, defined by

$$B_{i,0}(\xi) = \begin{cases} 1 & \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \dots, m - 1. \quad (2.1)$$

Given $p \geq q > 0$, the B-spline basis functions of degree q are defined recursively by

$$B_{i,q}(\xi) = \frac{\xi - \xi_i}{\xi_{i+q} - \xi_i} B_{i,q-1} + \frac{\xi_{i+q+1} - \xi}{\xi_{i+q+1} - \xi_{i+1}} B_{i+1,q-1} \quad i = 0, \dots, n_q = m - q - 1, q = 1, \dots, p. \quad (2.2)$$

Henceforward, n will denote the number of basis functions of degree p for a given knot vector, that is $n = n_p = m - p - 1$.

Remark 4. In this definition, the convention $\frac{0}{0} = 0$ is used.

In the following, when not otherwise stated, the B-splines will be assumed to be of degree p .

We list here some properties of the B-splines basis functions. The proofs are not presented; we refer the interested reader to [21].

Property 5.

- If the internal knots in Ξ are strictly increasing, then for every degree $0 \leq q \leq p$ the functions $B_{i,q}$ are linearly independent.
- The function $B_{i,p}$ has a compact support, such that $\text{supp}(B_{i,p}) \subseteq [\xi_i, \xi_{i+p+1}]$.

- For a fixed $0 \leq q \leq p$, the set $\{B_{i,q}\}_{i=1}^{n_q}$ forms a partition of unity, i.e., for any $\xi \in [0, 1]$

$$\sum_{i=1}^{n_q} B_{i,q}(\xi) = 1.$$

- The B-spline basis functions are positive, i.e., for any $\xi \in [0, 1]$

$$B_{i,q}(\xi) \geq 0 \quad \text{for } i = 1, \dots, n_q \text{ and for } 0 \leq q \leq p.$$

- It holds $B_{i,p}(\xi) \in C^\infty$ for $\xi \in (\xi_j, \xi_{j+1})$, $j = 1, \dots, m-1$, and $B_{i,p}(\xi_j) \in C^{p-r_j}$ for ξ_j of multiplicity r_j , $j = p+2, \dots, m-p-2$.
- If $B_{i,p}$ is k times differentiable in ξ , then it holds

$$D^k B_{i,p}(\xi) = p \left(\frac{D^{k-1} B_{i,p}(\xi)}{\xi_{i+p} - \xi_i} - \frac{D^{k-1} B_{i+1,p-1}(\xi)}{\xi_{i+p+1} - \xi_{i+1}} \right).$$

The functions $B_{i,p}$ just introduced will be referred to as basis functions since they constitute the basis of a space, called B-spline space, of major importance for the IGA setting. This space is described in the following definition.

Definition 6. Given an open knot vector Ξ of degree p and cardinality m , the B-spline space \mathcal{S} is defined to be

$$\mathcal{S}(\Xi, p) = \text{span}\{B_{i,p}(\xi), i = 1, \dots, n\}.$$

A theorem from Curry and Schoenberg [8] demonstrates that a definition of $\mathcal{S}(\Xi, p)$ equivalent to Definition 6 is

$$\mathcal{S}(\Xi, p) = \left\{ f : f|_{\xi_i, \xi_{i+1}} \in \mathbb{P}^p(\xi_i, \xi_{i+1}) \text{ } i = 1, \dots, m-1 \text{ and } f(\xi_j) \in C^{p-r_j} \text{ for } \xi_j \text{ of multiplicity } r_j, j = p+2, \dots, m-p-2 \right\},$$

where we used the notation $\mathbb{P}^p(x, y)$ to indicate polynomials of degree at most p in the interval (x, y) .

A first application of the B-splines consists in constructing B-spline curves. The idea is to multiply each B-spline function by a corresponding control point in \mathbb{R}^d and summing all the function thus obtained. This process leads to a B-spline curve as formalized in the following definition.

Definition 7. Let Ξ be an open knot vector of degree p and cardinality m , let $n = m-p-1$ and $\{B_{i,p}\}_{i=1}^n$ the B-spline basis of degree p . Let $\{\mathbf{P}_i : \mathbf{P}_i \in \mathbb{R}^d\}_{i=1}^n$ be a set of control points, with $d \in \mathbb{N}$. Then, a B-spline curve in \mathbb{R}^d of degree p on Ξ is defined by the formula

$$C(\xi) = \sum_{i=1}^n \mathbf{P}_i B_{i,p}(\xi).$$

In the following definition, a d -dimensional B-spline object is introduced as a scalar valued function constructed by the tensor product of d B-spline basis functions.

Definition 8. Let D be a positive integer and $\Xi^{(1)}, \dots, \Xi^{(D)}$ be open knot vectors of degrees $p^{(1)}, \dots, p^{(D)}$, respectively. The tensor product of D one-dimensional B-spline basis functions of degree p_j , for $j = 1, \dots, D$, defines a D -dimensional B-spline basis function, by the formula

$$\begin{aligned} B_{i_1, i_2, \dots, i_D; p_1, p_2, \dots, p_D} : [0, 1]^D &\longrightarrow \mathbb{R} \\ (\xi^{(1)}, \xi^{(2)}, \dots, \xi^{(D)}) &\longmapsto \bigotimes_{j=1}^d B_{i_j, p_j}(\xi^{(j)}). \end{aligned}$$

The span of the functions defines the D -dimensional B-spline space

$$\mathcal{S}(\Xi^{(1)}, \dots, \Xi^{(D)}; p^{(1)}, \dots, p^{(D)}) = \text{span} \{ B_{i_1, \dots, i_D; p_1, p_2, \dots, p_D} \}_{i_1=1, \dots, i_D=1}^{n^{(1)}, \dots, n^{(D)}},$$

where $n^{(j)} = m^{(j)} - p^{(j)} - 1$ for $j = 1, \dots, D$ and $n = \prod_{j=1}^D n^{(j)}$. With a process similar to the one applied from B-splines basis functions to B-spline curves, a *B-spline D -surface* can be introduced as

$$S(\xi^{(1)}, \dots, \xi^{(D)}) = \sum_{i_1=1, \dots, i_D=1}^{n^{(1)}, \dots, n^{(D)}} B_{i_1, i_2, \dots, i_D; p_1, p_2, \dots, p_D}(\xi^{(1)}, \dots, \xi^{(D)}) \mathbf{P}_{i_1, i_2, \dots, i_D},$$

where $\mathbf{P}_{i_1, i_2, \dots, i_D} \in \mathbb{R}^d$, $d \geq D + 1$, are control points.

All the properties of one-dimensional B-splines can be easily extended to this d -dimensional setting, here just two new properties are stressed without demonstration. For a deeper presentation of the d -dimensional B-splines and B-spline D -surfaces, we refer the interested reader to [7].

Property 9.

- Given an affine transformation $\Phi : \mathbb{R}^d \rightarrow \mathbb{R}^d$, the B-spline D -surface satisfies the affine invariance property, i.e.,

$$\begin{aligned} \Phi \left(\sum_{i_1=1, \dots, i_D=1}^{n^{(1)}, \dots, n^{(D)}} B_{i_1, i_2, \dots, i_D; p_1, p_2, \dots, p_D}(\xi^{(1)}, \dots, \xi^{(D)}) \mathbf{P}_{i_1, i_2, \dots, i_D} \right) &= \\ = \sum_{i_1=1, \dots, i_D=1}^{n^{(1)}, \dots, n^{(D)}} B_{i_1, i_2, \dots, i_D; p_1, p_2, \dots, p_D}(\xi^{(1)}, \dots, \xi^{(D)}) \Phi(\mathbf{P}_{i_1, i_2, \dots, i_D}). \end{aligned}$$

- A B-spline D -surface satisfies the strong convex hull property, i.e., for any $(\xi^{(1)}, \dots, \xi^{(D)}) \in \bigotimes_{k=1}^D [\xi_{i_k}^{(k)}, \xi_{i_k+1}^{(k)}]$, where $\xi_{i_k}^{(k)} \in \Xi_k$, then $S(\xi^{(1)}, \dots, \xi^{(D)})$ lies in the convex hull of $\{\mathbf{P}_{j_1, j_2, \dots, j_D}, j_k = i_k - p^{(k)}, \dots, i_k, k = 1, \dots, D\}$.

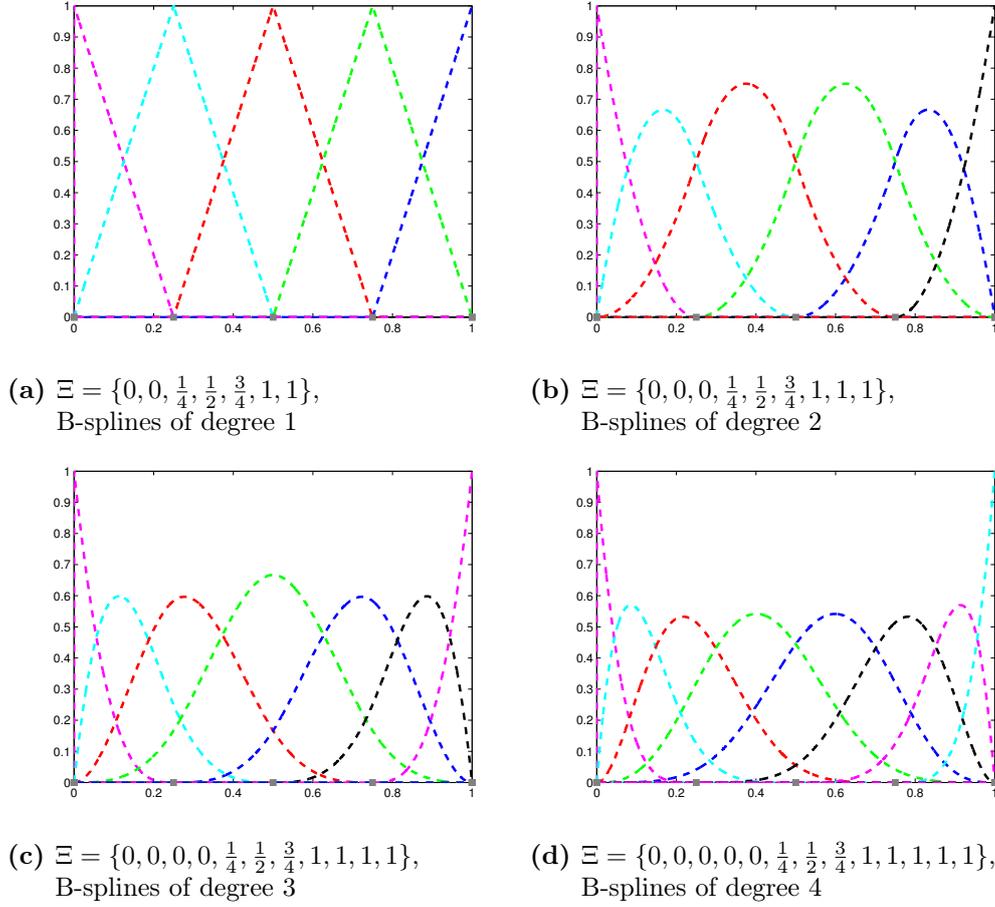


Figure 2.1: B-splines basis functions and corresponding knot vectors of degrees 1 to 4

Example 10. Let us present here a few plots, see Figure 2.1, to clarify the behavior of the B-spline basis functions depending on the knot vector used. In the four plots, the knot vectors considered are degree elevation of the knot vector $\Xi = \{0, 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, 1\}$, that is, the knot vectors considered are $\Xi(p, 1)$ with $p = 1, \dots, 4$.

Example 11. A slightly different case is when the multiplicity of an inner node is higher than one, therefore lowering the degree of continuity of the B-spline functions in that knot. As an example, in Figure 2.2 are presented the B-spline basis functions defined by the knot vector $\Xi = \{0, 0, 0, \frac{1}{4}, \frac{1}{4}, \frac{1}{2}, 1, 1, 1\}$.

Example 12. Let us here also present a graphical example of a curve in \mathbb{R}^2 . Let us take the knot vector of degree 3 of Example 10 and choose some points in the plane, such as

$$\begin{aligned} P_1 &= [1, 4]^T & P_2 &= [2, 3]^T & P_3 &= [3, 7]^T & P_4 &= [4, 1]^T \\ P_5 &= [7, 8]^T & P_6 &= [6, 4]^T & P_7 &= [5, 1]^T. \end{aligned}$$

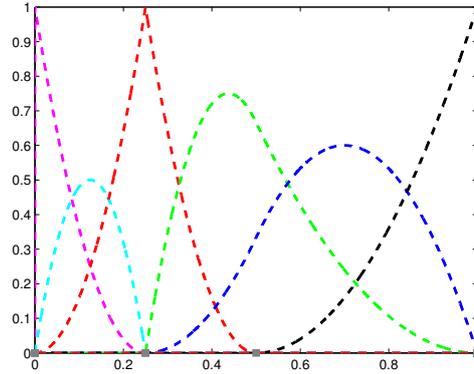


Figure 2.2: B-splines defined by Example 11

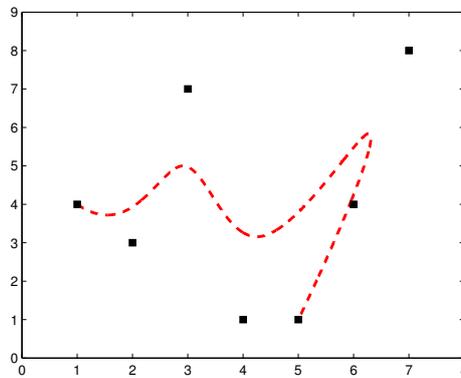


Figure 2.3: B-Spline curve as in Example 12, in black are the control points used

This combination gives the curve presented in Figure 2.3.

2.2 From B-splines to NURBS

In this section, we introduce the Non-Uniform Rational B-Splines, an extension of B-splines that will be used to construct our numerical domain during the modelization. The idea is to generalize the B-splines curves and surfaces by weighting each component differently, therefore giving more or less importance to each B-spline basis function in the space. This new elements have the important capacity of representing exactly conic sections, such as ellipses and circles. This is clearly a huge advantage over the polygonal domains representable with FEM.

In the following definition, the steps of the construction of the geometry are explained.

Definition 13. Given an open knot vector Ξ of degree p , a set of weights $\{\omega_i : \omega_i \in \mathbb{R}\}_{i=1}^n$ and a set of control points $\{\mathbf{P}_i : \mathbf{P}_i \in \mathbb{R}^d\}_{i=1}^n$, with $d \geq 1$, then the *Non-Uniform Rational*

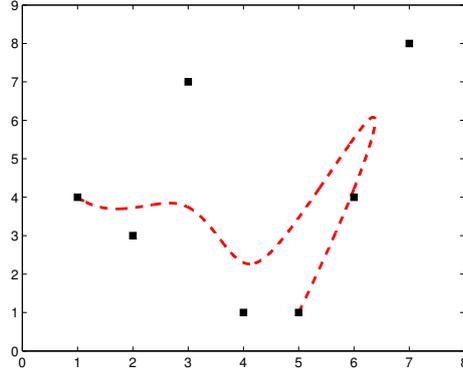


Figure 2.4: NURBS curve as in Example 14, in black are the control points used, that coincide with the control points of Figure 2.3

B-Spline curve (NURBS curve) of degree p in \mathbb{R}^d is defined as

$$C(\xi) = \frac{\sum_{i=1}^n B_{i,p}(\xi) \omega_i \mathbf{P}_i}{\sum_{i=1}^n B_{i,p}(\xi) \omega_i} \quad \forall \xi \in [0, 1], \quad (2.3)$$

or equivalently

$$C(\xi) = \sum_{i=1}^n R_{i,p}(\xi) \mathbf{P}_i, \quad \text{where} \quad R_{i,p}(\xi) = \frac{B_{i,p}(\xi) \omega_i}{\sum_{k=1}^n B_{k,p}(\xi) \omega_k}. \quad (2.4)$$

Example 14. In this example, let us present a NURBS curve using the same data of Example 12. Taking the same knot vector of degree 3, and the same control points, we will just add a set of 7 weights ω , such as

$$\begin{aligned} \omega_1 &= 1, & \omega_2 &= \frac{2}{5}, & \omega_3 &= \frac{1}{5}, & \omega_4 &= \frac{7}{10}, \\ \omega_5 &= \frac{3}{5}, & \omega_6 &= \frac{3}{10}, & \omega_7 &= \frac{1}{10}. \end{aligned}$$

The resulting plot can be seen in Figure 2.3.

Example 15. One of the main strengths of the NURBS is the possibility of drawing perfect arches, therefore the construction of an arch is presented in this example. By setting the knot vector to $\Xi = \{0, 0, 0, 1, 1, 1\}$ and selecting the points

$$\mathbf{P}_1 = [0, 0]^T, \quad \mathbf{P}_2 = [0, 1]^T, \quad \mathbf{P}_3 = [1, 1]^T,$$

and the weights

$$\omega_1 = 1, \quad \omega_2 = \cos\left(\frac{1}{2} \cdot \frac{\pi}{2}\right), \quad \omega_3 = 1,$$

we recover an arch of width $\frac{\pi}{2}$, such as presented in Figure 2.5.

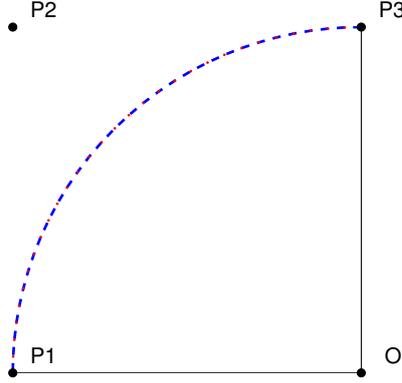


Figure 2.5: In blue, an arch defined by its analytic expression, in red the same arch plotted with NURBS, as defined in Example 15. As we can see, they are indistinguishable. The center of the circle was called O .

Bringing the definition to more than one dimension does not require new concepts, nevertheless, for the sake of completeness and because they will be largely used in the following, here is presented the definition of the NURBS surfaces.

Definition 16. Given two open knot vectors $\Xi^{(1)}$ and $\Xi^{(2)}$ of degree $p^{(1)}$ and $p^{(2)}$ respectively, a set of control points $\{\mathbf{P}_{i,j} : \mathbf{P}_{i,j} \in \mathbb{R}^d\}_{i=1, j=1}^{n^{(1)}, n^{(2)}}$, $d \geq 2$, and a set of weights $\{\omega_{i,j} : \omega_{i,j} \in \mathbb{R}\}_{i=0, j=0}^{n^{(1)}, n^{(2)}}$, then a *NURBS surface* defined by those elements is

$$S(\xi^{(1)}, \xi^{(2)}) = \frac{\sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} B_{i,j;p^{(1)}, p^{(2)}}(\xi^{(1)}, \xi^{(2)}) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} B_{i,j;p^{(1)}, p^{(2)}}(\xi^{(1)}, \xi^{(2)}) \omega_{i,j}}, \quad (2.5)$$

for all $(\xi^{(1)}, \xi^{(2)}) \in [0, 1]^2$, where $B_{i,j;p^{(1)}, p^{(2)}}$ are as in Definition 8.

Example 17. Let us present here a geometry constructed by a NURBS surface in \mathbb{R}^3 . In Figure 2.6, three quarters of a cylinder have been plotted, by setting

$$\Xi^{(1)} = \left\{ 0, 0, 0, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, 1, 1, 1 \right\}, \quad \Xi^{(2)} = \{0, 0, 1, 1\},$$

$$\mathbf{P}_{1,1} = [0, -1, 0]^T, \mathbf{P}_{2,1} = \cos\left(\frac{\pi}{4}\right) [1, -1, 0]^T, \mathbf{P}_{3,1} = [1, 0, 0]^T, \mathbf{P}_{4,1} = \cos\left(\frac{\pi}{4}\right) [1, 1, 0]^T,$$

$$\mathbf{P}_{5,1} = [0, 1, 0]^T, \mathbf{P}_{6,1} = \cos\left(\frac{\pi}{4}\right) [-1, 1, 0]^T, \mathbf{P}_{7,1} = [-1, 0, 0]^T,$$

$$\mathbf{P}_{1,2} = [0, -1, 3]^T, \mathbf{P}_{2,2} = \cos\left(\frac{\pi}{4}\right) [1, -1, 3]^T, \mathbf{P}_{3,2} = [1, 0, 3]^T, \mathbf{P}_{4,2} = \cos\left(\frac{\pi}{4}\right) [1, 1, 3]^T,$$

$$\mathbf{P}_{5,2} = [0, 1, 3]^T, \mathbf{P}_{6,2} = \cos\left(\frac{\pi}{4}\right) [-1, 1, 3]^T, \mathbf{P}_{7,2} = [-1, 0, 3]^T,$$

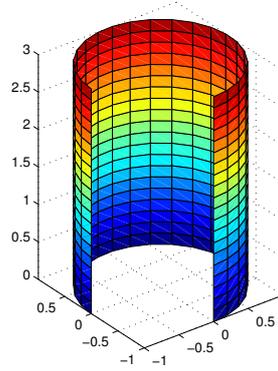


Figure 2.6: Cylinder constructed from the elements described in Example 17.

$$\begin{aligned} \omega_{i,1} = \omega_{i,2} &= 1, & \text{for } i = 1, 3, 5, 7, \\ \omega_{i,1} = \omega_{i,2} &= \cos\left(\frac{\pi}{4}\right), & \text{for } i = 2, 4, 6. \end{aligned}$$

2.3 Refinements

For a similarity to FEM, often the methods presented in this section are called refinements, even if just one of them is a real refinement of the grid, the others should rather be called enrichments of the B-spline spaces.

Once a B-spline space is defined, there are different ways to enrich it. One is by adding a knot to the knot vector, *knot insertion* or *h-refinement*, another is by augmenting the degree of the knot vector itself, *degree elevation* or *p-refinement*. A third way, consisting in combining the two previous, is called *k-refinement* and it is the most widely used. Each of this methods will be presented in a separate sub-section.

2.3.1 Knot insertion

The process of the knot insertion, also called *h-refinement*, starts by considering a knot vector Ξ , of degree p and associated basis functions $B_{i,j}$, and a new knot $\tilde{\xi}$. The idea is to add the knot $\tilde{\xi}$ to the knot vector, thus creating a new knot vector

$$\tilde{\Xi} = \Xi \cup \{\tilde{\xi}\}, \quad (2.6)$$

and defining the new set of basis functions $\tilde{B}_{i,p}$ on $\tilde{\Xi}$. Let us assume the knot $\tilde{\xi}$ will be the k -th entry of the knot vector $\tilde{\Xi}$.

A curve C defined in the initial functional space by the control points \mathbf{P}_i can be defined also in the enriched space, by constructing a new set of control points \mathbf{Q}_i that fulfills

$$C(\xi) = \sum_{i=1}^n B_{i,p}(\xi) \mathbf{P}_i = \sum_{i=1}^{n+1} \tilde{B}_{i,p}(\xi) \mathbf{Q}_i \quad \forall \xi \in [0, 1].$$

To satisfy this equation, the new control points will have to be defined as

$$\mathbf{Q}_i = \alpha_i \mathbf{P}_i + (1 - \alpha_i) \mathbf{P}_{i-1} \quad (2.7)$$

where

$$\alpha_i = \begin{cases} 1, & i \leq k - p, \\ \frac{\tilde{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leq i \leq k, \\ 0, & k + 1 \leq i. \end{cases}$$

This interpolation function permits to define the same curves (and similarly surfaces and hyper-surfaces) with the new knot vector. A relation between the basis functions defined by the two knot vectors can also be constructed.

Lemma 18. *Having Ξ and $\tilde{\Xi}$ defined as in Equation (2.6), for $k - p \leq i \leq k$, the relationship*

$$B_{i,p}(\xi) = \frac{\tilde{\xi} - \tilde{\xi}_i}{\tilde{\xi}_{i+p+1} - \tilde{\xi}_i} \tilde{B}_{i,p}(\xi) + \frac{\tilde{\xi}_{i+p+2} - \tilde{\xi}}{\tilde{\xi}_{i+p+2} - \tilde{\xi}_{i+1}} \tilde{B}_{i+1,p}(\xi)$$

holds, where the knots marked with a tilde belongs to the new knot vector $\tilde{\Xi}$.

With the help of this lemma, all the functions based on Ξ can be directly formulated depending on the new knot vector $\tilde{\Xi}$.

2.3.2 Degree elevation

The idea of the degree elevation, also called p -refinement, is to consider a knot vector Ξ of degree p and cardinality m . In order to bring it to be of degree $p + 1$ it is enough to repeat all the knots of the vector once more. This means that, starting from

$$\Xi = \{\underbrace{0, \dots, 0}_{p+1}, \dots, \underbrace{\xi_i, \dots, \xi_i}_{r_i}, \dots, \underbrace{1, \dots, 1}_{p+1}\} \quad (2.8)$$

it becomes

$$\tilde{\Xi} = \{\underbrace{0, \dots, 0}_{p+2}, \dots, \underbrace{\xi_i, \dots, \xi_i}_{r_i+1}, \dots, \underbrace{1, \dots, 1}_{p+2}\}. \quad (2.9)$$

The reason to add once also all the internal nodes is to preserve the regularity of the original basis functions on the internal nodes. As a matter of fact, the functions constructed with the knot vector Ξ are of order of continuity $p - r_i$ in ξ_i , and it is required from the refinement

to have the same order of continuity in the nodes. Therefore it is required to add once the knot ξ_i in $\tilde{\Xi}$.

Considering a curve C defined on Ξ by a set of control points \mathbf{P}_i , an important question is, as previously, how to construct the control points \mathbf{Q}_i , such that it is possible to define a curve \tilde{C} on $\tilde{\Xi}$ equal to C , i.e.,

$$C(\xi) = \sum_{i=1}^n B_{i,p}(\xi) \mathbf{P}_i = \sum_{i=1}^{\tilde{n}} \tilde{B}_{i,p+1}(\xi) \mathbf{Q}_i = \tilde{C}(\xi).$$

We refer the interested reader to [16] for the details about the algorithm to construct the new control points in this case.

2.3.3 k -refinement

Unlikely what happens in the FEM refinements, the two methods presented above do not commute, leading therefore to different spaces depending on the sequence of application. The k -refinement consists in applying first the degree elevation and then the knot insertion, in order to have a higher order of continuity in the newly added knot. This method has a significant place in literature due to its outstanding importance compared to the other two refinements, that are rarely used on their own.

2.4 Domain and functions with IGA

Often in applications, the domains to be discretized have already been plotted in Computer Aided Design (CAD), a technique based on NURBS. For this reason, a NURBS-based numerical method is of the outmost interest. This is the case for IGA, that does not require any further transformation of the mesh, unlike FEM. As a matter of fact, once a domain Ω is defined through a mapping S , the basis functions for the numerical analysis are already set, as explained in the following.

Keeping the notation used in Definition 16, let us consider a NURBS surface in \mathbb{R}^3 defining the numerical domain Ω , that is, a function

$$\begin{aligned} S : [0, 1]^2 &\longrightarrow \Omega \subset \mathbb{R}^3 \\ (\xi^{(1)}, \xi^{(2)}) &\longmapsto S(\xi^{(1)}, \xi^{(2)}) = \frac{\sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} B_{i,j}(\xi^{(1)}, \xi^{(2)}) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} B_{i,j}(\xi^{(1)}, \xi^{(2)}) \omega_{i,j}}, \end{aligned} \quad (2.10)$$

where the control points $\mathbf{P}_{i,j}$ are in \mathbb{R}^3 . In order to define such mapping, the two knot vectors $\Xi^{(1)}$ and $\Xi^{(2)}$ are given, as well as the weights $\omega_{i,j}$ and the control points $\mathbf{P}_{i,j}$. The domain constructed by this transformation of the unit square can be written as

$$\Omega = \left\{ (x, y, z) : (x, y, z) = S(\xi^{(1)}, \xi^{(2)}), (\xi^{(1)}, \xi^{(2)}) \in (0, 1)^2 \right\}.$$

The isogeometric concept is that the discrete function spaces used to discretize the functions on the domain Ω depend on the discretization of the domain itself. Once defined

the mapping S , the same B-splines used to construct Ω are also used for the discretization of the functions on Ω . This means that the functions considered are all the ones that can be written in the form

$$u : \Omega \in \mathbb{R}^3 \longrightarrow \mathbb{R}$$

$$(x, y, z) \longmapsto \hat{u}(S^{-1}(x, y, z)) = \sum_{i=1}^{n^{(1)}} \sum_{j=1}^{n^{(2)}} B_{i,j}(S^{-1}(x, y, z)) c_{i,j}, \quad (2.11)$$

with $c_{i,j}$ control variables and $B_{i,j}$ defined by the geometry used in Equation (2.10).

Once the geometry given through the transformation S , the function space for the discretization of our functions is set. In fact, by renumbering the indexes and setting

$$\phi_l = B_{i,j}, \quad \text{with } i = \left\lfloor \frac{l}{n} \right\rfloor \text{ and } j = l - i n, \quad (2.12)$$

Equation (2.11) can be rewritten

$$u(x, y, z) = \sum_{i=1}^N \phi_i(\xi^{(1)}, \xi^{(2)}) c_i \quad (2.13)$$

where $(\xi^{(1)}, \xi^{(2)}) = S^{-1}(x, y, z)$ and $N = n^{(1)}n^{(2)}$. This last formulation strongly resembles the notation usually used for FEM, therefore permitting us most of the time to carry on our analysis without changing the already familiar notation.

2.5 Solving a partial differential problem with IGA

Considering the problem of discretizing a PDE, the preceding sections provide at the same time both a parametrization of the physical domain and a discretization of the functional spaces. In this section will be presented how this concepts can be used in the solution of a differential model problem.

Let us consider here the strong form of the differential (second order) model problem, that will be denoted as

$$\text{find } u \in V : \quad \tilde{a}(u) = \tilde{F}(x), \quad \forall x \in \Omega, \quad (2.14)$$

where $\tilde{a}(\cdot) : V \rightarrow \mathbb{R}$ and $\tilde{F}(\cdot) : \Omega \rightarrow \mathbb{R}$ correspond to the weak formulation

$$\text{find } u \in V : \quad a(u, v) = F(v), \quad \forall v \in V, \quad (2.15)$$

where V is a functional space on the domain Ω , $a(\cdot, \cdot) : V \times V \rightarrow \mathbb{R}$ is a bilinear continuous coercive functional and $F(\cdot) : V \rightarrow \mathbb{R}$ is a linear continuous functional. In this first presentation V will be set equal to $H^1(\Omega)$, the boundary conditions are not yet considered, and it will be assumed that (2.15) is well posed.

Let us construct a NURBS mapping S from the unit square to the domain Ω , such as in (2.10). Once the mapping is constructed, also the B-spline space $\mathcal{S}(\Xi, p)$ can be defined

on the numerical domain, as presented in the previous Section 2.4. Using such approach, it is possible to build a space $V_h = V \cap \mathcal{S}$, which is an approximation of the function space V .

Given \mathcal{S} , let us denote, in accordance with (2.12), its basis functions as $\phi_i(\xi^{(1)}, \xi^{(2)})$, with $i = 1, \dots, N$ as previously. In this way, the variational problem applied to the discrete space V_h reads

$$\text{find } u \in V_h : a(u, v) = F(v), \quad \forall v \in V_h, \quad (2.16)$$

or equivalently, by writing u and v explicitly and applying the linearity of $a(\cdot, \cdot)$ and $F(\cdot)$ on v ,

$$\text{find } c_i \in \mathbb{R} : a\left(\sum_{i=1}^N c_i \phi_i, \phi_j\right) = F(\phi_j), \quad \forall j = 1, \dots, N.$$

By applying again the linearity of a , this time on the first element, the previous equation can be transformed into

$$\text{find } c_i \in \mathbb{R} : \sum_{i=1}^N c_i a(\phi_i, \phi_j) = F(\phi_j), \quad \forall j = 1, \dots, N,$$

and therefore to the linear system

$$A u = F, \quad (2.17)$$

where

$$\begin{aligned} A_{i,j} &= a(\phi_j, \phi_i), \\ F_j &= F(\phi_j), \\ u_i &= c_i. \end{aligned}$$

In this way, a second order differential problem was transformed into an algebraic linear system. By solving the linear system (2.17), an approximate solution to (2.15) is found in the space $V_h = V \cap \mathcal{S}$.

2.5.1 Dirichlet boundary conditions

An issue we have not addressed in the previous section is how to deal with boundary conditions, in particular Dirichlet boundary conditions.

As for FEM, the Dirichlet boundary conditions can be imposed directly on the function space V and consequentially on V_h . Let us here start from the general form of the differential problem presented in (2.14) and add two equations, one for Dirichlet and one for Neumann boundary conditions. The starting point of our discussion would therefore be

$$\begin{aligned} \text{find } u \in V : \tilde{a}(u) &= \tilde{F}, \quad \forall x \in \Omega, \\ u|_{\Gamma_D} &= u_D, \\ \frac{\partial u}{\partial n}\Big|_{\Gamma_N} &= 0, \end{aligned} \quad (2.18)$$

where Γ_D and Γ_N are a partition of the boundary of Ω , i.e., $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$. The first condition on the boundary is called Dirichlet boundary condition, while the second is a homogeneous Neumann boundary condition. It will be required in the following that $\Gamma_D \neq \emptyset$.

This problem corresponds to the weak form

$$\text{find } u \in V_D : \quad a(u, v) = F(v), \quad \forall v \in V_0, \quad (2.19)$$

with

$$V_D = \{v \in H^1(\Omega) : v|_{\Gamma_D} = u_D\} \quad \text{and} \quad V_0 = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$$

From the numerical point of view, in order to impose the nonhomogeneous Dirichlet boundary condition, the solution is forced to have the prescribed values at the boundary by fixing the degrees of freedom that characterize the solution on Γ_D . This is achieved by defining a *lifting function* $Ru \in V$ that fulfills the Dirichlet boundary conditions, and by requiring from the solution to be in the form $u = Ru + \tilde{u}$, where \tilde{u} is a function which is zero at the Dirichlet boundary, that means it is just defined in the internal degrees of freedom. Naturally, by imposing such form to our solution, the problem itself will be changed into

$$\begin{aligned} \text{find } \tilde{u} \in V_0 : \quad a(Ru + \tilde{u}, v) &= F(v), & \forall v \in V_0, \\ \text{find } \tilde{u} \in V_0 : \quad a(\tilde{u}, v) &= F(v) - a(Ru, v), & \forall v \in V_0. \end{aligned} \quad (2.20)$$

What was changed by this steps is both the function space, that now includes a homogeneous Dirichlet boundary condition, and the right-hand side that needs to take into account the chosen lifting function.

After solving the problem defined by (2.20), the solution \tilde{u} needs to be added to the lifting Ru , thus ensuring the solution to have exactly the required value at the Dirichlet boundary.

2.6 NURBS patches

In the case of a complex geometry, it can happen that just using a single transformation from the unit square to the geometry might lead to unwanted values of the Laplacian of the transformation, that is either too small or too big to be safe for numerical reasons. In those cases, the physical geometry can be divided into different patches, each of which will be parametrized with a different NURBS surface.

This means that, instead of equation (2.10), a physical domain Ω might be discretized by

$$\begin{aligned} S^{(k)} : [0, 1]^2 &\longrightarrow \Omega^{(k)} \subset \Omega \subset \mathbb{R}^3 \\ (\xi^{(1)}, \xi^{(2)}) &\longmapsto S^{(k)}(\xi^{(1)}, \xi^{(2)}) = \frac{\sum_{i=1}^{n^{(k,1)}} \sum_{j=1}^{n^{(k,2)}} B_{i,j}^{(k)}(\xi^{(1)}, \xi^{(2)}) \omega_{i,j}^{(k)} \mathbf{P}_{i,j}^{(k)}}{\sum_{i=1}^{n^{(k,1)}} \sum_{j=1}^{n^{(k,2)}} B_{i,j}^{(k)}(\xi^{(1)}, \xi^{(2)}) \omega_{i,j}^{(k)}}, \end{aligned}$$

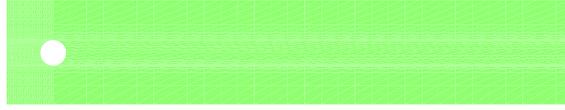


Figure 2.7: Rectangular domain without a circle

where $k = 1, \dots, M$ and M is the number of patches used, such that the intersection between two patches is at most a NURBS curve.

By dividing in multiple patches, all domains that would bring to a nearly singular Laplacian can now be modelled as well. The difficulty connected to the use of multiple patches has to do with the imposition of continuity along the borders of the patches. Numerically, what has to be done is to keep track of the basis functions that do not vanish at the interface and impose the solution to have the same value at both sides of the interface.

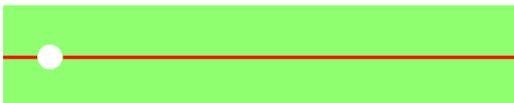
Example 19. A domain often used as a benchmark problem, see [14], is the domain that will be called, from now on, "rectangle without a circle". By this, the domain delined by a rectangle of dimensions $[-0.2, 2.0] \times [-0.2, 0.21]$, where a circle of diameter 0.1 and center $(0, 0)$ has been subtracted. This domain is a 2-dimensional simplification of the flow around an obstacle, that is a typical situation in applications. The interest of this domain in this project is the possibility of dividing it in patches in various ways. A representation of the domain can be seen in Figure 2.7. Possible decompositions of this domain that will be considered in the following chapters are shown in Figure 2.8.



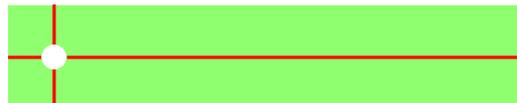
(a) Single patched domain, with internal boundary for $x = 0$, $-0.2 \leq y \leq -0.1$



(b) Domain divided in two patches along $x = 0$



(c) Domain divided in two patches along $y = 0$



(d) Domain divided both along $x = 0$ and $y = 0$

Figure 2.8: Four possible divisions in patches of the domain in Example 19

3

Stokes equations

In this chapter, Stokes equations are presented together with the numerical aspects connected with their discretization and numerical solution. Some solutions to these issues are also reported and numerical tests have been carried out to support our claims.

Stokes equations model the flow of a fluid when the ratio between inertia and viscosity forces in the flow tends to zero, as described in [10]. This means that the Stokes equations are a simplification to the limit of a set of more complex equations, the Navier-Stokes equations, that will be presented in Chapter 4. For a general understanding of the applications of Stokes and Navier-Stokes equations, we refer the interested reader to [6].

The Stokes equations are often used as a first step towards the approximation of the Navier-Stokes equations. This is due to the similarity of the two problems that permits to use the solution of the Stokes equations as a first approximation of the solution of the Navier-Stokes equations. Moreover, the Stokes problem already introduces some of the numerical issues connected with the solution of the Navier-Stokes equations, while removing the nonlinearity of the more general problem. For these reasons, we start by solving the Stokes problem.

3.1 Setting the problem

Stokes equations are the limit of Navier-Stokes equations in case of the Reynolds number tending to zero. The Reynolds number is a dimensionless number that takes into account the relative importance of inertial forces with respect to viscous forces. It is defined as

$$Re = \frac{V L}{\nu}, \quad (3.1)$$

where V is a characteristic velocity of the flow, L is a characteristic length and ν is the kinematic viscosity.

A flow that satisfy the condition $Re \ll 1$ can be modelled by the Stokes equations, without needing the more complex Navier-Stokes equations. In this project, the only case treated will be the steady incompressible flow. This restriction is based on the fact that the steady case already introduces all the issues connected with the numerical discretization and that the incompressible setting is the most studied in literature, also from an applicative point of view such as in [17].

The steady Stokes equations for incompressible flows read

$$-\nu \Delta \mathbf{u} + \nabla p = \mathbf{f}, \quad \text{in } \Omega, \quad (3.2)$$

$$\nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega, \quad (3.3)$$

$$\mathbf{u}|_{\Gamma_D} = \mathbf{u}_D, \quad \text{on } \Gamma_D, \quad (3.4)$$

$$\left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} \right) \Big|_{\Gamma_N} = \boldsymbol{\psi}, \quad \text{on } \Gamma_N, \quad (3.5)$$

where the following notation is used

$\Omega \subset \mathbb{R}^2$	domain,
$\nu \in \mathbb{R}$	kinematic viscosity,
$\mathbf{u} \in [H_D^1(\Omega)]^2$	flow velocity,
$p \in L^2(\Omega)$	pressure,
$\mathbf{f} \in L^2(\Omega)$	forcing term,
Γ_D, Γ_N	partition of the boundary $\partial\Omega$,
$\mathbf{u}_D \in [H^{\frac{1}{2}}(\Gamma_D)]^2, \boldsymbol{\psi} \in [L^2(\Gamma_N)]^2$	two given functions.

Remark 20. In case of $\Gamma_N = \emptyset$, the pressure is required to be in the space $L_0^2(\Omega)$, because it would be defined up to a constant.

The notation is set to be

$$\begin{aligned} [H_D^1(\Omega)]^2 &= \left\{ \mathbf{v} \in [H^1(\Omega)]^2 : \mathbf{v}|_{\Gamma_D} = \mathbf{u}_D \right\}, \\ [H_0^1(\Omega)]^2 &= \left\{ \mathbf{v} \in [H^1(\Omega)]^2 : \mathbf{v}|_{\partial\Omega} = \mathbf{0} \right\}, \\ L_0^2(\Omega) &= \left\{ p \in L^2(\Omega) : \int_{\Omega} p \, d\mathbf{x} = 0 \right\}. \end{aligned}$$

As for Finite Element method, also Isogeometric Analysis needs to be applied on the weak form of the problem. For a more comprehensive approach, we refer to [19].

The problem (3.2) presented in weak formulation with homogeneous Dirichlet boundary conditions on all the boundary, i.e., $\Gamma_N = \emptyset$, is as follows:
find $\mathbf{u} \in V = [H_0^1(\Omega)]^2$ and $p \in Q = L_0^2(\Omega)$ such that

$$\begin{cases} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (\mathbf{f}, \mathbf{v}) & \forall \mathbf{v} \in V, \\ b(\mathbf{u}, q) = 0 & \forall q \in Q, \end{cases}$$

where

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\mathbf{x}, \\ b(\mathbf{u}, q) &= - \int_{\Omega} q \nabla \cdot \mathbf{u} \, d\mathbf{x}, \\ (\mathbf{f}, \mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}. \end{aligned}$$

If non-homogeneous Dirichlet boundary conditions are imposed, a lifting function $R\phi \in [H^1(\Omega)]^2$ that fulfills Dirichlet boundary conditions needs to be introduced, as presented in Section 2.5.1. The solution will be therefore split into $\tilde{\mathbf{u}} = \mathbf{u} - R\phi$ and homogeneous Dirichlet boundary conditions will be applied to $\tilde{\mathbf{u}}$.

If both Dirichlet and Neumann non-homogeneous boundary conditions are introduced, the problem becomes find $\tilde{\mathbf{u}} \in V = [H_D^1(\Omega)]^2$ and $p \in L^2(\Omega)$ such that

$$\begin{cases} a(\tilde{\mathbf{u}}, \mathbf{v}) + b(\mathbf{v}, p) = \mathbf{F}(\mathbf{v}) & \forall \mathbf{v} \in V, \\ b(\tilde{\mathbf{u}}, q) = G(q) & \forall q \in Q, \end{cases} \quad (3.6)$$

where

$$\begin{aligned} \mathbf{F}(\mathbf{v}) &= (\mathbf{f}, \mathbf{v}) + \int_{\Gamma_N} \boldsymbol{\psi} \cdot \mathbf{v} \, d\sigma - a(R\phi, \mathbf{v}), \\ G(q) &= -b(R\phi, q). \end{aligned}$$

By choosing discrete spaces $V_h \subset V$ and $Q_h \subset Q$, of dimensions respectively n and m , and setting the basis functions to be respectively $\{\phi_j \in V_h\}_{j=1}^n$ and $\{\psi_k \in Q_h\}_{k=1}^m$, the discretized functions $\mathbf{u}_h \in V_h$ and $p_h \in Q_h$, approximations of $\mathbf{u} \in V$ and of $p \in Q$ respectively, can be written as

$$\mathbf{u}_h(\mathbf{x}) = \sum_{j=1}^n \phi_j(\mathbf{x}) u_j, \quad p_h(\mathbf{x}) = \sum_{k=1}^m \psi_k(\mathbf{x}) p_k,$$

and satisfy the discrete Stokes problem

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}, \quad (3.7)$$

with

$$A_{ij} = a(\phi_j, \phi_i), \quad B_{ij} = b(\phi_j, \psi_i) \quad \text{and} \quad \mathbf{F}_i = (\mathbf{f}, \phi_i).$$

By this steps, the Stokes problem presented in Equation (3.2) is approximated solving the linear system (3.7). This linear problem is well-posed if the choice of the spaces V_h and Q_h is compatible; in fact it can be proven that the matrix of the problem is non-singular if and only if one of the so-called inf-sup condition is satisfied. This conditions set a constraint

on the couple of spaces that can be chosen to solve the Stokes problem. It is known in FEM that some pairs of finite dimensional function spaces are unstable, therefore leading to ill-conditioned or singular systems. The same holds for the IGA setting, it is therefore required to find stable pairs of finite dimensional NURBS spaces. In the following, Taylor-Hood coupled spaces are proposed and tested.

Theorem 21. *The approximation of the Stokes problem (3.7) admits a unique solution if the following holds*

- the bilinear form $a(\cdot, \cdot)$ is coercive, that is there exists a strictly positive constant α such that

$$a(\mathbf{v}_h, \mathbf{v}_h) \geq \alpha \|\mathbf{v}_h\|_V^2 \quad \forall \mathbf{v}_h \in V_h^*,$$

where $V_h^* = \{\mathbf{v}_h \in V_h : b(\mathbf{v}_h, q_h) = 0, \forall q_h \in Q_h\}$,

- the bilinear form $a(\cdot, \cdot)$ is continuous, that is there exists a strictly positive constant γ such that

$$|a(\mathbf{u}_h, \mathbf{v}_h)| \leq \gamma \|\mathbf{u}_h\|_V \|\mathbf{v}_h\|_V \quad \forall \mathbf{u}_h, \mathbf{v}_h \in V_h,$$

- the bilinear form $b(\cdot, \cdot)$ is continuous, that is there exists a strictly positive constant δ such that

$$|b(\mathbf{v}_h, q_h)| \leq \delta \|\mathbf{v}_h\|_V \|q_h\|_Q \quad \forall \mathbf{v}_h \in V_h, \forall q_h \in Q_h,$$

- there exists a strictly positive constant β such that

$$\inf_{q_h \in Q_h \setminus \{0\}} \sup_{\mathbf{v}_h \in V_h \setminus \{0\}} \frac{b(\mathbf{v}_h, q_h)}{\|\mathbf{v}_h\|_V \|q_h\|_Q} \geq \beta. \quad (3.8)$$

This later requirement is called Brezzi's inf-sup condition.

The subscript h has been introduced to characterize the discretized values, for similarity with the FEM case. In this case, h is an indicator of the refinement of the mesh used. In the IGA setting, the value of h is defined as follows.

Definition 22. The value of the parameter h , that stands for the refinement of the grid used, is defined by

$$h = \min_{Q \in \mathcal{Q}_h} \text{diam } Q,$$

where \mathcal{Q}_h is defined to be the two dimensional mesh in the parametrized domain, i.e.,

$$\mathcal{Q}_h = \left\{ Q \subset [0, 1]^2 : Q = \bigotimes_{k=1}^2 \left(\xi_{i_k}^{(k)}, \xi_{i_k+1}^{(k)} \right), \text{ for } \xi_{i_k}^{(k)}, \xi_{i_k+1}^{(k)} \in \Xi_k \right\}.$$

Remark 23. The Theorem 21 itself does not require β to be independent from the parameter h , the refinement of the mesh, but an interesting choice of spaces would be such that β would not change dramatically while refining the mesh. If β tends to zero while refining the mesh, the problems are twofold. First of all, the linear system (3.7) would become ill-conditioned; secondly the approximation error would not be bounded while the mesh is refined. This latest fact is due to the approximation error bound by a constant depending on the inverse of β , as will be presented in Theorem 24.

The following theorem presents a first error approximation and is therefore called approximation theorem. It states that the numerical error will decrease while the numerical spaces V_h and Q_h tend to the analytic spaces V and Q and has first been proved in [3].

Theorem 24. *With the hypothesis and the notation introduced in the previous Theorem 21, the approximation error satisfies the optimal error estimate*

$$\|(\mathbf{u}_h, p_h) - (\mathbf{u}, p)\|_{V \times Q} \leq C \inf_{(\mathbf{v}_h, q_h) \in V_h \times Q_h} \|(\mathbf{v}_h, q_h) - (\mathbf{u}, p)\|_{V \times Q}, \quad (3.9)$$

where

$$\|(v, q)\|_{V \times Q} = \|v\|_V + \|q\|_Q.$$

More precisely, it holds

$$\|\mathbf{u} - \mathbf{u}_h\| \leq \left(1 + \frac{\gamma}{\beta}\right) \left(1 + \frac{\gamma}{\alpha}\right) \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_V + \frac{\delta}{\alpha} \inf_{q_h \in Q_h} \|p - q_h\|_Q,$$

and

$$\|p - p_h\|_Q \leq \frac{\gamma}{\beta} \left(1 + \frac{\gamma}{\alpha}\right) \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_V + \left(1 + \frac{\delta}{\beta} + \frac{\delta\gamma}{\alpha\beta}\right) \inf_{q_h \in Q_h} \|p - q_h\|_Q.$$

As anticipated in the previous remark, in this theorem the role of the constant β in the upper bound of the error is described. The inf-sup constant always appears in the denominator, therefore having β tending to zero with respect to h would not insure the convergence of the method.

3.2 Taylor-Hood spaces

While proceeding with the discretization of the Stokes problem, it is important to satisfy the inf-sup condition stated in (3.8) by an accurate choice of the discrete spaces for velocity and pressure. As a matter of fact, this requirement imposes non-trivial choices for the coupled spaces V_h and Q_h .

Two major issues need to be considered. First, the choice of appropriate coupled spaces, then a method to calculate the inf-sup constant depending on the chosen spaces and on the chosen mesh. For the first problem, we refer to [4] and for the second to [15].

In the FEM setting, coupled spaces that satisfy the inf-sup condition are the Taylor-Hood spaces. For a comprehensive introduction to such spaces in the frame of FEM, we refer the interested reader to [20]. Taking as intuitive starting point the Taylor-Hood spaces in FEM, our main interest here is to construct similar spaces in the setting of IGA that satisfy the inf-sup condition.

Remembering the notation introduced in Definition 2 and considering two-dimensional problems, one couple of IGA spaces that satisfy the discrete inf-sup condition is introduced.

Definition 25. Given two knot vectors $\Xi^{(1)}$ and $\Xi^{(2)}$ of degree 0, and two integers p and r with $r \leq p$, then the Taylor-Hood IGA space is defined as

$$\begin{aligned} V_h^{TH} &= \mathcal{S}\left(\Xi^{(1)}(p+1, r), \Xi^{(2)}(p+1, r)\right) \times \mathcal{S}\left(\Xi^{(1)}(p+1, r), \Xi^{(2)}(p+1, r)\right), \\ Q_h^{TH} &= \mathcal{S}\left(\Xi^{(1)}(p, r), \Xi^{(2)}(p, r)\right). \end{aligned} \quad (3.10)$$

There is numerical evidence that the Taylor-Hood spaces so defined satisfy the inf-sup condition, as stated in [4]. In order to support this assertion, let us here present a numerical method to calculate the discrete inf-sup constant β . It will be shown that for the presented spaces such constant does not tend to zero while refining the grid. The next discussion will be held in a more abstract way, such that it will be easier later to apply it to the Navier-Stokes case. We refer the interested reader to the discussion in [2] and in [15] for a more complete presentation of the calculation of the discrete inf-sup constant in Stokes and Navier-Stokes discretizations.

3.3 Calculating the inf-sup constant

Let us here consider two Hilbert spaces X and Y with discrete finite-dimensional subspaces X_h and Y_h of dimensions n and m respectively, and a continuous bilinear form $\mathbf{A}(\cdot, \cdot) : X \times Y \rightarrow \mathbb{R}$. In order to calculate the discrete inf-sup constant defined as

$$\beta_h = \inf_{\mathbf{v}_h \in X_h} \sup_{\mathbf{q}_h \in Y_h} \frac{\mathbf{A}(\mathbf{v}_h, \mathbf{q}_h)}{\|\mathbf{v}_h\|_X \|\mathbf{q}_h\|_Y}, \quad (3.11)$$

let us introduce some notation. Let $\{\phi_i\}_{i=1}^n$ and $\{\psi_i\}_{i=1}^m$ be two bases of $X_h \subset X$ and $Y_h \subset Y$, and the norm matrices $\mathbb{X} \in \mathbb{R}^{n,n}$ and $\mathbb{Y} \in \mathbb{R}^{m,m}$ be defined as

$$\mathbb{X}_{ij} = (\phi_i, \phi_j)_X \quad \mathbb{Y}_{ij} = (\psi_i, \psi_j)_Y$$

and the matrix $M \in \mathbb{R}^{n,m}$ is such that

$$M_{ij} = \mathbf{A}(\phi_j, \psi_i).$$

It can be shown, as in [15], that the discrete inf-sup constant (3.11) satisfies

$$\beta_h^2 = \inf_{\mathbf{v} \in \mathbb{R}^n} \frac{\mathbf{v}^T M^T \mathbb{Y}^{-1} M \mathbf{v}}{\mathbf{v}^T \mathbb{X} \mathbf{v}}.$$

This is equivalent to compute the minimum eigenvalue of the generalized eigenvalue problem

$$M^T \mathbb{Y}^{-1} M \mathbf{v} = \lambda \mathbb{X} \mathbf{v}, \quad (3.12)$$

and taking the square root, having therefore $\beta_h = \sqrt{\lambda_{\min}}$.

For a saddle point problem such as the Stokes problem, the interpretation of the previous procedure can be twofold. In fact, the bilinear continuous form $\mathbf{A}(\cdot, \cdot)$ can either be interpreted as the mixed form $b(\cdot, \cdot)$ of the Stokes formulation, or as the full left-hand side

of the system (3.6). These two interpretations brings to different inf-sup constants, but have been proved to be equivalent in [9] under the condition of $a(\cdot, \cdot)$ being continuous and coercive.

Let us denote by \mathbb{V} the norm matrix of the space V_h and by \mathbb{Q} the norm matrix of the space Q_h , as introduced in the Section 3.1. Let us first set $\mathbf{A}(\cdot, \cdot) = b(\cdot, \cdot)$. This approach has been introduced by Brezzi in [11], and leads therefore to the *Brezzi inf-sup constant*. In this case, the generalized eigenvalue problem (3.12) becomes

$$B^T \mathbb{Q} B \mathbf{v} = \lambda_{Br} \mathbb{V} \mathbf{v},$$

and the *Brezzi inf-sup constant* $\beta_{Br,h}$ is defined as

$$\beta_{Br,h} = (\min \{\lambda_{Br}\})^{\frac{1}{2}}.$$

In the second case presented, firstly introduced in [1] by Babuška, the choice is to set

$$\mathbf{A}((\mathbf{u}, p), (\mathbf{v}, q)) = \begin{pmatrix} a(\mathbf{u}, \mathbf{v}) & b(\mathbf{v}, p) \\ b(\mathbf{u}, q) & 0 \end{pmatrix}.$$

This choice, inserted in Equation (3.12), leads to

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}^T \begin{pmatrix} \mathbb{V} & 0 \\ 0 & \mathbb{Q} \end{pmatrix} \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix} = \lambda_{Ba} \begin{pmatrix} \mathbb{V} & 0 \\ 0 & \mathbb{Q} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix},$$

and define the *Babuška inf-sup constant* $\beta_{Ba,h}$ by

$$\beta_{Ba,h} = (\min \{\lambda_{Ba}\})^{\frac{1}{2}}.$$

Remark 26. In case the procedure is applied to Stokes or Navier-Stokes problem, when the pressure is defined up to a constant, i.e., when just Dirichlet boundary conditions are imposed, the first eigenvalue is zero and the first eigenvector is the constant vector for the pressure unknowns, reflecting the definition of the pressure up to a constant. It has been proved in [5] that in such a case the first non-zero eigenvalue needs to be taken instead of the first one.

3.4 Numerical tests

In this sections, some numerical tests for the Stokes problem are presented. We will here concentrate on problems defined in the unit square. The main point of the section is to test the inf-sup condition, therefore calculating Brezzi and Babuška discrete inf-sup constants in a variety of cases, while changing the polynomial degrees of the basis functions used, their continuity or the parametrization of the unit square.

The implementation was runned in MATLAB, using GMRES as the solver of the linear system. As rule of thumb, the code would take around half an hour to solve a problem with 500 degrees of freedom. Most of the computational time is spend on assembling the matrices of the linear system.

3.4.1 Poiseuille flow

The first application considered is the Poiseuille flow, that is the modelization of a viscous flow in a pipe. As physical domain, the unit square is chosen, $\Omega = (0, 1)^2$, with Dirichlet boundary conditions on all the boundary, more precisely

$$\begin{cases} \mathbf{u}(x, 0) = \mathbf{u}(x, 1) = \mathbf{0}, \\ \mathbf{u}(0, y) = \mathbf{u}(1, y) = \begin{pmatrix} y(1-y) \\ 0 \end{pmatrix}. \end{cases}$$

Setting $\nu = 1$, the right-hand side is given such that the analytical solution is

$$\begin{aligned} \mathbf{u}(x, y) &= \begin{pmatrix} y(1-y) \\ 0 \end{pmatrix}, \\ p(x, y) &= \frac{1}{2} - x. \end{aligned}$$

Since the pressure appears just as a derivative, it can be determined only up to a constant. In order to fix the constant, there exist two possibilities. Either the integral of the pressure is set to zero

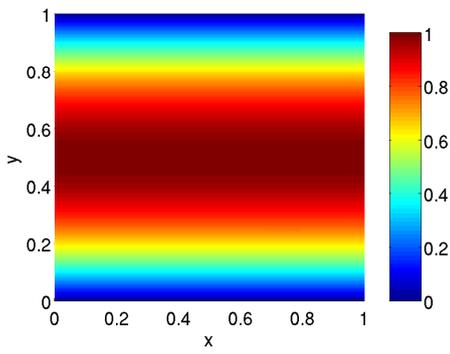
$$\int_{\Omega} p_h \, d\mathbf{x} = 0$$

and $p_h \in L_0^2(\Omega)$, or the value of one component of the discrete pressure is defined a priori.

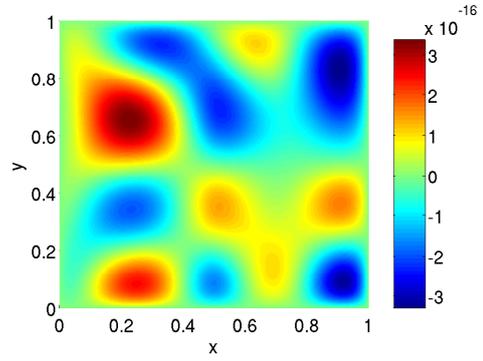
An interesting feature of this flow is that, as soon as a mesh allowing order 2 velocity is used, i.e., $p = 1$ in the definition of the Taylor-Hood spaces, the exact solution belongs to the space of numerical solutions: $(\mathbf{u}, p) \in V_h \times Q_h$. This means that the right-hand side of Equation (3.9) is null and, by Theorem 24, the error $\|(\mathbf{u}_h, p_h) - (\mathbf{u}, p)\|_{V \times Q}$ is zero as well. The numerical results confirm the theory, giving L^2 errors for velocity and pressure of the order of the machine error.

In Figure 3.1, the solution of the Poiseuille flow is plotted for a coarse mesh using Taylor-Hood elements. The space used for the pressure is based on the knot vector $\Xi_p = \{0, 0, 0.5, 1, 1\}$ for both x and y dimension, where the subscript p is just used to indicate all variables that deals with pressure alone. On the other hand, the velocity space has knot vector $\Xi = \{0, 0, 0, 0.5, 0.5, 1, 1, 1\}$. These spaces satisfy the formula (3.10) with $p = 1$ and $r = 1$.

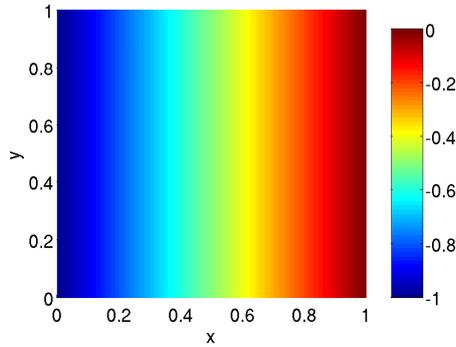
Once defined the mesh, and therefore the spaces used, it is possible to calculate both the Brezzi and Babuška constants for the specific problem treated. The results of such computations can be seen in Figure 3.2 and in Table 3.1. The two plots 3.2a and 3.2b present the evolution of the Brezzi and Babuška discrete inf-sup constants, respectively. Two remarks are interesting in this context. First of all, the two plots have similar behavior, thus intuitively confirming the connection between Brezzi and Babuška constants, secondly they tend to a constant while refining the grid. This second point supports our claim that the Taylor-Hood spaces satisfy the discrete inf-sup condition. Finally, these results are in agreement with the results presented in [4].



(a) First component of the velocity u_1 for the Poiseuille flow



(b) Second component of the velocity u_2 for the Poiseuille flow, numerically zero



(c) Pressure p for the Poiseuille flow

Figure 3.1: Numerical solution of the Poiseuille flow, two components of velocity and the pressure. Consider that the values of the second component are close to the machine error.

h	0.500000	0.250000	0.125000	0.062500	0.031250	0.015625
β_{Ba}	0.200667	0.191751	0.184198	0.178779	0.174859	0.171926
β_{Br}	0.485661	0.475558	0.465394	0.457835	0.452302	0.448124

Table 3.1: Brezzi and Babuška inf-sup constants for $p = 1$ and $r = 1$ in the unit square as a function of the refinement

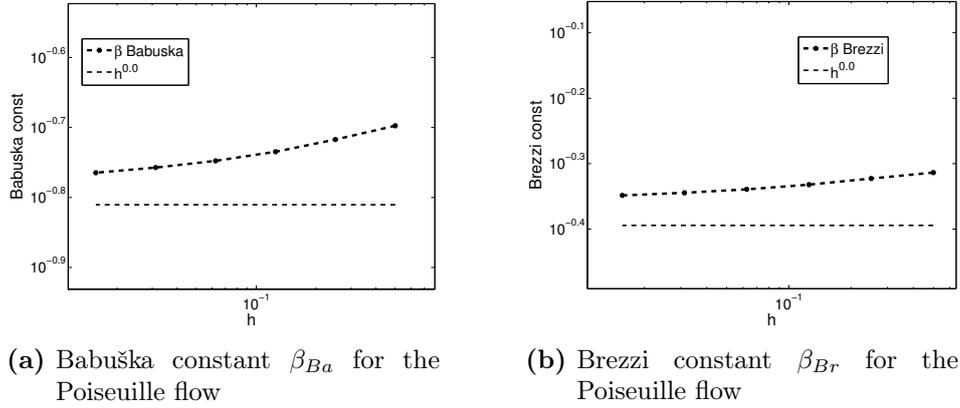


Figure 3.2: Brezzi and Babuška inf-sup constants for the Poiseuille flow for $p = 1$ and $r = 1$ as a function of the refinement

3.4.2 Vortex flux

A more complex example for the Stokes problem in the unit square is the following vortex flux. Homogeneous Dirichlet boundary conditions are imposed on all the boundary and the forcing term is calculated in order to obtain the solution

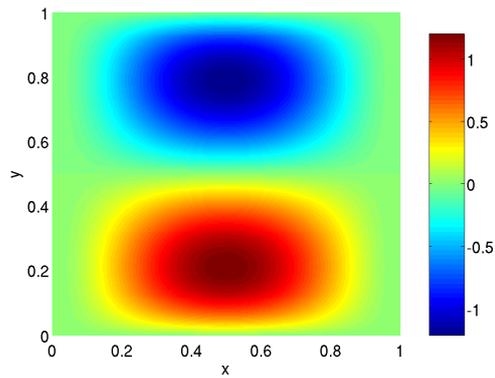
$$\mathbf{u} = 200 \begin{pmatrix} x^2(1-x)^2y(1-y)(1-2y) \\ -x(1-x)(1-2x)y^2(1-y)^2 \end{pmatrix}, \quad (3.13)$$

$$p = 10 \left(\left(x - \frac{1}{2} \right)^3 y^2 + (1-x)^3 \left(y - \frac{1}{2} \right)^3 \right).$$

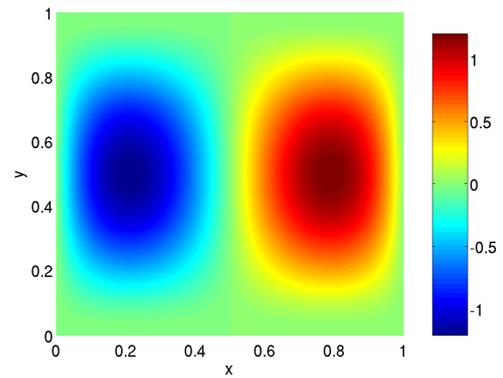
The numerical results can be seen in Figure 3.3. The differences between this example and the previous one are multiple. First of all, the solution for the vortex flow does not belong to the second order Taylor-Hood spaces, therefore, by using the same meshes as previously and discretizing just by knot insertion the exact solution will never belong to the space $V_h \times Q_h$. Since the solution is polynomial of order 4 in each variable for the velocity and of order 3 for the pressure, the analytical solution would belong to the numerical space $V_h \times Q_h$ only if $p > 2$ is chosen. The first consequence is that the convergence of the error as a function of the discretization gets more interesting, and for Taylor-Hood spaces with $p = 1$ and $r = 1$ it is plotted in Figure 3.4, dividing again between the velocity and the pressure. After the first steps, it can be seen how the error decreases constantly while refining the grid. The convergence rate of the velocity is of order 3, higher than the convergence rate for the pressure, 2, in agreement with the results presented in [18].

For $p > 2$, the exact solution was recovered in the numerical studies.

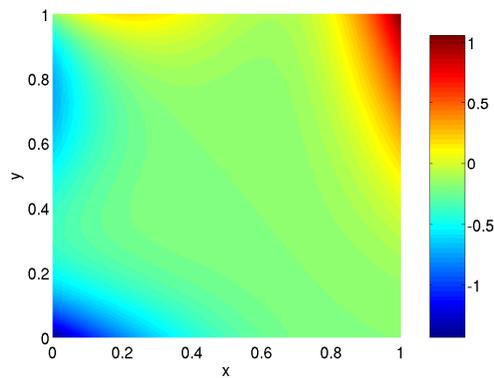
For what deals with the inf-sup constants, we remark that, by choosing the same spaces as in Section 3.4.1, the matrices needed here for the solution of the linear system do not differ from the ones used in the Poiseuille case, therefore leading to a linear system that has



(a) First component of the velocity u_1 for the vortex flux



(b) Second component of the velocity u_2 for the vortex flux



(c) Pressure p for the vortex flux

Figure 3.3: Numerical solution of the vortex flux, two components of velocity and the pressure

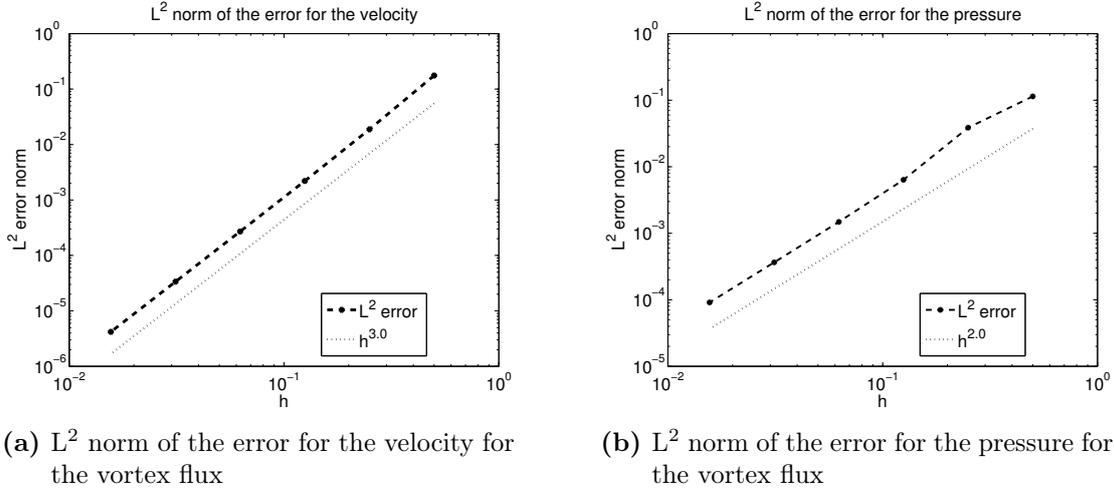


Figure 3.4: Convergence of the L^2 norm of the error for the vortex flux as a function of the grid width with Taylor-Hood spaces with $p = 1$, $r = 1$

equal left-hand side and equal norm matrices with respect to the previous one, just different right-hand side. This leads to the same discrete inf-sup constants as in the Poiseuille flow, plotted in Figure 3.2.

3.4.3 Different discretizations of the square

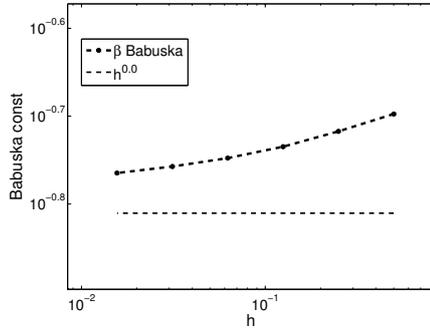
Further tests can be made changing the meshing of the domain and calculating the new inf-sup constants derived from these different discretizations. As example, here we present the results obtained on a "rotated domain" and a "splitted domain", two different discretizations of the unit square also applied to solve the vortex problem.

In the "rotated domain", the mapping between the unit square and the physical domain introduce a rotation of $\frac{\pi}{2}$ around the center of the domain, i.e., as example the point (0,0) in the unit square will be mapped into the point (1,0) of the physical domain. Distortion are not introduced. Taylor-Hood spaces have been used with $p = 1$ and $r = 1$ with respect to the notation used in Equation (3.10).

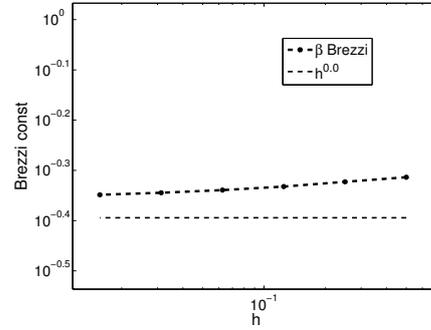
This rotation affects both the inf-sup constants, as can be seen in Figure 3.5, where the values of the Brezzi and Babuška inf-sup constants are plotted as a function of the refinement of the grid. Even if the numerical values change, the plots show a behavior similar to the one already encountered in the previous discretization. These results therefore confirm the stability of the chosen coupled spaces.

Let us remark that the rotation does not affect the order of convergence of the solution. In particular, if the Taylor-Hood spaces are chosen with $p > 2$ the numerical error will be comparable to the machine error.

By "split domain", a unit square is constructed by two patches, splitted along $y = \frac{1}{2}$.

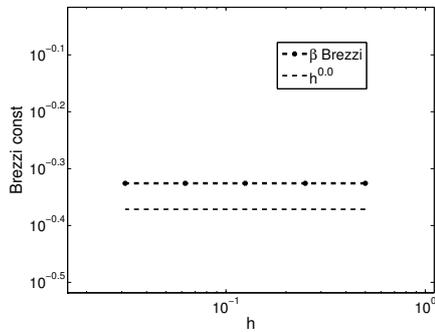


(a) Babuška constant β_{Ba} for the rotated domain

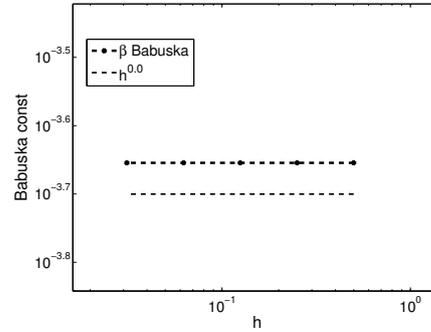


(b) Brezzi constant β_{Br} for the rotated domain

Figure 3.5: Brezzi and Babuška inf-sup constants for the rotated domain as a function of the refinement with Taylor-Hood spaces with $p = 1$ and $r = 1$



(a) Brezzi constant β_{Br} for the unit square in two patches



(b) Babuška constant β_{Ba} for the unit square in two patches

Figure 3.6: Discrete inf-sup constants for the unit square divided in two patches

This example is interesting to test how the division into patches can affect the discrete inf-sup constants. Results regarding the inf-sup constants are displayed in Figure 3.6 and show how little those values change in this context.

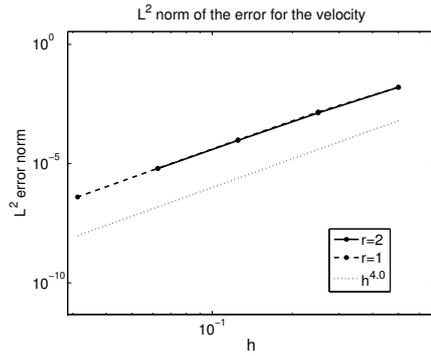
3.4.4 Higher order test functions

In this subsection, some more tests concerning the ideas presented in [4] are carried on, in particular, considering the possibility of augmenting the polynomial order of the V_h and Q_h spaces and changing the order of continuity at the knots.

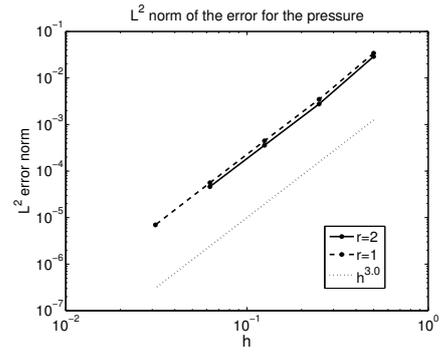
Let us here consider the unit square with homogeneous Dirichlet boundary conditions on the boundaries. By using order elevation, the order of the test functions used can also be augmented, while by knot insertion the continuity of the test functions can be decreased. The possibility of having higher order test functions is interesting, in particular to model solution that are known to be smooth, because in such case the right-hand side of the upper bound of the error presented in Theorem 24 converges faster. In this subsection, some numerical tests on the unit square domain meshed with knot vectors of higher order and different continuity are presented. The test are done by changing the value of p and r in the definitions of the Taylor-Hood spaces.

The results presented are referring to the vortex flux problem, as stated in Equation (3.13) with Taylor-Hood spaces (3.10) with $p = 2, 3, 4, 5$ and $r = 1$. The outcomes are plotted in Figures 3.7 and 3.8, 3.9 and 3.10 respectively. For $p = 2$ and $p = 3$, the behavior of all the plots is similar to what was already remarked for $p = 1$, that is, the behavior of the inf-sup constants is constant along refinements, even if a difference can be seen in the first refinements. In the higher orders, that is for $p = 4$ and $p = 5$, the inf-sup constants take a particularly low value for the coarsest mesh. But from the first refinement onwards, the result about the inf-sup constants confirms that they do not depend on h and therefore that the chosen spaces are stable.

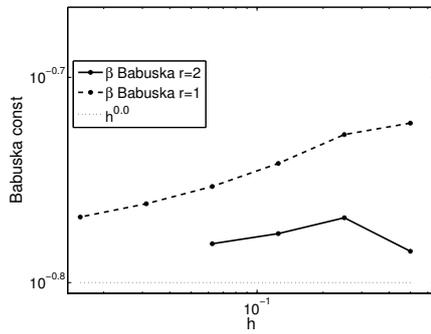
Error plots are not shown for $p > 2$ because the numerical error gets to be of the order of the machine error already from the first refinement.



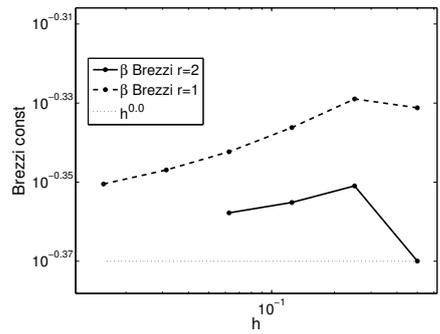
(a) L^2 norm of the error for the velocity for $p = 2$ and $r = 1, 2$ for the vortex flux



(b) L^2 norm of the error for the pressure for $p = 2$ and $r = 1, 2$ for the vortex flux

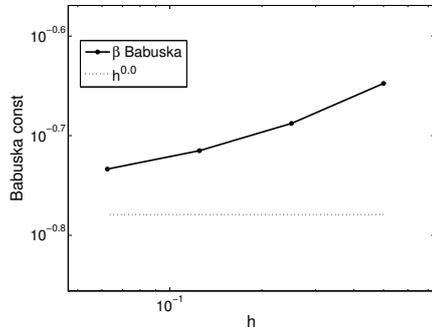


(c) Babuška constant β_{Ba} for $p = 2$ and $r = 1, 2$ for the vortex flux

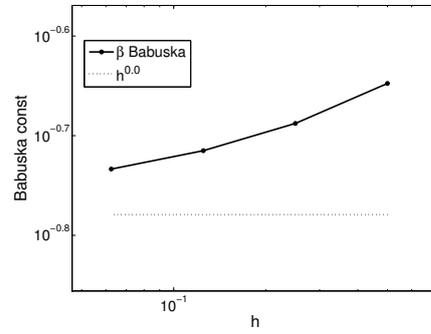


(d) Brezzi constant β_{Br} for $p = 2$ and $r = 1, 2$ for the vortex flux

Figure 3.7: Numerical results for the vortex flux with $p = 2$ and $r = 1, 2$

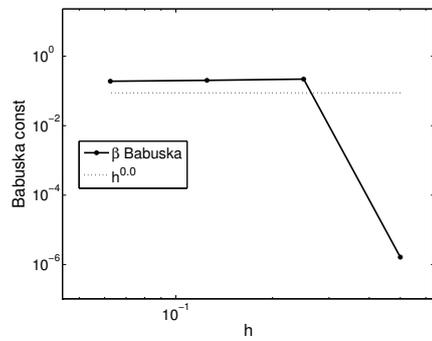


(a) Babuška constant for $p = 3$ and $r = 1$ for the vortex flux

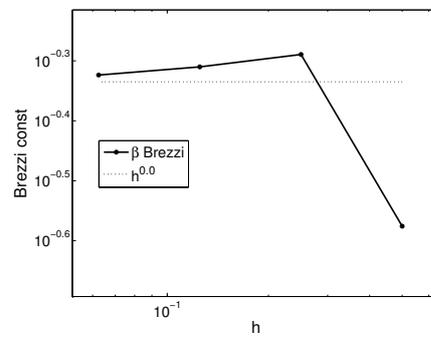


(b) Brezzi constant for $p = 3$ and $r = 1$ for the vortex flux

Figure 3.8: Numerical results for the vortex flux using Taylor-Hood elements with $p = 3$ and $r = 1$

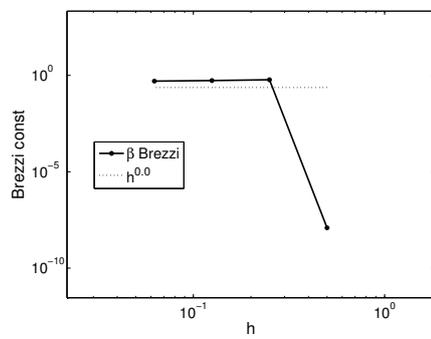


(a) Babuška constant for $p = 4$ and $r = 1$ for the vortex flux

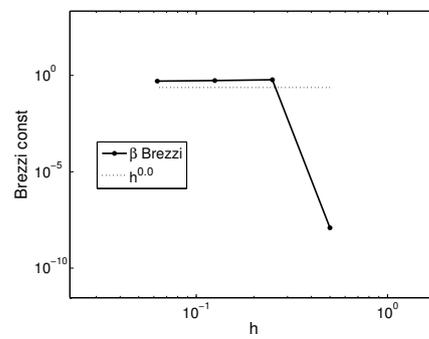


(b) Brezzi constant for $p = 4$ and $r = 1$ for the vortex flux

Figure 3.9: Numerical results for the vortex flux using Taylor-Hood elements with $p = 4$ and $r = 1$



(a) Babuška constant for $p = 5$ and $r = 1$ for the vortex flux



(b) Brezzi constant for $p = 5$ and $r = 1$ for the vortex flux

Figure 3.10: Numerical results for the vortex flux using Taylor-Hood elements with $p = 5$ and $r = 1$

4

Navier-Stokes equations

The aim of this chapter is to apply Isogeometric Analysis to solve the Navier-Stokes equations. This set of equations is strongly related to the Stokes equations presented in the previous chapter but is more general, compared to them, since it permits to consider also flows where the Reynolds number is not negligible. These equations are used in a variety of fields, from aircraft modeling to sailing boats simulations, in order to model the behavior of fluids. In practical applications, the Navier-Stokes equations are seldomly solvable analytically, but with the increase of computational power and the advances in numerical methods, recently more and more problems could be simulated numerically. For an accurate presentation of Navier-Stokes equations and an overview of most cases where the analytical solution is known, we refer to [12].

In this chapter, the numerical simulation of Navier-Stokes equations is considered for a steady, incompressible, two-dimensional flow using IGA. After presenting the problem, some words are spent on the discretization of a non-linear saddle point problem and finally the numerical results are presented. As in the previous chapter, the numerical tests will be at first carried out the square domain. Afterwards the domain considered will be the rectangle without a circle, with the discretizations introduced in Figure 2.8.

4.1 Equations

The Navier-Stokes equations are defined as follows

$$\begin{aligned}
 -\nu \Delta \mathbf{u} + \nabla p + (\mathbf{u} \cdot \nabla) \mathbf{u} &= \mathbf{f}, & \text{in } \Omega, \\
 \nabla \cdot \mathbf{u} &= 0, & \text{in } \Omega, \\
 \mathbf{u}|_{\Gamma_D} &= \mathbf{u}_D, & \text{on } \Gamma_D \\
 \left(\nu \frac{\partial \mathbf{u}}{\partial \mathbf{n}} - p \mathbf{n} \right) \Big|_{\Gamma_N} &= \boldsymbol{\psi}, & \text{on } \Gamma_N,
 \end{aligned} \tag{4.1}$$

where the data of the problem are

$\Omega \subset \mathbb{R}^2$	physical domain,
$\rho, \nu \in \mathbb{R}$	density and kinematic viscosity,
Γ_D, Γ_N	partition of $\partial\Omega$,
$\mathbf{f} \in L^2(\Omega)$	forcing term,
$\mathbf{u}_D \in [H^{\frac{1}{2}}(\Gamma_D)]^2, \boldsymbol{\psi} \in [L^2(\Gamma_N)]^2$	two vectorial functions.

and the unknowns are

$\mathbf{u} \in [H_D^1(\Omega)]^2$	the velocity profile,
$p \in L^2(\Omega)$	the pressure distribution,

with the same notation as used for the Stokes problem in the previous chapter.

The difference between the Stokes case and the Navier-Stokes one is the introduction of the non-linear term

$$(\mathbf{u} \cdot \nabla) \mathbf{u}. \tag{4.2}$$

For the incorporation of this additional term, an iterative method need to be set into place, but all the numerical background needed for the solution of the Stokes equations will still be required in the following.

Before starting the presentation of the iterative methods, let us recall some notation. As in the previous chapter, the first step towards the numerical solution of the Navier-Stokes equations will be to set them in weak formulation. The bilinear terms are as previously

$$\begin{aligned}
 a(\mathbf{u}, \mathbf{v}) &= \int_{\Omega} \nu \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, d\mathbf{x}, \\
 b(\mathbf{u}, q) &= - \int_{\Omega} q \nabla \cdot \mathbf{u} \, d\mathbf{x}, \\
 (\mathbf{f}, \mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x},
 \end{aligned}$$

with \mathbf{v} and q being test functions in the spaces $V = [H_0^1(\Omega)]^2$ and $Q = L_0^2(\Omega)$. For treating the term in Equation (4.2), the introduced notation is

$$n(\mathbf{w}, \mathbf{u}, \mathbf{v}) = \int_{\Omega} ((\mathbf{w} \cdot \nabla) \mathbf{u}) \cdot \mathbf{v} \, d\mathbf{x}. \quad (4.3)$$

Using this notation, it is possible to write the non-linear term as a trilinear term. This change in notation will be useful while applying an iteration process, by setting as first and second term of (4.3) two different approximations of the final solution.

4.2 Picard iterative method for Navier-Stokes equations

The approaches to solve the non-linearity of the new term (4.2) are usually iterative. Here the method known as Picard iteration will be presented.

An iterative method is a method that takes as input a first approximation $(\mathbf{u}^{(n)}, p^{(n)}) \in V \times Q$ of the solution $(\mathbf{u}, p) \in V \times Q$ and builds a new approximation $(\mathbf{u}^{(n+1)}, p^{(n+1)}) \in V \times Q$, such that, in principle,

$$\left\| (\mathbf{u}^{(n)}, p^{(n)}) - (\mathbf{u}, p) \right\|_{V \times Q} \rightarrow 0 \quad \text{for } n \rightarrow \infty.$$

The formulation for a step of a generic iterative method for our case is

$$\begin{pmatrix} \mathbf{u}^{(n+1)} \\ p^{(n+1)} \end{pmatrix} = \begin{pmatrix} \mathbf{u}^{(n)} \\ p^{(n)} \end{pmatrix} + L^{-1} \left(r \left(\mathbf{u}^{(n)}, p^{(n)} \right) \right),$$

where L is a linear functional to be determined depending on the method used and $r(\mathbf{u}^{(n)}, p^{(n)})$ is the weak residual of the method at step n . For the Navier-Stokes equations, it holds

$$\left(r(\mathbf{u}, p), \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix} \right) = \left(N(\mathbf{u}, \mathbf{u}, p), \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix} \right) - \left(\begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{v} \\ q \end{pmatrix} \right),$$

where N is defined as

$$N(\mathbf{w}, \mathbf{u}, p) = \begin{pmatrix} a(\mathbf{u}, \mathbf{v}) + n(\mathbf{w}, \mathbf{u}, p) + b(\mathbf{v}, p) \\ b(\mathbf{u}, q) \end{pmatrix}.$$

Different iterative methods might have different definitions of the linear operator L , that is generally based on the functional N for its definition.

In the following, the method known as Picard iterative method will be presented for the case of the Navier-Stokes equations.

In Picard iteration the linear operator is defined as

$$L = N \left(\mathbf{u}^{(n)}, \mathbf{u}^{(n+1)}, p^{(n+1)} \right),$$

and the complete system to solve becomes

find $(\mathbf{u}^{(n+1)}, p^{(n+1)}) \in V \times Q$

$$\begin{pmatrix} a(\mathbf{u}^{(n+1)}, \mathbf{v}) + n(\mathbf{u}^{(n)}, \mathbf{u}^{(n+1)}, \mathbf{v}) + b(\mathbf{v}, p^{(n+1)}) \\ b(\mathbf{u}^{(n+1)}, q) \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}, \quad \forall (\mathbf{v}, q) \in V \times Q \quad (4.4)$$

In this formulation, it is possible to see why the implementation of the Stokes equations is a useful first step towards the discretization of the Navier-Stokes equations, since most of the terms in Equation (4.4) were already presented in the previous chapter. The only difference is the transposition of the non-linear term, that now reads

$$n \left(\mathbf{u}^{(n)}, \mathbf{u}^{(n+1)}, \mathbf{v} \right) = \left(\left(\mathbf{u}^{(n)} \cdot \nabla \right) \mathbf{u}^{(n+1)}, \mathbf{v} \right).$$

By following the same transformations as in Section 3.7, it is possible to go from Equation (4.4) to a linear system of the form

$$\begin{bmatrix} A + N^{(n)} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}^{(n+1)} \\ \mathbf{P}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}, \quad (4.5)$$

where the introduced matrix $N^{(n)}$ depends on the solution of the previous iteration, such that

$$N_{ij}^{(n)} = \left(\left(\mathbf{u}_h^{(n)} \cdot \nabla \right) \phi_j, \phi_i \right).$$

In the notation used, $\mathbf{u}_h^{(n)}$ is to be interpreted as the numerical solution to the previous iterative step.

Since this method is iterative, an approximation of the solution need to be chosen for the step 0. In our code, the choice was to set

$$\left(\mathbf{u}^{(0)}, p^{(0)} \right) = (\mathbf{0}, 0),$$

that is equivalent to set the first iterate $(\mathbf{u}^{(n)}, p^{(n)})$ to be the numerical solution to the Stokes problem.

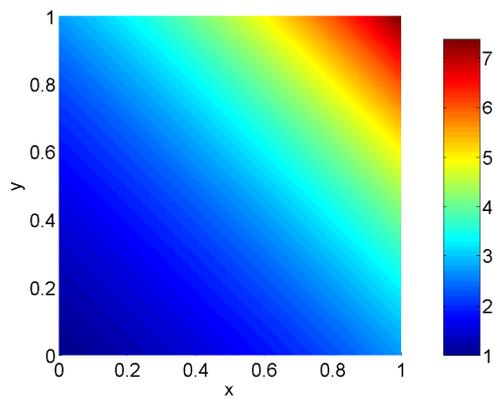
4.3 Numerical tests in the unit square

In the following, some tests in the unit square will be presented, in particular, a case with exponential solution will be treated.

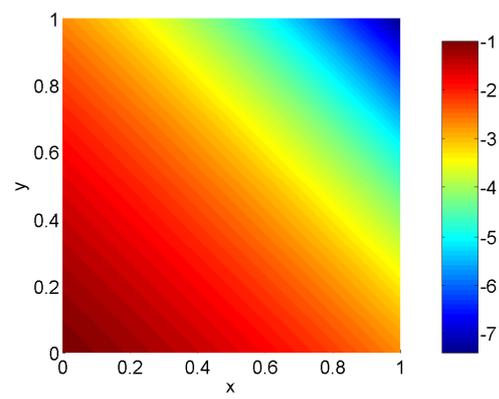
In the case of Navier-Stokes equations, the Brezzi inf-sup constant is constant along the iterations of the Picard method, since it just depends on the matrix B in Equation (4.5), while Babuška inf-sup constant changes at each iteration of the method, since it depends on the full left-hand side

$$\begin{bmatrix} A + N^{(n)} & B^T \\ B & 0 \end{bmatrix}.$$

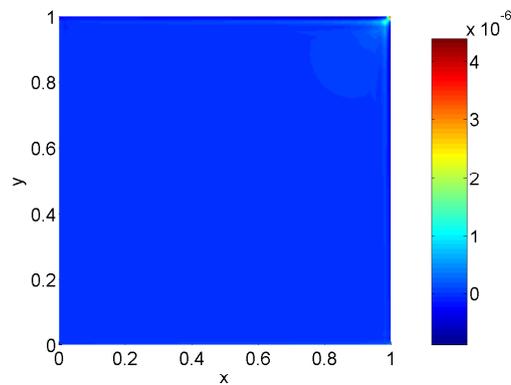
Moreover, given the discretization of the physical domain, Brezzi inf-sup constant is equal for both the Stokes and Navier-Stokes problems. It will therefore not be presented in this chapter while dealing with the discretization of the unit square. On the other hand, Babuška inf-sup constant can be studied in dependence of the iteration, and is therefore interesting for our study.



(a) First component of the velocity u_1 for the exponential flux



(b) Second component of the velocity u_2 for the exponential flux



(c) Pressure p for the exponential flux

Figure 4.1: Numerical solution of the exponential flux, two components of velocity and the pressure

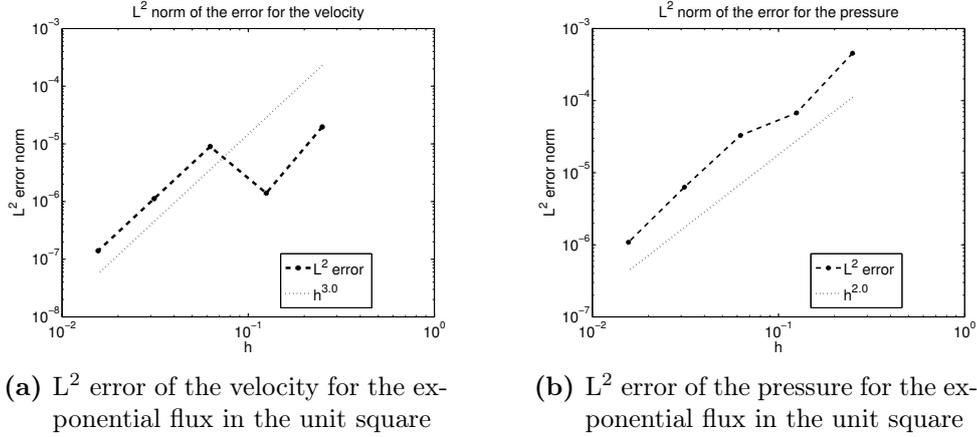


Figure 4.2: Errors for the velocity and pressure for the exponential flux the Taylor-Hood spaces with $p = 1$ and $r = 1$

4.3.1 Exponential flux

As first case, the following problem is solved. Non-homogeneous Dirichlet boundary conditions are imposed on all the boundary of the unit square, ν is set to 1 and the left-hand side is set such that the analytical solution of the Navier-Stokes equations is

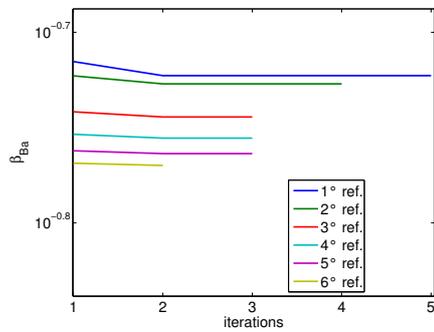
$$\begin{cases} \mathbf{u}(x, y) = \begin{pmatrix} \exp(x + y) \\ -\exp(x + y) \end{pmatrix}, \\ p(x, y) = 0. \end{cases}$$

This brings the Reynolds number to be $\exp(2)$. The numerical solution is displayed in Figure 4.1. The convergence of the error along refinements of the grid is plotted in Figure 4.2. In this case, it is possible to see how the order of convergence of the velocity and of the pressure coincide, being in both cases 4.5. The results show even better convergence with respect to the Stokes case for the vortex flow.

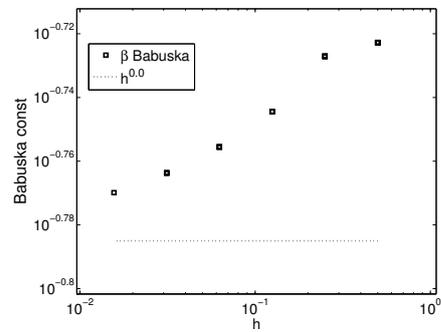
As previously introduced, the possibility of studying both the Babuška inf-sup constant along iterations is interesting. The results of these tests are presented in Figure 4.3. In this particular case of a uniform grid on the unit square, the two values displayed do not change drastically along the iterations, there is just a decrease in the Babuška inf-sup condition during the first iterative step, but afterwards both values are constant. Another feature that the graphs show is the gradual decrease of the number of iterations needed for the method to converge while the mesh is refined.

h	iteration 1	iteration 2	iteration 3	iteration 4	iteration 5
0.5000	0.192593	0.189346	0.189345	0.189345	0.189345
0.2500	0.189296	0.187468	0.187468	0.187468	
0.1250	0.181249	0.180113	0.180113		
0.0625	0.176379	0.175576	0.175576		
0.0313	0.172912	0.172315	0.172315		
0.0156	0.170334	0.169875			

Table 4.1: Babuška inf-sup constants for Navier-Stokes equations in the unit square for the case of the exponential solution with Taylor-Hood spaces with $p = 1$ and $r = 1$



(a) Babuška inf-sup constant for the exponential flux vs. iterations



(b) Babuška inf-sup constant for the exponential flux vs. refinements

Figure 4.3: Babuška inf-sup constant along iterations and refinements for the exponential flux with Taylor-Hood spaces with $p = 1$ and $r = 1$

4.4 The benchmark problem

The last results presented will deal with the benchmark problem of a flow around a cylinder. Such geometry has already been introduced in Section 2.6 and we will be here using the same divisions in patches are presented in Figure 19.

Given the geometry, Dirichlet boundary conditions are imposed in all the domain as follows

$$\begin{aligned} \mathbf{u}(-0.2, y) = \mathbf{u}(2.0, y) &= 0.41^{-2} \begin{pmatrix} 1.2(y + 0.2)(0.21 - y), \\ 0 \end{pmatrix} & -0.2 \leq y \leq 0.21, \\ \mathbf{u}(x, -0.2) = \mathbf{u}(x, 0.21) &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} & -0.2 \leq x \leq 2.0 \\ \mathbf{u}(x, y) &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} & x^2 + y^2 = 0.05^2. \end{aligned}$$

This correspond to have a parabolic inflow and outflow profile at $x = -0.2$ and $x = 2.0$, while imposing zero shear around the cylinder and for $y = -0.2$ and $y = 0.21$. The left-hand side is set to zero on all the domain. Setting $Re = 20$, the kinematic viscosity becomes 10^{-3} .

Let us start at first with a visual confrontation of the solutions recovered using the different discretizations proposed in Section 2.6. Solutions with similar number of degrees of freedom have been chosen, low enough to give an intuitive idea of how deeply the choice of the division in patches can affect the numerical solution. Results with number of degrees of freedom around 500 are presented in Figures 4.4 and 4.5, for the single-patch domain and the domain splitted along $y = 0$.

4.4.1 Babuška inf-sup constant vs iteration number for different geometries

In the case considered, the benchmark problem has been solved using different discretizations of the domain, that brought to have different Babuška inf-sup constants. It is in particular interesting to consider the behavior of the Babuška inf-sup constant vs. the number of iterations.

The results for this computations are plotted in Figure 4.6, where a confrontation between the one patch case and the domain divided along $y = 0$ is established. It can be seen that in both cases there are some oscillations in the first iterations of the method, that can probably be explained as the result of introducing a non-physical solution into the iterative method. But soon the inf-sup constant acquire a stable behavior, and the final values are similar to the value recovered for the first iteration.

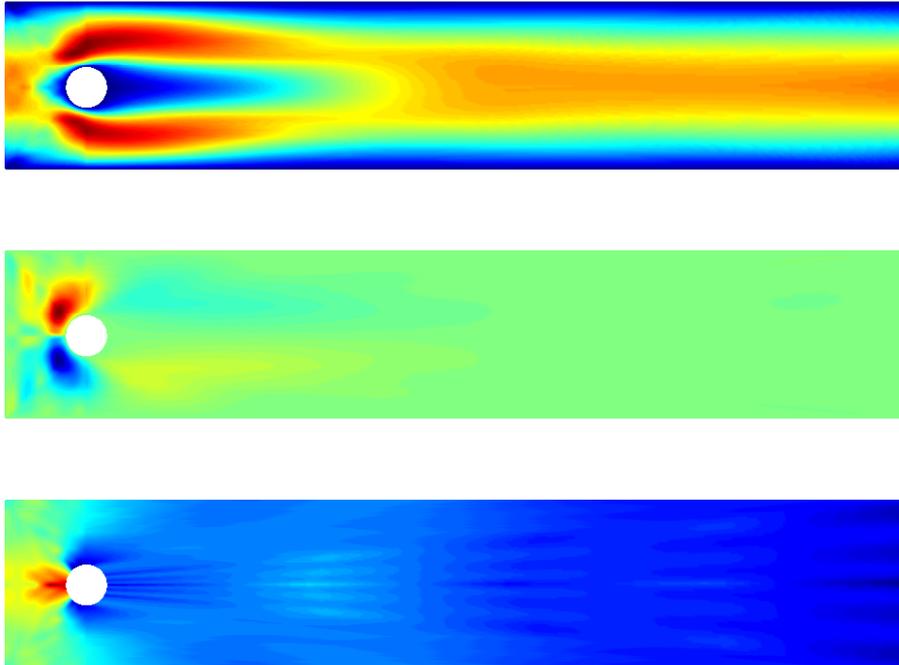


Figure 4.4: Numerical results for the single-patch domain with 421 DoF

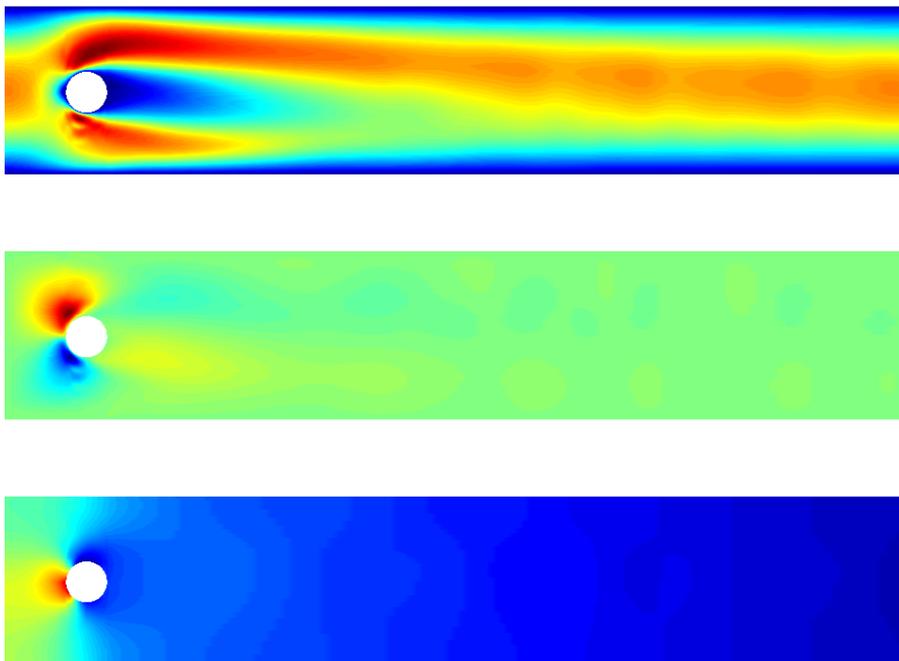
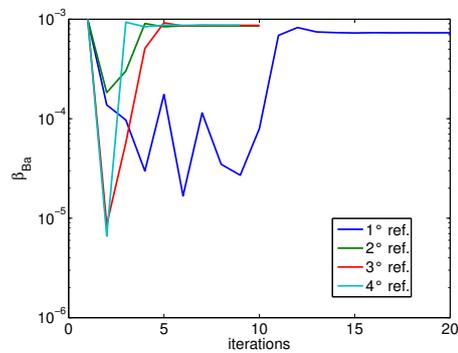
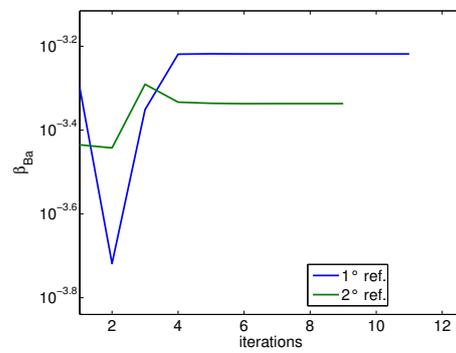


Figure 4.5: Numerical results for the domain divided along $y = 0$ with 756 DoF



(a) Babuška inf-sup constant for the single patch domain



(b) Babuška inf-sup constant for the domain divided along $y = 0$

Figure 4.6: Babuška inf-sup constant and condition number for the Navier-Stokes problem in the benchmark problem

Conclusions and outlook

In the field of the numerical solution of Partial Differential Equations, the method called Isogeometric Analysis is lately acquiring more importance, since the results have proven it to be a reliable and numerically accurate method. In particular, the capability of this method to discretize exactly curved geometries has a strong appeal in applications. If on one hand the available tests for Isogeometric Analysis are promising, on the other hand the theory and practice are not yet completely developed as the one of the Finite Element Method. For this reason, in recent years, studies have been proposed dealing with the stability of the numerical spaces used in Isogeometric Analysis for saddle point problems. This issue has already been solved for the Finite Element Method, where pairs of stable spaces are known and have given good results. On the other hand, in Isogeometric Analysis an extensive analysis in this sense is still missing.

In this project, the focus has been put on testing the fulfillment of the discrete inf-sup condition in the Stokes and Navier-Stokes problem for Taylor-Hood coupled spaces. In particular, the dependence of the Brezzi and Babuška inf-sup constants on the polynomial degree used for the solution and the parametrization of the domain are investigated.

The numerical tests presented suggest that the choice of the polynomial degree and of the degree of continuity has not a big impact on the discrete inf-sup constant. On the other hand, different discretizations of the same geometry might lead to strong differences in both the inf-sup constants of the linear system, as underlined by the results presented for the benchmark problem with multiple patches. This result is interesting, since it stresses the importance of the choice of the parametrization. Indeed, it means that also for Isogeometric Analysis the properties of the geometrical representation determine the quality of the solution, as is the case for the Finite Element Method, even if independently of geometrical approximations. Moreover, the Babuška inf-sup constant changes along the iterations of the Picard method used to solve the non-linear term in the Navier-Stokes equations.

Further developments of this research might include a careful investigation of the numerical approximation of the flow in benchmark problems, with particular attention to the computation of the drag and lift coefficients for different parametrization. Other interesting extensions are represented by testing of other inf-sup stable spaces.

Bibliography

- [1] I. Babuška. Error-bounds for finite element method. *Numerische Mathematik*, 16(4):322–333, 1971.
- [2] K.-J. Bathe, D. Hendriana, F. Brezzi, and G. Sangalli. Inf-sup testing of upwind methods. *international Journal for Numerical Methods in Engineering*, 48:745–760, 2000.
- [3] 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.
- [4] 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.
- [5] D. Chapelle and K.-J. Bathe. The inf-sup test. *Computers & Structures*, 47(4):537–545, 1993.
- [6] F. Charru. *Hydrodynamic Instabilities*. Number 37. Cambridge University Press, 2011.
- [7] J.A. Cottrell, T.J. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 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] L. Demkowicz. Babuška \leftrightarrow Brezzi? *ICES Report*, 2006.
- [10] P. G. Drazin and W. H. Reid. *Hydrodynamic Stability*. Cambridge University Press, 2004.
- [11] M. Fortin and F Brezzi. *Mixed and Hybrid Finite Element Methods*. Springer, 1991.
- [12] E. Guyon. *Physical Hydrodynamics*. Oxford University Press, 2001.
- [13] 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):4135–4195, 2005.

- [14] V. John and G. Matthies. Higher-order finite element discretizations in a benchmark problem for incompressible flows. *International Journal for Numerical Methods in Fluids*, 37(8):885–903, 2001.
- [15] F. Negri. Reduced basis method for parametrized optimal control problems governed by PDEs. Master thesis, Politecnico di Milano, 2011.
- [16] L.A. Piegl and W. Tiller. *The NURBS Book*. Springer, 1997.
- [17] S. B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [18] A. Pressan and G. Sangalli. Isogeometric discretizations of the Stokes problem: stability analysis by the macroelement technique. *IMA Journal of Numerical Analysis*, 2012.
- [19] A. Quarteroni. *Numerical Models for Differential Problems*. Springer, 2010.
- [20] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 1994.
- [21] L. Schumacher. Isogeometric analysis for scalar convection-diffusion equations. Diplomarbeit, Humboldt-Universität zu Berlin, 2013.
- [22] H. Si. TetGen. <http://www.tetgen.org>. Accessed: 2014-05-28.