STABILIZED FINITE ELEMENT METHODS FOR THE BRINKMAN EQUATION ON FITTED AND FICITITIOUS DOMAINS

by

Jarle Sogn

Thesis

for the degree of

MASTER OF SCIENCE

Master in applied mathematics and mechanics

Faculty of Mathematics and Natural Sciences

University of Oslo

February, 2014

Acknowledgements

I would like to thank my supervisors Kent-Andre Mardal, André Massing, Øyvind Evju and Martin Sandve Alnæs, who have helped me both with this thesis and further academical prospects. For that I am truly grateful.

I would like to express my gratitude to Magne, Håkon and Tom Andreas for good discussions, Andreas for proofreading most parts of this thesis, and the people in Study hall B1002 for creating a great work environment.

Finally, a thanks goes to my family for being patient with me during last Christmas.

Contents

Chapter 1

Introduction

Many aspects of nature's change can be described through mathematical models. A large part of these models are partial differential equations (PDEs). Most PDEs do not have a closed form analytical solution. To solve these equations we use numerical methods. One of the largest family of numerical methods for solving partial differential equations is the finite element method (FEM).

We are interested in computational fluid dynamics, in particular viscous and porous flow, which arises in many physical models. Stokes flow and Darcy flow describe viscous and porous flow, respectively. The parameter-dependent equation, the Brinkman equation, describes both flows. We will apply finite element methods on the Brinkman equation and treat the Stokes flow and Darcy flow as extreme cases of the Brinkman equation.

Saddle point problems, which the Brinkman equation is, need finite elements that satisfies a set of conditions known as the Brezzi conditions [1]. These elements are referred to as stable elements. Alternatively, stabilization techniques can be used to circumvent these conditions. The *mini*–elements are uniformly stable for Brinkman equation [2, 3]. In Chapter 4 we use the *mini*–elements as a comparison to two different stabilization methods; the Pressure Stabilized Petrov–Galerkin (PSPG) method and the Continuous Interior Penalty (CIP) method.

Finding error estimates for a numerical method is important. They give information on how the error behaves and how good the method is. Juntunen and Stenberg [3] prove an a priori error estimate for the *mini*–elements and the PSPG method. We will prove a similar a priori error estimate for the CIP method. Burman and Hansbo [4] have proved an error estimate for the CIP method. However, they consider other solution spaces.

Normally, finite element methods involve discretizing the domain into a mesh, a fitted mesh. The meshing of a domain can be challenging and time consuming, especially if the domain is complex. An alternative is using fictitious domain methods. These are methods in which the domain and mesh are independent from each other. We will consider a fictitious domain method with cut elements. This method is the similar to that of Burman and Hansbo [5, 6]. Fictitious domain methods create some new challenges, one being that the elements on the boundary need stabilizing, which is done with ghost–penalties [7, 8]. Another challenge is that Dirichlet boundary conditions need to be weakly imposed. To achieve this, we use the Nitsche method [9], which has been popular in recent years because of advances in fictitious domain methods.

The work of this master thesis was done at the Center for Biomedical Computing (CBC) and the initial task was to perform numerical simulations of blood flow through intracranial aneurysms with stents inserted. The extremely fine structure of the stents and the complex domain made normal methods insufficient and we have instead focused on the fictitious domain method. Even though there are many possible applications for the fictitious domain method, the medical application has been the main motivation and is explained in Chapter 2. To perform numerical simulations on large problems (like blood flow through aneurysms), good preconditioners are essential. Chapter 8 contains operator preconditioning for the unstabilized and the stabilized Brinkman equation. The theory of operator preconditioning is based on Mardal and Winther [10, 11]. These articles suggests a preconditioner for the unstabilized Brinkman equation and we suggest our own preconditioner for the stabilized Brinkman equation. Both preconditioners are examined by finding the condition number and the number of iterations needed to solve the system. The operator preconditioning approach also gives a deeper understanding of the nature of the Brinkman equation.

The structure of this thesis is: In Chapter 2 we present motivation and applications for fictitious domain methods in the context of treatment of intracranial aneurysms. The Navier–Stokes equations are derived in Chapter 3, which we discretize with respect to time and obtain the Brinkman equation. In Chapter 4 and 5 we consider the finite element method and the fictitious domain method respectively, with regards to the Brinkman equation. In Chapter 6 we examine the Darcy case and present numerical results for the fictitious domain method. An a priori error estimate is proved in Chapter 7. Chapter 8 contains preconditioning and the discussion can be found in Chapter 9.

Chapter 2

Motivation and application

2.1 Intracranial aneurysms

An intracranial aneurysm (cerebral aneurysm) is an abnormal blood vessel inside the brain which has a balloon-like bulge of the vessel wall. Unruptured intracranial aneurysms occur in 5-8% of the general population [12]. Aneurysms are subject to spontaneous rupture. If this happens, blood bleeds into the subarachnoid space, which is called subarachnoid hemorrhage (SAH). In western countries the annual incidence of aneurysmal SAH is approximately 9 / 100 000. This number varies with several factors, such as country, gender and smoking or non–smoking [13, 14]. 30-40% of those who get aneurismal SAH, die from the initial bleed. 7-20% of those surviving the initial treatment, die within one year due to complications such as acute re-bleeding or vasospasm. Of the survivors approximately 30% sustain significant neurological deficits [15].

An aneurysm ruptures when hemodynamic loads exceed the wall strength. The details of why the wall is weakened and the evolution of the aneurysm towards stabilization or rupture, are not fully understood. However, several risk factors have been studied [12]. The probability that an aneurysm will rupture is dependent on many factors, such as size, location and geometry. Hemodynamic factors are considered to be important for the growth and rupture of aneurysms. A lot of computational studies have and are being done to model the blood with computational fluid dynamics to calculate the wall shear stress (WSS) and pressure on the vessel wall [12, 16, 15]. One of the aims of these studies is to help physicians better determine the risk of rupture and course of treatment. There are three main treatment options for intracranial aneurysms: Surgery, endovascular coiling and endovascular insertion of a stent [15]. The third option is often in combination with the second option. We explain the two endovascular treatment options in the following two paragraphs.

The aneurysm is filled up with coil material using cerebral angiography techniques. This lowers the intra-aneurysmal hemodynamic stress and induces thrombus formation which

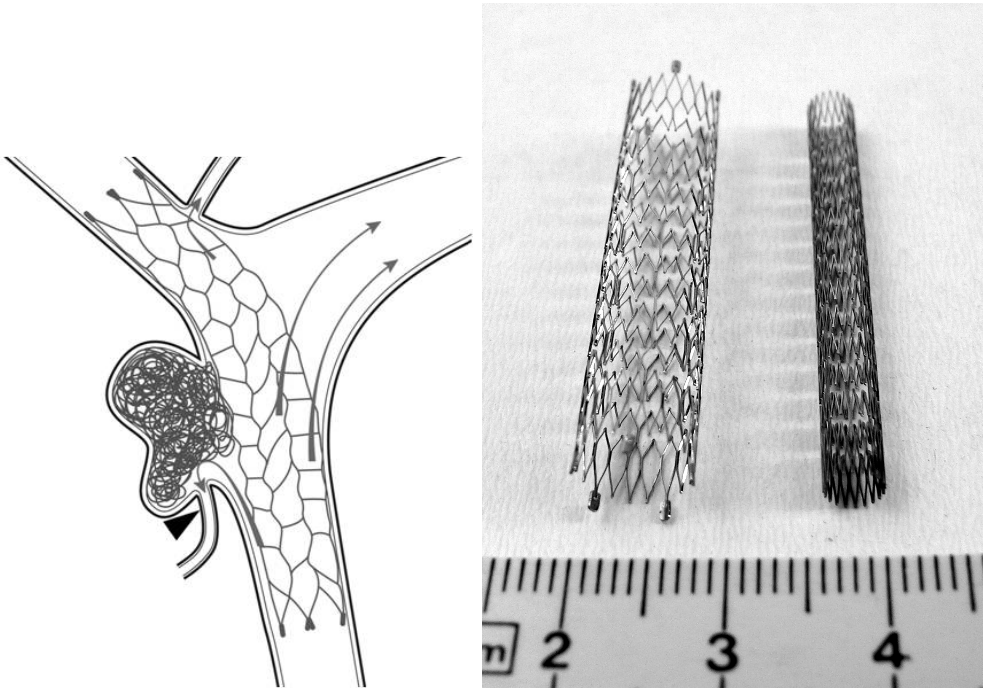

Figure 2.1: The left figure illustrates a wide neck aneurysm which is both coiled and stented. The stent keeps the coil in place. The right figure shows two stents. Source: US National Institutes of Health and Wikimedia Commons.

eventually clods the aneurysm. Endovascular coiling is the most preferred method. Its disadvantages is that the coil can "collapse" and wide neck aneurysms can not be coiled directly.

A stent (see Figure 2.1) is placed in a vessel such that some of the blood is diverted away from the aneurysm sack and WSS and pressure is reduced. There are several different stent designs with different space between the threads. Stents which allow no blood from passing through are called flow diverters. One of the disadvantage, with using stents is that unwanted thrombosis can occur and blood clots can be created. This can be treated with medications.

2.2 Motivation for using methods of non-matching meshes

For our computations our domain is a blood vessel with a stent inserted. Figure 2.2 shows a simple stent design. We would like to change the number of twists and number of stent threads to find an "optimal" patient specific stent and also consider other designs. Since stents have an extremely fine structure (thread thickness is $30 - 50\mu m$ [17]) and we wish to consider many different stents, meshing the domain (blood vessel and several different

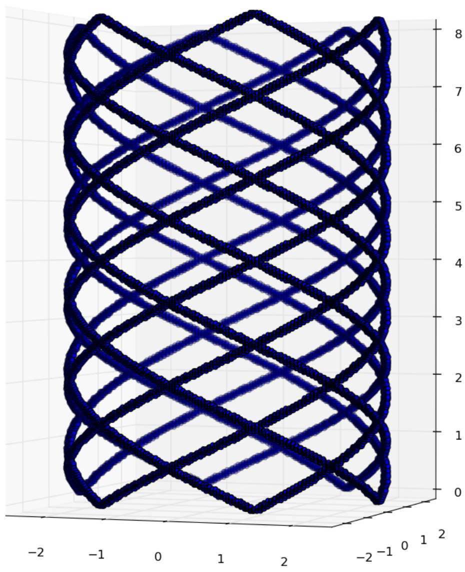

Figure 2.2: Simple stent design. The spacial axis x, y and z axis where discretized with 125, 125 and 250 points, respectively. A program uses an analytical function (level set function) to determines if the point is on/in the stent or not.

stents) is difficult and the cell size must be small in order to represent a stent properly. One solution is to refine the mesh such that the size of cells close to the stent decreases. However, the number of cells will grow with the power of 2–3 and the mesh becomes too large to handle. Therefore we seek a method that can accurately represent the stent without the need for extensive refinement. Non-matching methods (unfitted methods) have this property. The methods we will consider are not well studied and we need to further develop the methodology, which is the focus of this thesis.

In the unfitted methods we will consider, the domain is described through analytical functions (level set functions) and is independent of the mesh. This makes the methods desirable for many other application where the domain is complex and difficult to mesh.

Chapter 3

Mathematical model

3.1 The Navier–Stokes equations

The Navier–Stokes equations are important equations used to describe the motion of fluid. They are derived from conservation of mass, Newton's second law and stress-strain relations. In this section we will derive the incompressible Navier–Stokes equations [18]. We assume that the fluid is incompressible. This is a reasonable assumption if the flow speeds we are considering are low compared to the speed of sound. We need to use the transport theorem.

Theorem 3.1. ***Reynold's transport theorem.***

Let $\Omega(t) \subset \mathbb{R}^3$ be a material volume. Assume $f : \Omega(t) \times [0, T] \to \mathbb{R}$ and that f is smooth. Then the following holds:

$$\frac{\partial}{\partial t} \int\limits_{\Omega(t)} f d\mathbf{x} = \int\limits_{\Omega(t)} \frac{\partial f}{\partial t} d\mathbf{x} + \int\limits_{\partial\Omega(t)} f\, \mathbf{u} \cdot \mathbf{n}\, dS, \tag{3.1}$$

where $\mathbf{n}$ is the normal vector to the surface $\partial\Omega(t)$ and $\mathbf{u}$ is the surface velocity.

Proof. Proof can be found in [18]. □

Let $\Omega(t) \subset \mathbb{R}^3$ be a material volume. The mass of this volume is $\int_{\Omega(t)} \rho(\mathbf{x}, t) d\mathbf{x}$. We have assumed incompressibility, so $\rho(\mathbf{x}, t) = \rho$. From the principal of conservation of mass we get

$$\frac{\partial}{\partial t} \int\limits_{\Omega(t)} \rho\, d\mathbf{x} = 0. \tag{3.2}$$

We apply the transport theorem to the equation above and obtain

$$\rho \int\limits_{\partial\Omega(t)} \mathbf{u} \cdot \mathbf{n}\, dS = 0. \tag{3.3}$$

The next step is to apply the divergence theorem to the equation above, which gives

$$\rho \int\limits_{\Omega(t)} \nabla \cdot \mathbf{u} \, d\mathbf{x} = 0. \tag{3.4}$$

Since the volume $\Omega(t)$ is arbitrary, it follows that

$$\nabla \cdot \mathbf{u} = 0. \tag{3.5}$$

This is one part of the Navier–stokes equations. It states that the divergence of the velocity field is zero everywhere, which follows from conservation of mass and incompressibility.

Next will use Newton's second law. The momentum per unit volume is $\rho\mathbf{u}(\mathbf{x}, t)$. There are two kinds of forces: Body forces and surface forces. The surface forces (per unit volume) can be written as the divergence of the Cauchy stress tensor, which gives us $\nabla \cdot \sigma$. Let $\mathbf{f}$ denote the body force. Newton's second law can now be stated directly as

$$\frac{\partial}{\partial t} \int\limits_{\Omega(t)} \rho\mathbf{u}(\mathbf{x}, t) \, d\mathbf{x} = \int\limits_{\Omega(t)} \rho \, \mathbf{f} \, d\mathbf{x} + \int\limits_{\partial\Omega(t)} \nabla \cdot \sigma \, dS. \tag{3.6}$$

We use the transport theorem on the left-hand side of equation (3.6) to obtain

$$\int\limits_{\Omega(t)} \rho \frac{\partial \mathbf{u}}{\partial t} \, d\mathbf{x} + \int\limits_{\partial\Omega(t)} \rho\mathbf{u}(\mathbf{u} \cdot \mathbf{n}) \, dS = \int\limits_{\Omega(t)} \rho \, \mathbf{f} \, d\mathbf{x} + \int\limits_{\partial\Omega(t)} \nabla \cdot \sigma \, dS. \tag{3.7}$$

Using Green's formula and Equation (3.5) on the surface term on the left-hand side, we end up with

$$\int\limits_{\Omega(t)} \rho \frac{\partial \mathbf{u}}{\partial t} \, d\mathbf{x} + \int\limits_{\Omega(t)} \rho\mathbf{u} \cdot \nabla\mathbf{u} \, d\mathbf{x} = \int\limits_{\Omega(t)} \rho \, \mathbf{f} \, d\mathbf{x} + \int\limits_{\partial\Omega(t)} \nabla \cdot \sigma \, dS. \tag{3.8}$$

Since the volume $\Omega(t)$ is arbitrary, we can remove the integrals and get

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla\mathbf{u} = \mathbf{f} + \frac{1}{\rho} \nabla \cdot \sigma. \tag{3.9}$$

Finally we need to find an expression for the stress tensor σ. We assume that the fluid is isotropic which means that the fluid is uniform in all orientations. If the fluid is at rest, the stress will be $\sigma = -pI$, where p is the thermodynamic pressure and I is the identity matrix. However, we have a moving fluid which will develop additional stress due to viscosity. We use a simple linear stress-strain relation. Fluids that follow such stress-strain behavior are called Newtonian fluids. The stress tensor becomes $\sigma = -pI + 2\mu\varepsilon(\mathbf{u})$, where μ is the dynamic viscosity and $\varepsilon(\mathbf{u})$ is the strain rate tensor. This tensor is symmetric and can be represented as

$$\varepsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T). \tag{3.10}$$

Inserting our expression for the stress tensor σ in Equation (3.9), we obtain the incompressible Navier–Stokes equations

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = \mathbf{f} - \frac{1}{\rho}\nabla p + 2\nu\nabla\cdot\varepsilon(\mathbf{u}), \tag{3.11}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{3.12}$$

3.2 The Brinkman equation

The Brinkman equation describes flow through porous media. It can also be seen as a singular perturbation problem. Here we use it to describe viscous flow by discretizing the Navier–Stokes equations in a certain fashion.

3.2.1 Discretizing the Navier–Stokes equation in time

The divergence of the strain rate tensor (3.10) can be written as $\frac{1}{2}\nabla\cdot(\nabla\mathbf{u} + \nabla\mathbf{u}^T) = \frac{1}{2}(\Delta\mathbf{u} + \nabla(\nabla\cdot\mathbf{u}))$ which reduces to the vector Laplacian since the $\mathbf{u}$ is divergence free. The Navier–Stokes equations are rewritten as

$$\frac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \nu\Delta\mathbf{u} + \frac{1}{\rho}\nabla p = \mathbf{f}, \tag{3.13}$$

$$\nabla\cdot\mathbf{u} = 0. \tag{3.14}$$

First we discretize in time. Let $t \in (0, T]$ and let M be the number of time steps. $\Delta t = \frac{T}{M+1}$ is the time step. We use $\mathbf{u}^n$ to denote the velocity at time $n\Delta t$; that is, $\mathbf{u}^n = \mathbf{u}(\mathbf{x}, n\Delta t)$. Similar notation is used for $\mathbf{f}$ and p. We use an implicit scheme for the time discretization, except for the convective term $\mathbf{u}\cdot\nabla\mathbf{u}$, for which we use an explicit scheme. We get

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n\cdot\nabla\mathbf{u}^n - \nu\Delta\mathbf{u}^{n+1} + \frac{1}{\rho}\nabla p^{n+1} = \mathbf{f}^{n+1}, \tag{3.15}$$

$$\nabla\cdot\mathbf{u}^{n+1} = 0. \tag{3.16}$$

By rearranging equation (3.15), we get

$$\mathbf{u}^{n+1} - \Delta t\nu\Delta\mathbf{u}^{n+1} + \frac{\Delta t}{\rho}\nabla p^{n+1} = \Delta t\left(\mathbf{f}^{n+1} - \mathbf{u}^n\cdot\nabla\mathbf{u}^n\right) + \mathbf{u}^n. \tag{3.17}$$

We simplify the equation by scaling the pressure such that it contains $\frac{\Delta t}{\rho}$. Furthermore we redefine $\mathbf{f}^{n+1}$ such that it contains the whole right-hand side of equation (3.17). We get

$$\mathbf{u}^{n+1} - \Delta t \nu \Delta \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{f}^{n+1}. \tag{3.18}$$

Finally, we define $\epsilon^2 := \Delta t \nu$ which give us the Brinkman equation,

$$\begin{aligned} \mathbf{u} - \epsilon^2 \Delta \mathbf{u} + \nabla p &= \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0. \end{aligned} \tag{3.19}$$

The Brinkman equation (3.19) will be our main focus from now on. We refer to [19], for other suggestions on discretizing the Navier–Stokes equations.

3.2.2 Some properties of the Brinkman equation

For simplicity we use pure Dirichlet boundary conditions for the velocity; that is,

$$\mathbf{u}\left(\mathbf{x}\right) = \mathbf{g}\left(\mathbf{x}\right), \quad \forall \mathbf{x} \in \partial\Omega = \Gamma. \tag{3.20}$$

It follows that the pressure is only determined uniquely up to a constant. To obtain uniqueness we require the additional condition,

$$\int\limits_{\Omega} p \, d\mathbf{x} = 0. \tag{3.21}$$

We use the notation $p \in L_0^2\left(\Omega\right)$, meaning that the pressure is in the L^2-space and satisfies condition (3.21).

The parameter ϵ is a number between zero and one. For $\epsilon = 1$, Equation (3.19) is a Stokes problem with a non-harmful lower order term. The solution spaces are $\mathbf{u} \in \left[H_g^1\left(\Omega\right)\right]^d$ and $p \in L_0^2\left(\Omega\right)$. For $\epsilon = 0$, Equation (3.19) is a mixed Poisson problem (also called Darcy flow) which has two possible solution spaces, one of them is $\mathbf{u} \in \left[L_g^2\left(\Omega\right)\right]^d$ and $p \in H^1\left(\Omega\right) \cap L_0^2\left(\Omega\right)$. The subscript g implies that the function is equal to $g(\mathbf{x})$ on the boundary[1]. We will discuss the solution space in further detail in Chapter 8 and 9. Remark: When $\epsilon = 0$ the boundary condition (3.20) reduces to $\mathbf{u} \cdot \mathbf{n} = \mathbf{g} \cdot \mathbf{n}$.

[1] The space $L_g^2\left(\Omega\right)$ does not make sense since $\mathbf{u}$ does not need to be continuous at the boundary.

Chapter 4

Numerical methods I: Fitted meshes

In this chapter we explain the finite element method and use it to numerically solve the Brinkman equation. We introduce the *mini*–element and some stabilization methods to circumvent the Brezzi conditions. The convergence rates are measured for the different methods. The mesh $\mathcal{T}$ is assumed to be fitted to the domain Ω; that is, $\bigcup_{T\in\mathcal{T}} T = \bar{\Omega}$.

4.1 The finite element method

The finite element method (FEM) is an elegant mathematical framework for solving differential equations. It emerged in the second half of the 20th century and is today a widely used method. The method can be viewed in four steps:

1. A (strong) formulation of the problem
2. A weak formulation of the problem
3. Discretizing with a suitable finite element
4. Solution algorithm

To explain these steps we use an example problem. The strong formulation of the example problem reads:

Problem 4.1. ***Poisson's equation****:*

Find a function $u(\mathbf{x})$ in some function space $V(\Omega)$ such that

$$\begin{aligned} -\Delta u &= f(\mathbf{x}) \quad \forall\, \mathbf{x} \in \Omega, \\ u(\mathbf{x}) &= g(\mathbf{x}) \quad \forall\, \mathbf{x} \in \Gamma_D, \\ \partial_{\mathbf{n}} u(\mathbf{x}) &= h(\mathbf{x}) \quad \forall\, \mathbf{x} \in \Gamma_N, \end{aligned} \tag{4.1}$$

where $\partial_{\mathbf{n}} u(\mathbf{x}) = \nabla u(\mathbf{x}) \cdot \mathbf{n}$ and Ω is the domain. The boundary is $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N$, where Γ_D denotes the Dirichlet part and Γ_N denotes the Neumann part. We assume that they do not overlap. If Γ_D is nonempty and $f \in L^2(\Omega)$, then the problem is well-posed.

4.1.1 The weak formulation

Problem 4.1 is strongly formulated. We seek to reformulate it as a weak formulation. Let v be a function in some function space $\hat{V}$. We call v the test function. We multiply the test function v with the first line of Equation (4.1) and integrate over the domain,

$$\int_{\Omega} -\Delta u \, v \, d\mathbf{x} = \int_{\Omega} f \, v \, d\mathbf{x}. \tag{4.2}$$

The left-hand side is integrated by parts,

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{x} - \int_{\Gamma_D} \frac{\partial u}{\partial \mathbf{n}} v \, dS - \int_{\Gamma_N} \frac{\partial u}{\partial \mathbf{n}} v \, dS = \int_{\Omega} f \, v \, d\mathbf{x}. \tag{4.3}$$

We seek a solution that is equal to $g(\mathbf{x})$ on the Dirichlet boundary, $u|_{\Gamma_D} = g(\mathbf{x})$. We let the test function be zero on the Dirichlet boundary, $v|_{\Gamma_D} = 0$. Equation (4.3) becomes

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{x} = \int_{\Omega} f \, v \, d\mathbf{x} + \int_{\partial\Omega_N} h \, v \, dS. \tag{4.4}$$

By writing the differential equation in this form, we no longer require u to be two times differentiable. Instead, we require that u is in the Sobolev space $H^1_{g,D}(\Omega)$ (see A.3). That is, $u \in V = H^1_{g,D}(\Omega)$, where the subscript, "$g{,}D$", means that u is equal to $g(\mathbf{x})$ on the Dirichlet boundary. A proper space for the test function, is $v \in V = H^1_{0,D}(\Omega)$. We now state the weak formulation of Problem 4.1:

Problem 4.2. *Find $u \in H^1_{g,D}(\Omega)$ such that*

$$\int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{x} = \int_{\Omega} f \, v \, d\mathbf{x} + \int_{\partial\Omega_N} h \, v \, dS \quad \forall \, v \in H^1_{0,D}(\Omega). \tag{4.5}$$

4.1.2 The finite element

We use the definition of the finite element and the nodal basis as defined by Ciarlet in 1978 [20] [21].

Definition 4.1. ***Finite element.***

A finite element is defined by a triple $(T, \mathcal{V}, \mathcal{L})$, where

- *the domain T is a bounded, closed subset of $\mathbb{R}^d$ (for $d = 1, 2, 3, \ldots$) with nonempty interior and piecewise smooth boundary;*
- *the space $\mathcal{V} = \mathcal{V}(T)$ is a finite dimensional function space on T of dimension n;*
- *the set of degrees of freedom (nodes) $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$ is a basis for the dual space $\mathcal{V}'$; that is, the space of bounded linear functionals on $\mathcal{V}$.*

Definition 4.2. ***Nodal basis.***

The nodal basis $\{\phi_i\}_{i=1}^n$ for a finite element $(T, \mathcal{V}, \mathcal{L})$ is the unique basis satisfying

$$\ell_i(\phi_j) = \delta_{ij}.$$

There are many kinds of finite elements which have different properties that are suited for different kinds of problems. One of the simplest and possibly the most common finite element, is the *Lagrange* element. It is also known as the continuous Galerkin element, CG. Let P_q denote the Lagrange element, where q is the polynomial order of the element. Figure 4.1 shows a Lagrange element for $q = 1$, on a interval, triangle and tetrahedron.

Definition 4.3. ***The* Lagrange *element.***

The Lagrange *element (P_q) is defined for $q = 1, 2, \ldots$ by*

$$T = \{\textit{interval, triangle, tetrahedon}\}, \tag{4.6}$$

$$\mathcal{V} = \mathcal{P}_q(T), \tag{4.7}$$

$$\ell_i(v) = v\left(\mathbf{x}^i\right), \quad i = 1, \ldots, n(q), \tag{4.8}$$

where $\{\mathbf{x}^i\}_{i=1}^{n(q)}$ is an enumeration of points in T defined by

$$x = \begin{cases} i/q, & 0 \leqslant i \leqslant q, & T \textit{ interval}, \\ (i/q, j/q), & 0 \leqslant i + j \leqslant q, & T \textit{ triangle}, \\ (i/q, j/q, k/q), & 0 \leqslant i + j + k \leqslant q, & T \textit{ tetrahedron}. \end{cases} \tag{4.9}$$

We also define the *bubble* element B_q, which we will use later for the *mini*–element.

Definition 4.4. ***The* bubble *element.***

The bubble *element (B_q) is defined for $q \geq (d + 1)$ by*

$$T = \{\textit{interval, triangle, tetrahedon}\}, \tag{4.10}$$

$$\mathcal{V} = \{\mathcal{P}_q(T) : v|_{\partial T} = 0\}, \tag{4.11}$$

$$\ell_i(v) = v\left(\mathbf{x}^i\right), \quad i = 1, \ldots, n(q), \tag{4.12}$$

where $\{\mathbf{x}^i\}_{i=1}^{n(q)}$ is an enumeration of points in T defined in such a way that $v\left(\mathbf{x}^i\right)$ for $i = 1, \ldots, n(q)$ is a basis for the dual space of $\mathcal{V}$.

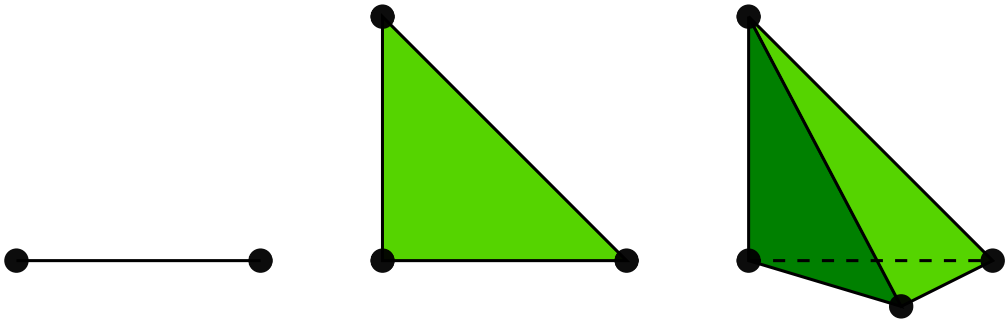

Figure 4.1: The linear Lagrange interval, triangle and tetrahedron. The black dots are the degrees of freedom.

For the reference triangle and tetrahedron, the *bubble* function is $27xy(1-x-y)$ and $256xy(1-x-y-z)$, respectively.

4.1.3 Discretizing the weak formulation

Now we have defined the finite element, the *Lagrange* element and the *bubble* element. We continue on our test Problem (4.1), by discretizing Equation (4.5). Assume we have chosen a finite element. We discretize the domain such that $\bar{\Omega} = \bigcup_{i=1}^{N} T_i$. We seek a function $u_h = V_{h,g,D}\left(\mathcal{T}\right)$ such that

$$\int\limits_{\mathcal{T}} \nabla u_h \cdot \nabla v_h \, d\mathbf{x} = \int\limits_{\mathcal{T}} f \, v_h \, d\mathbf{x} + \int\limits_{\partial\mathcal{T}_N} h \, v_h \, dS \quad \forall \, v_h \in V_{h,0,D}\left(\mathcal{T}\right), \tag{4.13}$$

where $V_h\left(\mathcal{T}\right)$ is a finite dimensional function space. The discrete function u_h, can be written as a linear combination of the basis functions, $\{\phi\}_{j=1}^{N}$, of V_h;

$$u_h = \sum_{j=1}^{N} U_j \phi_j \tag{4.14}$$

Without loss of generality we set $v_h = \phi_i$, for $i = 1, 2, \ldots, N$. Inserting this, Equation (4.13) can be written as

$$\int\limits_{\mathcal{T}} \nabla \sum_{j=1}^{N} U_j \phi_j \cdot \nabla \sum_{i=1}^{N} \phi_i \, d\mathbf{x} = \int\limits_{\mathcal{T}} f \sum_{i=1}^{N} \phi_i \, d\mathbf{x} + \int\limits_{\partial\mathcal{T}_N} h \sum_{i=1}^{N} \phi_i \, dS, \tag{4.15}$$

$$\sum_{j=1}^{N} \sum_{i=1}^{N} U_j \int\limits_{\mathcal{T}} \nabla \phi_j \cdot \nabla \phi_i \, d\mathbf{x} = \sum_{i=1}^{N} \int\limits_{\mathcal{T}} f \, \phi_i \, d\mathbf{x} + \sum_{i=1}^{N} \int\limits_{\partial\mathcal{T}_N} h \, \phi_i \, dS. \tag{4.16}$$

We transform Equation (4.16) to linear system of equations,

$$A\mathbf{U} = B, \tag{4.17}$$

where $\mathbf{U}$ is a vector containing U_j, for $j = 1, 2, \ldots, N$, A is a $N \times N$ matrix,

$$A_{j,i} = \int\limits_{\mathcal{T}} \nabla\phi_j \cdot \nabla\phi_i \, d\mathbf{x}, \tag{4.18}$$

and B is a vector of length N,

$$B_i = \int\limits_{\mathcal{T}} f\,\phi_i \, d\mathbf{x} + \int\limits_{\partial\mathcal{T}_N} h\,\phi_i \, dS. \tag{4.19}$$

The example Problem (4.1) has been transformed from a PDE to a linear system of Equations (4.17). After finding the matrix entries $A_{j,i}$ and the load vector B, Equation (4.17) is solved. This can be done either with a direct solver or with an iterative method with a preconditioner. We will discuss this further in Chapter 8.

4.1.4 Abstract formulation

The finite element method can be summarized in this abstract formulation:

1. Find $u \in V$ such that
$$Lu = f, \tag{4.20}$$
where V is a function space, $L : V \to V^{'}$ is a spatial differential operator and $f \in V^{'}$, where $V^{'}$ is the dual space of V.

2. Introduce a suitable test space $\hat{V}$ and define a bilinear form, $a(\cdot,\cdot)$, and a linear form, $l(\cdot)$.

   Find $u \in V$ such that
$$a\,(u, v) = l\,(v) \quad \forall\, v \in \hat{V}. \tag{4.21}$$

3. Chose a suitable finite element and discretize the domain.

   Find $u_h \in V_h$ such that
$$a\,(u_h, v_h) = l\,(v_h) \quad \forall\, v_h \in \hat{V}_h. \tag{4.22}$$

4. Find an algorithm to find the degrees of freedom of u_h.

4.2 Weak formulation of the Brinkman equation

Let us first restate the Brinkman Equation (3.19).

Problem 4.3. *Find* $\mathbf{u} \in V_g$ *and* $p \in Q$ *such that*

$$\begin{aligned} \mathbf{u} - \epsilon^2 \Delta \mathbf{u} + \nabla p &= \mathbf{f}, \qquad \forall \mathbf{x} \in \Omega, \\ \nabla \cdot \mathbf{u} &= 0, \qquad \forall \mathbf{x} \in \Omega, \\ \mathbf{u} &= \mathbf{g}, \qquad \forall \mathbf{x} \in \Gamma, \end{aligned} \tag{4.23}$$

holds for all $\epsilon \in [0, 1]$*, where* V_g *and* Q *are the solution spaces.*

To obtain a weak formulation of Equation (4.23), we multiply the first line with a test function $\mathbf{v} \in \hat{V_0}$, where $\mathbf{v} = \mathbf{0}$ on the boundary. Then we integrate over the domain,

$$\int\limits_{\Omega} \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} + \int\limits_{\Omega} -\epsilon^2 \Delta \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} + \int\limits_{\Omega} \nabla p \cdot \mathbf{v} \, d\mathbf{x} = \int\limits_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}. \tag{4.24}$$

Now we use integration by parts on two of the terms on the left-hand side,

$$\int\limits_{\Omega} \mathbf{u} \cdot \mathbf{v} \, d\mathbf{x} + \int\limits_{\Omega} \epsilon^2 \nabla \mathbf{u} : \nabla \mathbf{v} \, d\mathbf{x} - \int\limits_{\Omega} p \left(\nabla \cdot \mathbf{v} \right) \, d\mathbf{x} = \int\limits_{\Omega} \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x}. \tag{4.25}$$

The surface integrals vanish since $\mathbf{v} = \mathbf{0}$ on the surface. The divergence term in Equation (4.23) is multiplied with another test function $q \in \hat{Q}$ and integrated over the domain,

$$\int\limits_{\Omega} \left(\nabla \cdot \mathbf{u} \right) \, q \, d\mathbf{x} = 0. \tag{4.26}$$

This integral is subtracted from from Equation (4.25). From now on, we use a simplified notation; $(\cdot, \cdot)_\Omega$ and $(\cdot, \cdot)_\Gamma$ will denote the L^2 inner product over the domain and on the boundary respectively. The weak formulation of the Brinkman Equation (4.23) becomes:

Problem 4.4. *Find* $(\mathbf{u}, p) \in V_g \times Q$ *such that*

$$a \left(\mathbf{u}, \mathbf{v} \right) + b \left(p, \mathbf{v} \right) + b \left(q, \mathbf{u} \right) = \left(\mathbf{f}, \mathbf{v} \right)_\Omega \quad \forall \left(\mathbf{v}, q \right) \in \hat{V_0} \times \hat{Q}, \tag{4.27}$$

where

$$a \left(\mathbf{u}, \mathbf{v} \right) = \left(\mathbf{u}, \mathbf{v} \right)_\Omega + \epsilon^2 \left(\nabla \mathbf{u}, \nabla \mathbf{v} \right)_\Omega , \tag{4.28}$$

$$b \left(p, \mathbf{v} \right) = - \left(p, \nabla \cdot \mathbf{v} \right)_\Omega . \tag{4.29}$$

The saddle point Problem 4.4 is well-posed for $V_g \times Q = \left[H^1_g \left(\Omega \right) \right]^d \times L^2_0 \left(\Omega \right)$ for $\epsilon > 0$ and for the case $\epsilon = 0$, it is well-posed for the spaces $\left[L^2_g \left(\Omega \right) \right]^d \times H^1 \left(\Omega \right) \cap L^2_0 \left(\Omega \right)$ and $H \left(div; \Omega \right) \times L^2_0 \left(\Omega \right)$, assuming $\mathbf{f}$ is in a suitable dual space [2].

4.3 Stabilization of the Brinkman equation

For saddle point problems of the form (4.27) – (4.29), the following four conditions (Brezzi's conditions) have to be satisfied to have a well-posed discrete problem [1]:

$$a(\mathbf{u}_h, \mathbf{v}_h) \leq C_1 \|\mathbf{u}_h\|_{V_h} \|\mathbf{v}_h\|_{V_h}, \quad \forall\, \mathbf{u}_h,\, \mathbf{v}_h \in V_h, \tag{4.30}$$

$$b(\mathbf{u}_h, q_h) \leq C_2 \|\mathbf{u}_h\|_{V_h} \|q_h\|_{Q_h}, \quad \forall\, \mathbf{u}_h \in V_h,\, q_h \in Q_h, \tag{4.31}$$

$$a(\mathbf{u}_h, \mathbf{u}_h) \geq C_3 \|\mathbf{u}_h\|^2_{V_h}, \quad \forall\, \mathbf{u}_h \in V_h, \tag{4.32}$$

$$\sup_{\mathbf{u}_h \in V_h} \frac{b(\mathbf{u}_h, q_h)}{\|\mathbf{u}_h\|_{V_h}} \geq C_4 \|q_h\|_{Q_h}, \quad \forall\, q_h \in Q_h. \tag{4.33}$$

The first two conditions are boundedness of a and b, which can be shown using Cauchy–Schwarz's inequality (A.9). The third condition is coercivity of a, this is only need for the kernel of Q_h. However, we show this for the Nitsche method in Section 7.2 for V_h. The last condition is known as the Babuška–Brezzi condition (also called the inf-sup condition) and can be rewritten as

$$C_4 \leq \inf_{p_h \in Q_h} \sup_{\mathbf{v}_h \in V_{h,g}} \frac{b(\mathbf{v}_h, p_h)}{\|\mathbf{v}_h\|_{V_h} \|p_h\|_{Q_h}}. \tag{4.34}$$

We need to use elements that satisfies (4.30)–(4.33) . Lagrangian elements of same order do not satisfy condition (4.33), nor do the P_1–P_0 elements [22]. This is a problem since the software we use for the unfitted meshes, only have P_0 and P_1 elements implemented. We use the P_1–P_1 elements and need to circumvent the Brezzi conditions. This is done by two different stabilization techniques: The pressure stabilized Petrov–Galerkin method (PSPG) and the continuous interior penalty method (CIP). The discrete stabilized formulation of the Brinkman Problem 4.4 is:

Problem 4.5. *Find* $(\mathbf{u}_h, p_h) \in V_{g,h} \times Q_h$ *such that*

$$\begin{aligned} &a(\mathbf{u}_h, \mathbf{v}_h) + b(p_h, \mathbf{v}_h) + b(q_h, \mathbf{u}_h) + c(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h) \\ &= (f, \mathbf{v}_h)_\Omega + \Phi(\mathbf{v}_h, q_h,), \qquad \forall\, (\mathbf{v}_h, q_h) \in \hat{V}_{0,h} \times \hat{Q}_h, \end{aligned} \tag{4.35}$$

where

$$a(\mathbf{u}_h, \mathbf{v}_h) = (\mathbf{u}_h, \mathbf{v}_h)_\Omega + \epsilon^2 (\nabla \mathbf{u}_h, \nabla \mathbf{v}_h)_\Omega, \tag{4.36}$$

$$b(p_h, \mathbf{v}_h) = -(p_h, \nabla \cdot \mathbf{v}_h)_\Omega, \tag{4.37}$$

where $c(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h)$ *and* $\Phi(\mathbf{v}_h, q_h,)$ *are stabilization terms which depend on the stabilization technique we use.*

Note that the problem is no longer on the form (4.27) – (4.29) and the conditions for discrete well-posedness have also changed.

4.3.1 The *mini*–element

The stabilization techniques will be compared toward a reference element that satisfies the Brezzi conditions. We have chosen to use the *mini*–element [23]. This element consists of P_1 elements with a cubic bubble B_3 for the velocity and P_1 elements for the pressure. The reason for choosing the *mini*–element instead of for example the Crouzeix–Raviart element, is that the *mini*–element is stable both for the Stokes case ($\epsilon = 1$) and the Darcy case ($\epsilon = 0$), where as the solution does not converge when using the Crouzeix–Raviart element for the Darcy case [2]. No stabilization is needed for the *mini*–element, thus $c\left(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h\right)$ and $\Phi\left(\mathbf{v}_h, q_h,\right)$ are zero.

4.3.2 Pressure stabilized Petrov–Galerkin method

The first stabilization method we consider is the "classical" pressure stabilized Petrov–Galerkin method (PSPG). It was introduced by Hughes [24] in 1986 (for the Stokes problem). In [3] the method is applied for the Brinkman equation. The idea is to subtract the residual from the left-hand side of Equation (4.35):

$$-\beta_s \sum_{T\in\mathcal{T}} \frac{h_T^2}{\epsilon^2 + h_T^2} \left(\mathbf{u}_h - \epsilon^2\Delta\mathbf{u}_h + \nabla p_h - \mathbf{f}, \mathbf{v}_h - \epsilon^2\Delta\mathbf{v}_h + \nabla q_h\right)_T, \tag{4.38}$$

where β_s is a stability constant and h_T is the size of the element T, defined as the distance from the interior node to the boundary. For the P_1 element we have

$$\Delta\mathbf{u}_h|_T = 0 \quad \text{and} \quad \Delta\mathbf{v}_h|_T = 0.$$

After multiplying out Equation (4.38), we get

$$\begin{aligned} c\left(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h\right) = -\beta_s \sum_{T\in\mathcal{T}} \frac{h_T^2}{\epsilon^2 + h_T^2} [\,(\mathbf{u}_h, \mathbf{v}_h)_T + (\mathbf{u}_h, \nabla q_h)_T \\ + (\nabla p_h, \mathbf{v}_h)_T + (\nabla p_h, \nabla q_h)_T], \end{aligned} \tag{4.39}$$

where we have moved the following term to the right-hand side

$$\Phi\left(\mathbf{v}_h, q_h,\right) = -\beta_s \sum_{T\in\mathcal{T}} \frac{h_T^2}{\epsilon^2 + h_T^2} \left[(\mathbf{f}, \mathbf{v}_h)_T + (\mathbf{f}, \nabla q_h)_T\right]. \tag{4.40}$$

4.3.3 Continuous interior penalty method

The second stabilization method we use is the continuous interior penalty method (CIP). This method is presented by Burman and Hansbo article [4] for the Brinkman equation, where they also prove stability. The method consists of adding the term

$$c\left(p_h, q_h\right) = \beta_s \sum_{T \in \mathcal{T}} \frac{h_T^3}{\epsilon^2 + h_T^2} \left(\left[\partial_n p_h\right], \left[\partial_n q_h\right]\right)_{\partial T}, \tag{4.41}$$

to the left-hand side of Equation (4.35). Here, $[\partial_n p_h]$ is the pressure gradient jump over the interior facets of element T, $\beta_s > 0$ is a stabilization parameter. The scaling function $h_T^3/(\epsilon^2 + h_T^2)$ in Equation (4.41) is not used in [4]. Burman and Hansbo use h_T^3 for $\epsilon^2 \geq h_T$ and h_T^2 for $\epsilon^2 < h_T$, which is not smooth when $\epsilon^2 \approx h_T$. We do not add anything on the right-hand side,

$$\Phi\left(\mathbf{v}_h, q_h,\right) = 0. \tag{4.42}$$

Analysis of the CIP stabilization method can be found in Chapter 7.

4.4 A priori error estimates

We wish to measure the convergence rate of our solution to verify the implementation and to compare the different stabilization techniques. To do this, we use an a priori error estimate for the *mini*–element; that is, $V_h = \left[P_1\left(\mathcal{T}\right) + B_3\left(\mathcal{T}\right)\right]^d$ and $Q_h = P_1\left(\mathcal{T}\right)$. To get this error estimate, we need some ingredients:

Theorem 4.1. ***Babuška–Brezzi theorem.***

Assume that we have a saddle point problem on the form (4.27) and the conditions (4.30)–(4.33) hold for some V_h and Q_h. Then

$$\|\mathbf{u} - \mathbf{u}_h\|_{V_h} + \|p - p_h\|_{Q_h} \leq C \left\{ \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_{V_h} + \inf_{q_h \in Q_h} \|p - q_h\|_{Q_h} \right\}. \tag{4.43}$$

Proof. Proof can be found in [1]. □

Theorem 4.2. ***Approximation by interpolation.***

There exists an interpolation operator $\pi_h : H^t\left(\Omega\right) \to V_h$ where V_h is a piecewise polynomial field of order $t-1$, with the property that for any $u \in H^t(\Omega)$

$$\|u - \pi_h u\|_{H^m} \leq C h^{t-m} \|u\|_{H^t}. \tag{4.44}$$

Proof. Proof can be found in [21]. □

4.4.1 A priori error estimate: Stokes case

For the Stokes case ($\epsilon = 1$) the solution spaces are $\mathbf{u} \in \left[H_g^1(\Omega)\right]^d$ and $p \in L_0^2(\Omega)$. Theorem 4.1 becomes

$$\|\mathbf{u} - \mathbf{u}_h\|_{H^1} + \|p - p_h\|_{L^2} \leq C \left\{ \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_{H^1} + \inf_{q_h \in Q_h} \|p - q_h\|_{L^2} \right\}.$$

We use the interpolation approximation from Theorem 4.2 on both $\inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_{H^1}$ and $\inf_{q_h \in Q_h} \|p - q_h\|_{L^2}$, where m is 1 and 0, respectively, and t is 2 and 1, respectively.

$$\|\mathbf{u} - \mathbf{u}_h\|_{H^1} + \|p - p_h\|_{L^2} \leq Ch \left(\|\mathbf{u}\|_{H^2} + \|p\|_{H^1}\right). \tag{4.45}$$

Form the equation above we see that the L^2-error for the pressure will have at least first order convergence and the H^1-error for the velocity will have at least first order convergence. We measure the L^2-error and not the H^1-error of the velocity, thus we expect second order convergence (from Aubin-Nitsche's duality techniques [22]).

4.4.2 A priori error estimate: Darcy case

For the Darcy case ($\epsilon = 0$) we choose the solution spaces to be $\mathbf{u} \in \left[L_g^2(\Omega)\right]^d$ and $p \in H^1(\Omega) \cap L_0^2(\Omega)$. Theorem 4.1 becomes

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2} + \|p - p_h\|_{H^1} \leq C \left\{ \inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_{L^2} + \inf_{q_h \in Q_h} \|p - q_h\|_{H^1} \right\}.$$

As before, we use the interpolation approximation from Theorem 4.2 on both $\inf_{\mathbf{v}_h \in V_h} \|\mathbf{u} - \mathbf{v}_h\|_{H^1}$ and $\inf_{q_h \in Q_h} \|p - q_h\|_{L^2}$, but now m is 0 and 1, respectively, and t is 1 and 2, respectively. We obtain

$$\|\mathbf{u} - \mathbf{u}_h\|_{L^2} + \|p - p_h\|_{H^1} \leq Ch \left(\|\mathbf{u}\|_{H^1} + \|p\|_{H^2}\right). \tag{4.46}$$

We see that the L^2-error for the velocity has at least first order convergence. The H^1-error for the pressure has first order convergence, thus we expect second order convergence for the L^2-error.

When applying Theorem 4.2 for the two previous error estimates, we assumed one order higher regularity then what the solution space is.

4.5 Results

4.5.1 Methods of manufactured solutions

The method of manufactured solutions can be used to measure the error for a particular problem, and thus the convergence rate. The method consists of choosing a simple domain, we use the unit square, and choosing a solution $(\mathbf{u}, p)$ that satisfies the initial condition and boundary condition. In our case we need to choose a divergence free velocity field $\mathbf{u}$. We choose the same velocity field as used in [2], $\mathbf{u} = \nabla \times (\sin^2(\pi x)\sin^2(\pi y))$. This is divergence free. The pressure is chosen to be $p = -sin(2\pi x)$. Now we calculate the source function $\mathbf{f}$ such that Equation (4.23) hold; that is, we calculate $\mathbf{f} = \mathbf{u} - \epsilon^2 \Delta \mathbf{u} + \nabla p$. Let us summarize the test problem:

$$\begin{aligned} \Omega &= \left\{(x, y) \in \mathbb{R}^2 |\; x, y \in (0, 1)\right\}, \\ \mathbf{u} &= \nabla \times \left(\sin^2(\pi x)\sin^2(\pi y)\right), \\ p &= -\sin(2\pi x), \\ \mathbf{f} &= \mathbf{u} - \epsilon^2 \Delta \mathbf{u} + \nabla p, \\ \beta_s &= 0.1, \end{aligned} \tag{4.47}$$

where β_s is the stabilization constant. The mesh consists of $N \times N$ triangles. After solving this problem, we measure the relative L^2-error as,

$$\frac{\|\mathbf{u} - \mathbf{u_h}\|_\Omega}{\|\mathbf{u}\|_\Omega} = \frac{\sqrt{\int_\Omega \left(\mathbf{u} - \mathbf{u}_h\right)^2 \, d\mathbf{x}}}{\sqrt{\int_\Omega \mathbf{u}^2 \, d\mathbf{x}}}, \tag{4.48}$$

and similar for the pressure. The convergence rate is found by taking the logarithm of both the errors and mesh size h, then using the least squares method.

Python code

```python
from numpy import *

def convergence_rate(error_array, h_array):
    y = log(error_list)
    x = log(h_list)
    A = array([ x, ones(len(x))])
    rate = linalg.lstsq(A.T,y)[0]
    return rate[0]
```

The problem is implemented in python using the *FEniCS* software [25], code can be found in Appendix B.

4.5.2 The *mini*–element

Table 4.1 and 4.2 show the errors from the *mini*-element for velocity and pressure respectively. The left column contains the ϵ values and the right column shows the convergence rates. The theory from Section 4.4 tells us that we would expect second order convergence for the velocity for $\epsilon = 1.0$ and first order for $\epsilon = 0.0$. However, we get second order for all ϵ. Thus we get a bit better results than expected. For the pressure we expect first order convergence for $\epsilon = 1.0$ and second order for $\epsilon = 0.0$. From the table we see that we get slightly better convergence rates.

$\epsilon \backslash N$	8	16	32	64	128	rate
1	1.12e-01	2.87e-02	7.20e-03	1.80e-03	4.48e-04	1.99
2^{-2}	9.69e-02	2.43e-02	6.06e-03	1.51e-03	3.77e-04	2.00
2^{-4}	5.52e-02	1.25e-02	3.02e-03	7.48e-04	1.86e-04	2.05
2^{-8}	1.35e-01	2.86e-02	4.29e-03	6.69e-04	1.84e-04	2.45
0	1.49e-01	4.20e-02	1.10e-02	2.82e-03	7.13e-04	1.93

Table 4.1: The relative L^2-error in velocity obtained by the *mini*–element.

$\epsilon \backslash N$	8	16	32	64	128	rate
1	2.81e-00	8.85e-01	2.95e-01	1.02e-01	3.58e-02	1.57
2^{-2}	1.91e-01	5.76e-02	1.88e-02	6.45e-03	2.25e-03	1.60
2^{-4}	5.23e-02	1.33e-02	3.42e-03	8.99e-04	2.45e-04	1.94
2^{-8}	3.93e-02	1.05e-02	2.83e-03	7.61e-04	1.99e-04	1.90
0	3.32e-02	7.77e-03	1.89e-03	4.66e-04	1.16e-04	2.04

Table 4.2: The relative L^2-error in pressure obtained by the *mini*–element.

4.5.3 Pressure stabilized Petrov–Galerkin method

Table 4.3 and 4.4 show the errors from the PSPG stabilization for velocity and pressure, respectively. The results are similar to the *mini*-element. We have second order convergence rates through the whole range of ϵ, for the velocity.

4.5.4 Continuous interior penalty method

Table 4.5 and 4.6 show the errors from the CIP stabilization for the velocity and the pressure, respectively. These are similar to the *mini*-element and the PSPG stabilization.

These results indicate that both the PSPG and CIP stabilization are suitable to use for the Brinkman equation for $\epsilon \in [0, 1]$.

$\epsilon \backslash N$	8	16	32	64	128	rate
1	1.61e-01	4.44e-02	1.14e-02	2.88e-03	7.21e-04	1.95
2^{-2}	1.65e-01	4.68e-02	1.22e-02	3.08e-03	7.71e-04	1.94
2^{-4}	1.13e-01	3.63e-02	1.19e-02	3.44e-03	9.01e-04	1.73
2^{-8}	8.38e-02	2.12e-02	5.36e-03	1.39e-03	3.98e-04	1.94
0	8.36e-02	2.11e-02	5.29e-03	1.32e-03	3.30e-04	2.00

Table 4.3: The relative L^2-error in velocity obtained by the PSPG stabilization ($\beta_s = 0.1$).

$\epsilon\backslash N$	8	16	32	64	128	rate
1	7.53e-01	2.93e-01	1.03e-01	3.62e-02	1.27e-02	1.48
2^{-2}	7.32e-02	2.14e-02	7.08e-03	2.39e-03	8.18e-04	1.61
2^{-4}	6.14e-02	1.52e-02	3.66e-03	8.98e-04	2.26e-04	2.03
2^{-8}	5.97e-02	1.52e-02	3.82e-03	9.55e-04	2.37e-04	1.99
0	5.97e-02	1.52e-02	3.83e-03	9.57e-04	2.39e-04	1.99

Table 4.4: The relative L^2-error in pressure obtained by the PSPG stabilization ($\beta_s = 0.1$).

$\epsilon\backslash N$	8	16	32	64	128	rate
1	1.64e-01	4.61e-02	1.18e-02	2.97e-03	7.43e-04	1.95
2^{-2}	1.57e-01	4.13e-02	1.04e-02	2.61e-03	6.54e-04	1.98
2^{-4}	1.97e-01	4.62e-02	8.16e-03	1.66e-03	4.03e-04	2.27
2^{-8}	2.25e-01	6.23e-02	1.56e-02	3.80e-03	8.74e-04	2.01
0	2.25e-01	6.25e-02	1.58e-02	3.91e-03	9.69e-04	1.97

Table 4.5: The relative L^2-error in velocity obtained by the CIP stabilization ($\beta_s = 0.1$).

$\epsilon\backslash N$	8	16	32	64	128	rate
1	1.06e+00	4.61e-01	1.36e-01	3.94e-02	1.20e-02	1.65
2^{-2}	2.15e-01	4.45e-02	1.02e-02	2.76e-03	8.08e-04	2.01
2^{-4}	1.49e-01	3.47e-02	6.16e-03	1.11e-03	2.42e-04	2.35
2^{-8}	1.35e-01	3.61e-02	9.08e-03	2.24e-03	5.29e-04	2.00
0	1.35e-01	3.61e-02	9.11e-03	2.27e-03	5.67e-04	1.98

Table 4.6: The relative L^2-error in pressure obtained by the CIP stabilization ($\beta_s = 0.1$).

Chapter 5

Numerical methods II: Unfitted meshes

In Chapter 4 we presented numerical methods for the Brinkman equation. Chapter 4 assumes that the mesh $\mathcal{T}$ matches the domain Ω, which is common for finite element methods. In this chapter we introduce a fictitious domain method. Figure 5.1 shows a matching mesh (left) and a non-matching mesh (right). Ω is the domain, $\hat{\mathcal{T}}^*$ is called the background mesh. $\hat{\mathcal{T}}^*$ is structured independently of the domain.

To describe the domain Ω we use level set functions $S(\mathbf{x})$. The level set functions are designed such that the following properties hold,

$$\begin{aligned} S\left(\mathbf{x}\right) &< 1 \quad \text{if} \quad \mathbf{x} \in \Omega, \\ S\left(\mathbf{x}\right) &= 1 \quad \text{if} \quad \mathbf{x} \in \Gamma, \\ S\left(\mathbf{x}\right) &> 1 \quad \text{if} \quad \mathbf{x} \notin \Omega. \end{aligned} \tag{5.1}$$

Figure 2.2 was constructed from a level set function. The source code can be found in appendix B.

The method of unfitted meshes creates several challenges which we deal with in this chapter. In Chapter 6 we test the methods in this chapter on similar test case as Section 4.5.

In Problem 4.5, we found a weak formulation of the Brinkman equation. We looked for a solution $(\mathbf{u}_h, p_h) \in V_{g,h} \times Q_h$ and we used the test function spaces $\hat{V}_{0,h}$ and $\hat{Q}_h$. In the unfitted case we can no longer let $\mathbf{u}_h \in V_{g,h}$ since we have no guarantee that the nodes will lie on the boundary Γ. Instead we let $\mathbf{u}_h, \mathbf{v}_h \in V_h$ and weakly impose the Dirichlet boundary condition.

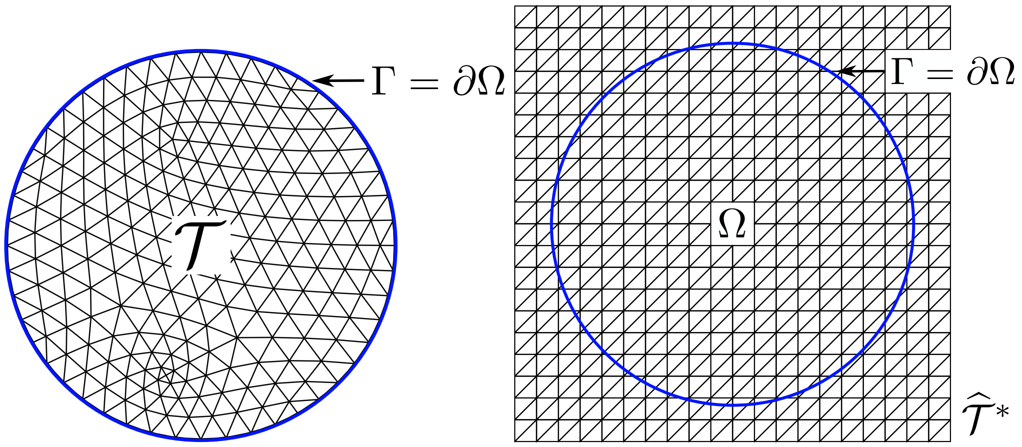
$\Gamma = \partial\Omega$
$\mathcal{T}$
$\Gamma = \partial\Omega$
Ω
$\widehat{\mathcal{T}}^*$

Figure 5.1: The left figure shows a matching mesh. The right figure shows a non-matching mesh. The mesh $\widehat{\mathcal{T}}^*$ in the right figure is called a background mesh.

5.1 The Nitsche method

Nitsche's method is a method to weakly impose Dirichlet boundary conditions. The idea of the method was introduced by Nitsche [9]. However, we have used [26] for the description of the method. This article shows stability and an a priori error estimate for the Poisson problem. We will apply the method to the Brinkman Equation (4.23) and derive a weak formulation.

We multiply the discrete version of Equation (4.23) with a test function $\mathbf{v}_h \in V_h$ and integrate over the domain,

$$(\mathbf{u}_h, \mathbf{v}_h)_\Omega - \epsilon^2 \left(\Delta \mathbf{u}_h, \mathbf{v}_h\right)_\Omega + \left(\nabla p_h, \mathbf{v}_h\right)_\Omega = \left(\mathbf{f}, \mathbf{v}_h\right)_\Omega . \tag{5.2}$$

Now we use integration by parts on two of the terms on the left-hand side,

$$\begin{aligned} &(\mathbf{u}_h, \mathbf{v}_h)_\Omega + \epsilon^2 \left(\nabla \mathbf{u}_h, \nabla \mathbf{v}_h\right)_\Omega - \left(p_h, \nabla \cdot \mathbf{v}_h\right)_\Omega \\ &\qquad -\epsilon^2 \left(\partial_\mathbf{n} \mathbf{u}_h, \mathbf{v}_h\right)_\Gamma + \left(p_h, \mathbf{v}_h \cdot \mathbf{n}\right)_\Gamma = \left(\mathbf{f}, \mathbf{v}_h\right)_\Omega . \end{aligned} \tag{5.3}$$

The surface integrals no longer disappear. We add a penalty term $\gamma \frac{\epsilon^2}{h} \left(\mathbf{u}_h - \mathbf{g}, \mathbf{v}_h\right)_\Gamma$ to Equation (5.3), where γ is a penalty parameter. The incompressibility equation is multiplied with a test function $q_h \in Q_h$ and integrated over the domain. We add $\left(\left(\mathbf{u}_h - \mathbf{g}\right) \cdot \mathbf{n}, q_h\right)_\Gamma$ to the incompressibility equation. We add the term $-\epsilon^2 \left(\left(\mathbf{u}_h - \mathbf{g}\right), \partial_\mathbf{n} \mathbf{v}_h\right)_\Gamma$ to Equation (5.3). The last two steps gives us symmetry. The problem is stated as:

Problem 5.1. *Find $\mathbf{u}_h \in V_h$ and $p_h \in Q_h$ such that*

$$A\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right) = L\left(\mathbf{v}_h, q_h\right), \quad \forall\, q_h \in Q_h,\ \mathbf{v}_h \in V_h, \tag{5.4}$$

where

$$\begin{aligned}
A\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right) &= a\left(\mathbf{u}_h, \mathbf{v}_h\right) + b\left(p_h, \mathbf{v}_h\right) + b\left(\mathbf{u}_h, q_h\right) + \underbrace{c\left(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h\right)}_{\textit{Stabilization}},\\
L\left(\mathbf{v}_h, q_h\right) &= \left(\mathbf{f}, \mathbf{v}_h\right)_\Omega - \epsilon^2\left(\mathbf{g}, \partial_\mathbf{n}\mathbf{v_h}\right)_\Gamma + \left(\mathbf{g}\cdot\mathbf{n}, q_h\right)_\Gamma\\
&\quad + \underbrace{\Phi\left(\mathbf{v}_h, q_h\right)}_{\textit{Consistency of stabilization}} + \underbrace{\gamma\frac{\epsilon^2}{h}\left(\mathbf{g}, \mathbf{v}_h\right)_\Gamma}_{\textit{Nitsche penalty}},\\
a\left(\mathbf{u}_h, \mathbf{v}_h\right) &= \left(\mathbf{u}_h, \mathbf{v}_h\right)_\Omega + \epsilon^2\left(\nabla\mathbf{u}_h, \nabla\mathbf{v}_h\right)_\Omega\\
&\quad - \underbrace{\epsilon^2\left(\partial_\mathbf{n}\mathbf{u}_h, \mathbf{v}_h\right)_\Gamma}_{\textit{Consistency}} - \underbrace{\epsilon^2\left(\mathbf{u}_h, \partial_\mathbf{n}\mathbf{v}_h\right)_\Gamma}_{\textit{Symmetry}} + \underbrace{\gamma\frac{\epsilon^2}{h}\left(\mathbf{u}_h, \mathbf{v}_h\right)_\Gamma}_{\textit{Nitsche penalty}},\\
b\left(p_h, \mathbf{v}_h\right) &= -\left(p_h, \nabla\cdot\mathbf{v}_h\right)_\Omega + \underbrace{\left(p_h, \mathbf{v}_h\cdot\mathbf{n}\right)_\Gamma}_{\textit{Symmetry}},
\end{aligned}$$

where $c\left(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h\right)$ and $\Phi\left(\mathbf{v}_h, q_h\right)$ depend on the stabilization, see Section 4.3.

5.1.1 The Nitsche method without stabilization

We test the Nitsche method on the same test problem as in Section 4.5 on a matching mesh for P_1–P_1 elements without any stabilization; that is, $c\left(\mathbf{u}_h, p_h, ; \mathbf{v}_h, q_h\right) = \Phi\left(\mathbf{v}_h, q_h\right) = 0$. This problem does not satisfy the inf-sup condition (4.34) and we expect suboptimal convergence rates. The errors and convergence rates are found in Table 5.1 and 5.2. From these tables we see that the error and convergence rates are good and similar to what we found in Section 4.5. However, plot (Figure 5.2) of the pressure shows small oscillations. It seems that the Nitsche method has a stabilizing effect. This will be discussed further in Chapter 6, 8 and 9.

$\epsilon\backslash N$	8	16	32	64	128	rate
1	1.61e-01	4.51e-02	1.17e-02	2.95e-03	7.41e-04	1.95
2^{-2}	1.44e-01	3.99e-02	1.03e-02	2.60e-03	6.52e-04	1.95
2^{-4}	9.55e-02	2.51e-02	6.37e-03	1.60e-03	4.01e-04	1.98
2^{-8}	8.08e-02	2.07e-02	5.20e-03	1.30e-03	3.26e-04	1.99
0	8.08e-02	2.07e-02	5.20e-03	1.30e-03	3.26e-04	1.99

Table 5.1: The relative L^2-error in velocity obtained by the Nitsche method. No stabilization is used.

$\epsilon\backslash N$	8	16	32	64	128	rate
1	1.42e+00	4.81e-01	1.53e-01	5.48e-02	2.28e-02	1.50
2^{-2}	1.12e-01	3.51e-02	1.06e-02	3.62e-03	1.46e-03	1.58
2^{-4}	5.48e-02	1.43e-02	3.63e-03	9.23e-04	2.42e-04	1.96
2^{-8}	5.39e-02	1.40e-02	3.52e-03	8.82e-04	2.21e-04	1.98
0	5.39e-02	1.40e-02	3.52e-03	8.82e-04	2.21e-04	1.99

Table 5.2: The relative L^2-error in pressure obtained by the Nitsche method, no stabilization is used.

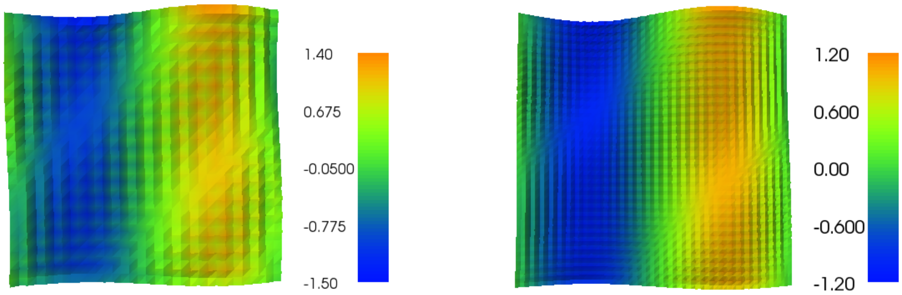

Figure 5.2: The plots show the pressure when solving with the Nitsche method for P_1–P_1 elements without stabilization. $N = 32$ for the left plot and $N = 64$ for the right plot.

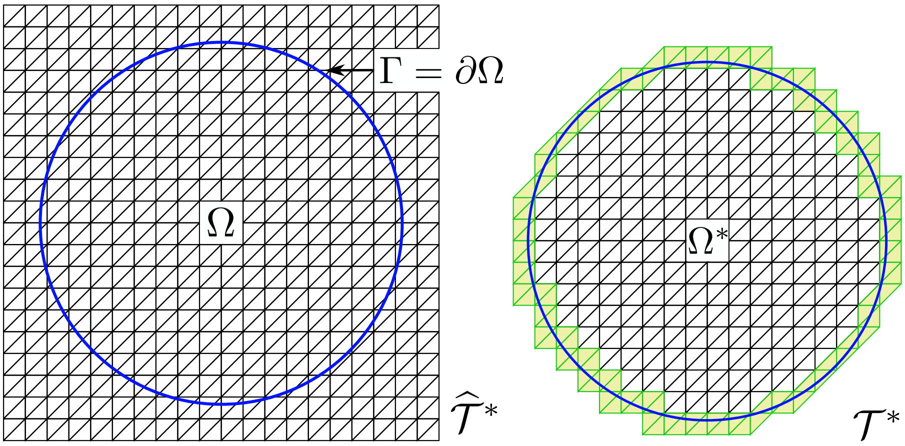

Figure 5.3: A reduced background mesh $\mathcal{T}^*$ for the fictitious domain Ω^*. The green elements are the cut elements.

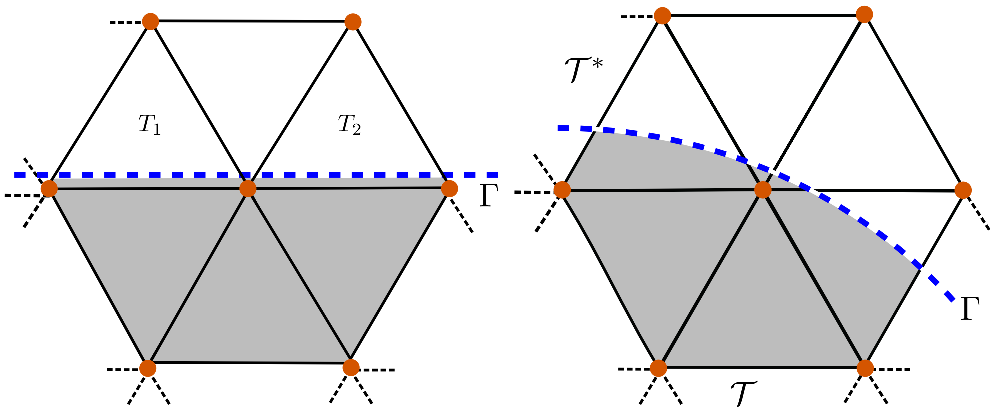

Figure 5.4: The left figure illustrates a case where the constant C_I in Equation (7.14) becomes large. The right figure shows a case where the matrix stability becomes a problem.

5.2 Cut elements

The left figure in Figure 5.3 shows a unfitted mesh. We see that some of the elements in the background mesh $\hat{\mathcal{T}}^*$ do not interact with $\bar{\Omega}$. These elements can be removed from the mesh $\hat{\mathcal{T}}^*$; that is, we remove all $T \in \hat{\mathcal{T}}^*$ such that

$$T \cap \bar{\Omega} = \emptyset,$$

and obtain a reduced mesh $\mathcal{T}^*$. This is illustrated in the right part of Figure 5.3. The elements that are intersected by the boundary Γ; that is,

$$T \cap \Omega \neq \emptyset \quad \text{and} \quad T \setminus \bar{\Omega} \neq \emptyset,$$

are called cut elements and are denoted by the symbol $\mathcal{T}_\Gamma^*$. The cut elements are marked green in Figure 5.3. We define $\mathcal{F}_\Gamma^*$ as the set of all interior facets belonging to elements which are intersected by the boundary Γ. We also define a fictitious domain Ω^* which contains Ω and the cut elements. Ω will be referred to as the physical domain and Ω^* as the fictitious domain.

We use the same assumptions on Γ and $\mathcal{T}^*$ as in [8]:

I The intersection between Γ and a facet $F \in \partial_i \mathcal{T}^*$ is simply connected; that is, Γ does not cross an interior facet multiple times.

II For each element T intersected by Γ, there exists a plane S_T and a piecewise smooth parametrization $\Phi : S_T \cap T \to \Gamma \cap T$.

III The number of facets to be crossed in order to "walk" from a cut element T to a non-cut element $T' \subset \Omega$ is bounded.

One major problem with the cut elements is that we have no upper bound for inverse estimate 7.1. This estimate is used to show coercivity, $a(\mathbf{u}_h, \mathbf{u}_h) \geq C_3 \|\mathbf{u}_h\|_{\epsilon,h}^2$, which we will come back to in Chapter 7. The consequence is that the constant C_3 becomes large when the area of the cut elements are small compared to the surface. The left illustration in Figure 5.4 shows such a case. C_3 will depend on how the boundary cuts the mesh, which in turn makes the penalty parameter γ depend on how the boundary cuts the mesh. See the proof of Theorem 7.5 for details. In addition the matrix A may become almost singular if a cut element is cut in such a way that only a small fraction is inside the domain. The right illustration in Figure 5.4 shows such a case.

Extending the coercivity from the physical domain to the fictitious domain will fix these problems. Burman and Hansbo [7, 5] did this by adding artificial stabilization operators (ghost-penalties), $i_h(\mathbf{u}_h, \mathbf{v}_h)$ and $j_h(p_h, q_h)$, in the interface zone, also outside the physical domain. We use the following ghost-penalty for velocity and pressure respectively,

$$i_h(\mathbf{u}_h, \mathbf{v}_h) = \beta_{\mathbf{u}} \sum_{F \in \mathcal{F}_\Gamma^*} s_{\mathbf{u}}\left(h_F\right) \left(\left[\partial_{\mathbf{n}} \mathbf{u}_h\right], \left[\partial_{\mathbf{n}} \mathbf{v}_h\right]\right)_F, \tag{5.5}$$

$$j_h(p_h, q_h) = \beta_p \sum_{F \in \mathcal{F}_\Gamma^*} s_p\left(h_F\right) \left(\left[\partial_{\mathbf{n}} p_h\right], \left[\partial_{\mathbf{n}} q_h\right]\right)_F, \tag{5.6}$$

where $s_{\mathbf{u}}$ and s_p are scaling functions, $\beta_{\mathbf{u}}$ and β_p are penalty parameters. For the Stokes problem (with P_1–P_1 elements) $s_{\mathbf{u}} = h_T$ and $s_p = h_T^3$ are used in [6] and [8]. Massing et al. [8] provide a useful proposition:

Proposition 5.1.

There is a constant $C > 0$ such that the following estimates hold. For $q_h \in P_1\left(\mathcal{T}^*\right)$:

$$\|\nabla q_h\|_{\Omega^*}^2 \leq C \left(\|\nabla q_h\|_{\Omega}^2 + \sum_{F \in \mathcal{F}_\Gamma^*} h_F \left(\left[\partial_{\mathbf{n}} q_h\right], \left[\partial_{\mathbf{n}} q_h\right]\right)_F \right) \leq C \|\nabla q_h\|_{\Omega^*}^2, \tag{5.7}$$

$$\|q_h\|_{\Omega^*}^2 \leq C \left(\|q_h\|_{\Omega}^2 + \sum_{F \in \mathcal{F}_\Gamma^*} h_F^3 \left(\left[\partial_{\mathbf{n}} q_h\right], \left[\partial_{\mathbf{n}} q_h\right]\right)_F \right) \leq C \|q_h\|_{\Omega^*}^2. \tag{5.8}$$

Proof. The proof can be found in [8]. This proposition also extends to vector valued functions like $\mathbf{v}_h$. □

We use Proposition 5.1 to find suitable scaling functions $s_{\mathbf{u}}$ and s_p. The velocity field is measured in the H^1-norm for the Stokes case and the L^2-norm for the Darcy case. From the proposition we want $s_{\mathbf{u}} = h_T$ for $\epsilon = 1$ and $s_{\mathbf{u}} = h_T^3$ for $\epsilon = 0$. Therefore, we construct the scaling function

$$s_{\mathbf{u}} = h_T \left(\epsilon^2 + h_T^2\right). \tag{5.9}$$

For the pressure we wish to measure the L^2-norm for the Stokes case and the H^1-norm for the Darcy case. From the Proposition 5.1 we have $s_p = h_T^3$ for $\epsilon = 1$ and $s_p = h_T$ for $\epsilon = 0$. We construct the scaling function,

$$s_p = \frac{h_T^3}{\epsilon^2 + h_T^2}. \tag{5.10}$$

5.2.1 Quadrature integration rules

Integration over the cut elements is a challenging task since the geometry is no longer a tetrahedron, but in a sense arbitrary. One approach is subtetrahedralization, which divides tetrahedron into several subtetrahedron to fit the geometry and uses standard integration schemes on the subtetrahedron. However, this is also quite challenging. We use quadrature integration rules which are implemented in the library DOLFIN-OLM [27], which is an extension of the *FEniCS* [25] software. These integration rules are only implemented for the P_0 and P_1 elements. For higher order elements these integration rules are challenging to implement. This is the main reason why we are restricted to the P_1–P_1 element. Integration over the cut elements is a huge topic which we will not explore further in this thesis. See [27, 28, 29] for more information on this topic.

5.2.2 The final problem

In this subsection we state the final problem. Let $V_h = [P_1(\mathcal{T}^*)]^d$ and $Q_h = P_1(\mathcal{T}^*)$, $\beta_{\mathbf{u}}$, β_p, β_s and γ are penalty constants,

Problem 5.2. *Find* $\mathbf{u}_h \in V_h$ *and* $p_h \in Q_h$ *such that*

$$A\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right) + J\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right) = L\left(\mathbf{v}_h, q_h\right), \quad \forall\, q_h \in Q_h,\ \mathbf{v}_h \in V_h,$$

where

$$\begin{aligned}
A\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right) &= a\left(\mathbf{u}_h, \mathbf{v}_h\right) + b\left(p_h, \mathbf{v}_h\right) + b\left(\mathbf{u}_h, q_h\right) + \underbrace{c\left(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h\right)}_{\textit{Stabilization}}, \\
J\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right) &= \underbrace{\beta_{\mathbf{u}} \sum_{F \in \mathcal{F}_\Gamma^*} s_{\mathbf{u}}\left(\left[\partial_{\mathbf{n}} \mathbf{u}_h\right], \left[\partial_{\mathbf{n}} \mathbf{v}_h\right]\right)_F}_{\textit{Ghost-penalty}} + \underbrace{\beta_p \sum_{F \in \mathcal{F}_\Gamma^*} s_p\left(\left[\partial_{\mathbf{n}} p_h\right], \left[\partial_{\mathbf{n}} q_h\right]\right)_F}_{\textit{Ghost-penalty}}, \\
L\left(\mathbf{v}_h, q_h\right) &= \left(\mathbf{f}, \mathbf{v}_h\right)_\Omega - \epsilon^2\left(\mathbf{g}, \partial_{\mathbf{n}} \mathbf{v_h}\right)_\Gamma - \left(\mathbf{g} \cdot \mathbf{n}, q_h\right)_\Gamma \\
&\quad + \underbrace{\Phi\left(\mathbf{v}_h, q_h\right)}_{\textit{Consistency of stabilization}} + \underbrace{\gamma \frac{\epsilon^2}{h}\left(\mathbf{g}, \mathbf{v}_h\right)_\Gamma}_{\textit{Nitsche penalty}}, \\
a\left(\mathbf{u}_h, \mathbf{v}_h\right) &= \left(\mathbf{u}_h, \mathbf{v}_h\right)_\Omega + \epsilon^2\left(\nabla \mathbf{u}_h, \nabla \mathbf{v}_h\right)_\Omega \\
&\quad - \underbrace{\epsilon^2\left(\partial_{\mathbf{n}} \mathbf{u}_h, \mathbf{v}_h\right)_\Gamma}_{\textit{Consistency}} - \underbrace{\epsilon^2\left(\mathbf{u}_h, \partial_{\mathbf{n}} \mathbf{v}_h\right)_\Gamma}_{\textit{Symmetry}} + \underbrace{\gamma \frac{\epsilon^2}{h}\left(\mathbf{u}_h, \mathbf{v}_h\right)_\Gamma}_{\textit{Nitsche penalty}}, \\
b\left(p_h, \mathbf{v}_h\right) &= -\left(p_h, \nabla \cdot \mathbf{v}_h\right)_\Omega + \underbrace{\left(p_h, \mathbf{v}_h \cdot \mathbf{n}\right)_\Gamma}_{\textit{Symmetry}},
\end{aligned}$$

where $c\left(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h\right)$ *and* $\Phi\left(\mathbf{v}_h, q_h\right)$ *depend on the stabilization, see Section 4.3. The scaling functions* $s_{\mathbf{u}}$ *and* s_p *depend on* h_T *and* ϵ^2*, see Equation (5.9) and (5.10) for suggestions.*

Chapter 6

Results

This chapter presents the errors and convergence rates from solving Problem 5.2 with the method of manufactured solutions.

6.1 The test problem

We use the same manufactured solution as in Section 4.5. The physical domain Ω is the unit square and the fictitious domain is a unit square which is δ larger at each side, see Figure 6.1. When $\delta \to 0$, the problem reduces to a matching mesh with the Nitsche method. We let $\delta = 1.1 \cdot 10^{-3}$ unless stated otherwise. Let us state the manufactured solution:

$$
\begin{aligned}
\Omega &= \Big\{(x,y) \in \mathbb{R}^2 |\ x,y \in (0,1)\Big\}, \\
\Omega^* &= \Big\{(x,y) \in \mathbb{R}^2 |\ x,y \in (-\delta, 1+\delta)\Big\}, \\
\mathbf{u} &= \nabla \times \Big(\sin^2(\pi x)\sin^2(\pi y)\Big), \\
p &= -\sin(2\pi x), \\
\mathbf{f} &= \mathbf{u} - \epsilon^2 \Delta \mathbf{u} + \nabla p, \\
\beta_s &= \beta_p = \beta_{\mathbf{u}} = 0.1, \quad \gamma = 10.0.
\end{aligned}
\tag{6.1}
$$

We solved the problem with *C++* and using the DOLFIN-OLM [27] library. The UFL code [30] for Problem 5.2 with CIP stabilization, can by found in B.3.

Remark: The software we used only supported 3-dimensional problems and our test problem is a 2-dimensional problem. To solve this we scaled the z-axis with $1/N$ and use homogeneous Neumann conditions on the xy-surfaces. This created a quasi 2-dimensional problem.

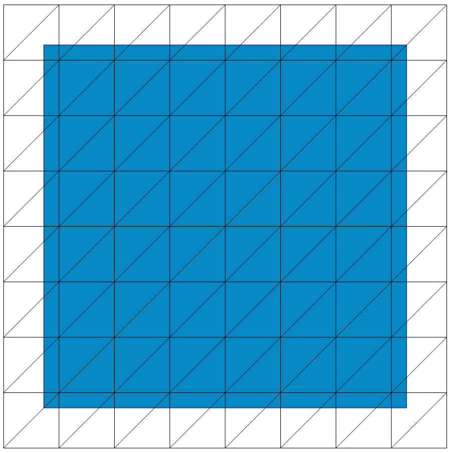

Figure 6.1: Illustration of the meshes. The physical domain is an unit square (blue) and the background is a box which is δ lager in sides.

$s_{\mathbf{u}} \backslash N$	8	16	32	64	128	rate
h_T^3	2.45e-01	1.16e-01	3.55e-01	1.06e+00	1.30e+00	-0.80
h_T^1	4.35e-01	1.52e-01	5.26e-02	3.43e-02	1.20e-01	0.59
h_T^{-1}	6.27e-01	2.69e-01	9.41e-02	3.18e-02	1.05e-02	1.49

Table 6.1: The relative L^2-error for velocity obtained by the CIP stabilization.

6.2 Darcy flow

When computing the Darcy flow; that is, $\epsilon = 0$, we did not get the expected convergence rates. From Proposition 5.1 we expect $s_{\mathbf{u}} = h_T^3$ since the velocity is in the L^2 space. We tested for $s_{\mathbf{u}} = h_T$ and $s_{\mathbf{u}} = h_T^{-1}$ as well, s_p was set to h_T. The results for the CIP stabilization are shown in Table 6.1 and 6.2.

From the tables we see that the error does not converge for $s_{\mathbf{u}} = h_T^3$ or $s_{\mathbf{u}} = h_T$. For $s_{\mathbf{u}} = h_T^{-1}$ we get a sufficient convergence for velocity and pressure (other values for s were also tested). These somewhat unexpected results motivate further investigation of the weak formulation of the Darcy flow, which we do for the fitted case. First we state

$s_p \backslash N$	8	16	32	64	128	rate
h_T^3	9.45e-02	4.92e-02	1.55e-01	3.12e-01	3.94e-01	-0.68
h_T^1	1.18e-01	2.61e-02	8.64e-03	2.27e-02	9.43e-02	0.08
h_T^{-1}	2.60e-01	6.53e-02	1.34e-02	2.70e-03	5.82e-04	2.22

Table 6.2: The relative L^2-error in pressure obtained by the CIP stabilization.

the Darcy equation on the strong form,

$$\begin{aligned} \mathbf{u} + \nabla p &= \mathbf{f} && \text{for } \mathbf{x} \in \Omega, \\ \nabla \cdot \mathbf{u} &= 0 && \text{for } \mathbf{x} \in \Omega, \\ \mathbf{u} &= \mathbf{g} && \text{for } \mathbf{x} \in \Gamma. \end{aligned} \tag{6.2}$$

Note that if we take the divergence of the first equation in (6.2), we get the Poisson equation with respect to the pressure with Neumann boundary conditions, $\mathbf{n} \cdot \nabla p = \mathbf{n} \cdot (\mathbf{f} - \mathbf{g})$. Thus the Darcy flow is a mixed Poisson equation. It can be viewed in two different ways, which we call the primal mixed method and the dual mixed method.

6.2.1 The primal mixed method

We multiply the first equation in the discrete version of (6.2) with a test function $\mathbf{v}_h \in V_h$ and integrate over the domain. For the second equation we multiply with $q_h \in Q_h$ and perform an integration by parts, obtaining

$$(\mathbf{u}_h, \nabla q_h)_\Omega = (\mathbf{g}, q_h \, \mathbf{n})_\Gamma \,. \tag{6.3}$$

Adding the equations together we obtain the weak formulation:

Problem 6.1. *Find $\mathbf{u}_h \in V_h$ and $p_h \in Q_h$ such that*

$$(\mathbf{u}_h, \mathbf{v}_h)_\Omega + (\nabla p_h, \mathbf{v}_h)_\Omega + (\mathbf{u}_h, \nabla q_h)_\Omega = (\mathbf{f}, \mathbf{v}_h)_\Omega + (\mathbf{g}, q_h \, \mathbf{n})_\Gamma \,, \tag{6.4}$$

for all $\mathbf{v}_h \in V_h$ and $q_h \in Q_h$.

We see that $V_h \subset \left[L^2(\Omega)\right]^d$ and $Q_h \subset H^1(\Omega)$. Condition (3.21) is enforced to achieve uniqueness of the pressure; that is, $Q_h \subset H^1(\Omega) \cap L_0^2(\Omega)$. These are the function spaces which were mentioned at the end of Chapter 3.

6.2.2 The dual mixed method

For the dual mixed method only the first equation in (6.2) is integrated by parts and not the second equation (unlike the primal mixed method). This is what we do for our Brinkman equation. Thus we can set $\epsilon = 0$ in Problem 5.1 and we obtain the same problem.

Problem 6.2. *Find $\mathbf{u}_h \in V_h$ and $p_h \in Q_h$ such that*

$$A\left((\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)\right) = L(\mathbf{v}_h, q_h), \quad \forall\, \mathbf{v}_h \in V_h,\ q_h \in Q_h, \tag{6.5}$$

where

$$\begin{aligned} A\left((\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)\right) &= (\mathbf{u}_h, \mathbf{v}_h)_\Omega + b(p_h, \mathbf{v}_h) + b(\mathbf{u}_h, q_h) + c(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h), \\ L(\mathbf{v}_h, q_h) &= (\mathbf{f}, \mathbf{v}_h)_\Omega + (\mathbf{g} \cdot \mathbf{n}, q_h)_\Gamma + \Phi(\mathbf{v}_h, q_h), \\ b(p_h, \mathbf{v}_h) &= -(p_h, \nabla \cdot \mathbf{v}_h)_\Omega + (p_h, \mathbf{v}_h \cdot \mathbf{n})_\Gamma \,. \end{aligned}$$

$c\left(\mathbf{u}_h, p_h; \mathbf{v}_h, q_h\right)$ *and* $\Phi\left(\mathbf{v}_h, q_h\right)$ *are stabilization terms.*

Now $V_h \subset H\left(div; \Omega\right)$ and $Q_h \subset L_0^2\left(\Omega\right)$ [2]. This means that the solution space in Problem 4.5, 5.1 and 5.2 are subset of $H\left(div; \Omega\right) \times L_0^2\left(\Omega\right)$ when $\epsilon = 0$ and not $\left[L^2\left(\Omega\right)\right]^d \times H^1\left(\Omega\right) \cap L_0^2\left(\Omega\right)$ as suggested in Chapter 3.

Problem 6.1 and 6.2 have two different solution spaces. However, for the P_1–P_1 elements these formulations are equivalent since $P_1\left(\Omega\right) \subset H^1\left(\Omega\right)$.

6.3 Brinkman problem with unfitted meshes

Table 6.1 and 6.2 show that we do not get sufficient convergence rates. We chose to try a different scaling function $s_{\mathbf{u}}(h_T)$. From now on we use

$$s_{\mathbf{u}}(h_T) = \frac{h_T}{\epsilon^2 + h_T^2} \quad \text{and} \quad s_p(h_T) = \frac{h_T^3}{\epsilon^2 + h_T^2} \tag{6.6}$$

and not the one suggested in Equation (5.9).

Table 6.3 and 6.4 show the relative L^2-error for PSPG stabilization for the velocity and pressure, respectively. The convergence rate for the velocity starts at second order for $\epsilon = 1$ and goes to one and a half order convergence for $\epsilon = 0$. In the fitted case (Table 4.3) we have second order convergence throughout the range of ϵ. Note that the convergence rate for $\epsilon = 2^{-4}$ is a bit lower than the other values of ϵ. The convergence rates for the pressure behaves similar to the fitted case (Table 4.4). The rates start at one and a half and goes to second order as $\epsilon \to 0$. The convergence rate for $\epsilon = 2^{-2}$ is quite low. An explanation is the error for $N = 16$ is larger than the error for $N = 8$. The method converges after some refinements of the mesh.

Table 6.5 and 6.6 show the relative L^2-error for CIP stabilization for the velocity and pressure, respectively. The convergence rates for the velocity and pressure are very similar to the PSPG stabilization. Also here, the convergence rate for the velocity at $\epsilon = 2^{-4}$ is a bit low compared to the other values of ϵ and it requires some refinements before the solution converges for the pressure at $\epsilon = 2^{-2}$.

$\epsilon \backslash N$	8	16	32	64	128	rate
1	1.45e-01	3.46e-02	8.04e-03	1.92e-03	4.71e-04	2.07
2^{-2}	6.15e-01	2.71e-01	8.69e-02	2.00e-02	3.64e-03	1.86
2^{-4}	6.99e-01	3.42e-01	1.44e-01	6.34e-02	2.71e-02	1.18
2^{-8}	5.49e-01	2.29e-01	8.65e-02	3.40e-02	1.27e-02	1.36
0	5.59e-01	2.42e-01	8.76e-02	3.04e-02	1.02e-02	1.45

Table 6.3: The relative L^2-error in velocity. PSPG stabilization was used.

$\epsilon \backslash N$	8	16	32	64	128	rate
1	7.89e-01	2.61e-01	8.92e-02	3.27e-02	1.34e-02	1.48
2^{-2}	1.12e-01	1.36e-01	9.63e-02	3.23e-02	7.59e-03	0.98
2^{-4}	1.54e-01	6.21e-02	2.33e-02	1.06e-02	5.13e-03	1.24
2^{-8}	1.70e-01	4.80e-02	1.23e-02	2.95e-03	6.88e-04	1.99
0	1.69e-01	4.30e-02	9.09e-03	1.82e-03	3.64e-04	2.23

Table 6.4: The relative L^2-error in pressure. PSPG stabilization was used.

$\epsilon \backslash N$	8	16	32	64	128	rate
1	1.47e-01	3.59e-02	8.25e-03	1.94e-03	4.70e-04	2.08
2^{-2}	6.09e-01	2.70e-01	8.71e-02	2.00e-02	3.64e-03	1.85
2^{-4}	7.09e-01	3.42e-01	1.43e-01	6.31e-02	2.71e-02	1.19
2^{-8}	6.13e-01	2.53e-01	9.17e-02	3.52e-02	1.29e-02	1.40
0	6.27e-01	2.69e-01	9.41e-02	3.18e-02	1.05e-02	1.49

Table 6.5: The relative L^2-error in velocity. CIP stabilization was used.

$\epsilon \backslash N$	8	16	32	64	128	rate
1	7.83e-01	4.66e-01	1.39e-01	4.10e-02	1.35e-02	1.52
2^{-2}	2.78e-01	2.03e-01	1.18e-01	3.73e-02	8.82e-03	1.24
2^{-4}	2.90e-01	1.01e-01	2.87e-02	1.17e-02	5.62e-03	1.45
2^{-8}	2.63e-01	7.09e-02	1.68e-02	3.84e-03	8.68e-04	2.07
0	2.60e-01	6.53e-02	1.34e-02	2.70e-03	5.82e-04	2.22

Table 6.6: The relative L^2-error in pressure. CIP stabilization was used.

$\epsilon\backslash N$	8	16	32	64	128	rate
1	1.48e-01	3.60e-02	8.26e-03	1.94e-03	4.69e-04	2.08
2^{-2}	6.10e-01	2.74e-01	8.77e-02	2.00e-02	3.64e-03	1.86
2^{-4}	7.13e-01	3.44e-01	1.43e-01	6.33e-02	2.72e-02	1.19
2^{-8}	5.93e-01	2.50e-01	9.32e-02	3.63e-02	1.34e-02	1.37
0	6.03e-01	2.62e-01	9.40e-02	3.24e-02	1.08e-02	1.46

Table 6.7: The relative L^2-error in velocity. No stabilization is used.

$\epsilon\backslash N$	8	16	32	64	128	rate
1	1.01e+00	5.65e-01	2.15e-01	7.74e-02	2.69e-02	1.33
2^{-4}	1.88e-01	2.61e-01	1.42e-01	4.93e-02	1.57e-02	0.96
2^{-4}	2.55e-01	1.04e-01	2.89e-02	1.34e-02	7.44e-03	1.31
2^{-4}	2.41e-01	6.55e-02	1.56e-02	3.61e-03	8.52e-04	2.05
0	2.38e-01	5.85e-02	1.15e-02	2.18e-03	4.23e-04	2.30

Table 6.8: The relative L^2-error in pressure. No stabilization is used.

In Subsection 5.1.1 we got good results for $P1-P1$ elements without any stabilization. This deserves further investigation: We run the same test problem for $P1-P1$ elements without CIP or PSPG stabilization (ghost-penalties are used on the cut elements). The results are displayed in Table 6.7 and 6.8 for velocity and pressure, respectively. From these tables we see the same behavior as for PSPG and CIP stabilization. The only difference is that the error for the pressure in the Stokes case is slightly higher.

Table 6.7 and 6.8 suggest that our test problem might be stable with respect to the inf-sup condition. We will investigate this further in Chapter 8. Comparing two stabilization methods on a test problem that might be stable, is not useful. From now on we only consider the CIP stabilization throughout the chapter, since we do not see any significant difference in the tables presented in this section.

6.3.1 Robustness test

Our fictitious domain is a square with lenght $1+2\delta$. This mesh consists of $2\times N\times N$ right triangles. The hypotenus of the triangles are $\sqrt{2}\,a$, where a is the adjacent, $a=(1+2\delta)/N$. The results presented in Table 6.1 – 6.8 used $\delta=1.1\cdot10^{-3}$. This δ value gives “nice” cut elements for N between 8 and 128. By “nice” cut elements we mean cut elements where a large percentage of the area of the triangle is in Ω. We want to construct “bad” cut elements to test the robustness of the method. Figure 5.4 illustrates such “bad” cut elements. We create “bad” cut elements by setting

$$\delta=\frac{9}{10}a \quad\Leftrightarrow\quad \delta=\frac{9}{10N-2\cdot9}. \tag{6.7}$$

$\epsilon\backslash N$	8	16	32	64	128	rate
1	1.77e-01	3.77e-02	8.55e-03	2.14e-03	6.46e-04	1.88
2^{-2}	5.40e-01	1.95e-01	6.57e-02	1.93e-02	6.94e-03	1.47
2^{-4}	5.96e-01	2.04e-01	7.66e-02	3.61e-02	2.28e-02	1.11
2^{-8}	6.05e-01	1.56e-01	4.54e-02	1.70e-02	6.66e-03	1.50
0	6.29e-01	1.70e-01	4.82e-02	1.52e-02	5.18e-03	1.60

Table 6.9: The relative L^2-error in velocity for “bad” cut elements. CIP stabilization was used.

$\epsilon\backslash N$	8	16	32	64	128	rate
1	4.96e-01	2.51e-01	1.15e-01	5.83e-02	3.63e-02	0.89
2^{-2}	4.13e-01	1.55e-01	9.50e-02	5.77e-02	3.69e-02	0.78
2^{-4}	3.77e-01	7.51e-02	1.94e-02	1.72e-02	2.13e-02	0.98
2^{-8}	2.88e-01	5.11e-02	1.17e-02	3.54e-03	1.41e-03	1.78
0	2.82e-01	4.83e-02	1.08e-02	3.31e-03	1.29e-03	1.80

Table 6.10: The relative L^2-error in pressure for “bad” cut- lements. CIP stabilization was used.

Results for this scaling with CIP stabilization are shown in Table 6.9 and 6.10. The convergence rates are slightly less then expected. We have good convergence rate for low N, but for N between 64 and 128 it slows down. Again observe see that the error and convergence rate for $\epsilon = 2^{-4}$ is a bit low.

Remark: If we measure the error on the entire background mesh we get higher errors since the values at the nodes outside of the physical domain are zero. The elements on the boundary were removed to measure the error on the physical domain only. The consequence is that we neglect some of the errors close to the boundary. However, only a small portion of the “bad” cut elements is in the physical domain.

Chapter 7

A priori error estimate

In this chapter we show an a priori error estimate for the CIP stabilized Brinkman equation with Nitsche's method on a fitted mesh. We consider the solution space $V = \left[L^2(\Omega) \cap \epsilon H^1(\Omega)\right]^d$ for the velocity and $Q = \left(H^1(\Omega) \cap L_0^2(\Omega)\right) + \epsilon^{-1} L_0^2(\Omega)$ for the pressure. We recall the problem:

Problem 7.1. *Find* $\mathbf{u}_h \in V_h$ *and* $p_h \in Q_h$ *such that*

$$A\left((\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)\right) = L(\mathbf{v}_h, q_h) \quad \forall \, \mathbf{v}_h \in V_h, \; q_h \in Q_h, \tag{7.1}$$

where

$$\begin{aligned}
A\left((\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)\right) &= a(\mathbf{u}_h, \mathbf{v}_h) + b(p_h, \mathbf{v}_h) + b(\mathbf{u}_h, q_h) - c(p_h, q_h), & (7.2)\\
L(\mathbf{v}_h, q_h) &= (\mathbf{f}, \mathbf{v}_h)_\Omega - \epsilon^2 (\mathbf{g}, \partial_{\mathbf{n}} \mathbf{v}_h)_\Gamma + (\mathbf{g} \cdot \mathbf{n}, q_h)_\Gamma + \gamma \epsilon^2 \left(h^{-1} \mathbf{g}, \mathbf{v}_h\right)_\Gamma, & (7.3)\\
a(\mathbf{u}_h, \mathbf{v}_h) &= (\mathbf{u}_h, \mathbf{v}_h)_\Omega + \epsilon^2 (\nabla \mathbf{u}_h, \nabla \mathbf{v}_h)_\Omega, & (7.4)\\
&\quad - \epsilon^2 (\partial_{\mathbf{n}} \mathbf{u}_h, \mathbf{v}_h)_\Gamma - \epsilon^2 (\mathbf{u}_h, \partial_{\mathbf{n}} \mathbf{v}_h)_\Gamma + \gamma \epsilon^2 \left(h^{-1} \mathbf{u}_h, \mathbf{v}_h\right)_\Gamma, & (7.5)\\
b(p_h, \mathbf{v}_h) &= \begin{cases} -(p_h, \nabla \cdot \mathbf{v}_h)_\Omega + (p_h, \mathbf{v}_h \cdot \mathbf{n}) & \textit{for } \epsilon > 0, \\ (\nabla p_h, \mathbf{v}_h)_\Omega & \textit{for } \epsilon = 0, \end{cases} & (7.6)\\
c(p_h, q_h) &= \beta_s \sum_{T \in \mathcal{T}_h} \frac{h_T^3}{\epsilon^2 + h_T^2} \left([\partial_{\mathbf{n}} p_h], [\partial_{\mathbf{n}} q_h]\right)_{\partial T}. & (7.7)
\end{aligned}$$

V_h is the space of continuous, piecewise linear $\mathbb{R}^d$- valued vector fields defined relative to a standard conforming tessellation $\mathcal{T}$ of Ω, Q_h is the space of continuous, piecewise linear functions defined relative to $\mathcal{T}$.

Note that $b(p_h, \mathbf{v}_h)$ is different from the one in Problem 5.1. This is because of the solution space we use. We now state both the continues norms and the discrete norms:

$$\|\mathbf{v}\|_\epsilon^2 := \|\mathbf{v}\|_\Omega^2 + \epsilon^2\|\nabla\mathbf{v}\|_\Omega^2 + \epsilon^2\gamma\|h^{-\frac{1}{2}}\mathbf{v}\|_\Gamma^2 + \epsilon^2\|h^{\frac{1}{2}}\partial_{\mathbf{n}}\mathbf{v}\|_\Gamma^2, \tag{7.8}$$

$$\|\mathbf{v}_h\|_{\epsilon,h}^2 := \|\mathbf{v}_h\|_\Omega^2 + \epsilon^2\|\nabla\mathbf{v}_h\|_\Omega^2 + \epsilon^2\gamma\|h^{-\frac{1}{2}}\mathbf{v}_h\|_\Gamma^2, \tag{7.9}$$

$$|||q|||_\epsilon := \sup_{\mathbf{v}\in V} \frac{b\,(q,\mathbf{v})}{\|\mathbf{v}\|_{\epsilon,h}}, \tag{7.10}$$

$$|||q_h|||_{\epsilon,h}^2 := \sum_{T\in\mathcal{T}} \frac{h_T^2}{\epsilon^2 + h_T^2}\|\nabla q_h\|_T^2, \tag{7.11}$$

$$\|\,(\mathbf{v}_h, q_h)\,\|_\epsilon^2 := \|\mathbf{v}_h\|_\epsilon^2 + |||q_h|||_\epsilon^2, \tag{7.12}$$

$$\|\,(\mathbf{v}_h, q_h)\,\|_{\epsilon,h}^2 := \|\mathbf{v}_h\|_{\epsilon,h}^2 + |||q_h|||_{\epsilon,h}^2. \tag{7.13}$$

If a norm $\|\cdot\|$ has no subscript, then it is the L^2-norm over the domain Ω. C (without any subscript) is a generic constant. We use the symbol $\lesssim$ to state the a is less or equal to b multiplied with a generic constant C; that is, $a \lesssim b \Leftrightarrow a \leq Cb$. We assume that the mesh is shape regular and $h_{max} \leq Ch_{min}$, where C is independent of h. We call this a quasi-uniform mesh.

The outline of this chapter is as follows: First, we state some ingredients needed to show the error estimate. Then we prove the four Brezzi's conditions. Finally, we prove the a priori error estimate.

7.1 Ingredients:

Lemma 7.1. ***Inverse estimates and trace inequality.***

Let h_F be the facet length on the element T. F is the facets of element T, then

$$\|h_F^{1/2}\partial_{\mathbf{n}}\mathbf{v}_h\|_F^2 \leq C_I\|\nabla\mathbf{v}_h\|_T^2, \tag{7.14}$$

$$\|\nabla\mathbf{v}_h\|_T \leq C_{I,1}\,h_T^{-1}\|\mathbf{v}_h\|_T, \tag{7.15}$$

$$\|\mathbf{v}_h\|_F \leq C_T\left(h_T^{-1/2}\|\mathbf{v}_h\|_T + h_T^{1/2}\|\nabla\mathbf{v}_h\|_T\right), \tag{7.16}$$

hold for all $\mathbf{v}_h \in V_h$, $T \in \mathcal{T}$ and $F \in \mathcal{F}$.

Proof. Proof can be found in [31] and [1]. □

Lemma 7.2.

$$\|\mathbf{v}_h\|_\epsilon \leq C_n\|\mathbf{v}_h\|_{\epsilon,h} \quad \forall\, \mathbf{v}_h \in V_h \tag{7.17}$$

Proof. We use the inverse estimate (7.14) and sum over all elements:

$$\|\mathbf{v}_h\|_\epsilon^2 = \|\mathbf{v}_h\|_{\epsilon,h}^2 + \epsilon^2\|h^{\frac{1}{2}}\partial_{\mathbf{n}}\mathbf{v}_h\|_\Gamma^2 \leq \|\mathbf{v}_h\|_{\epsilon,h}^2 + \epsilon^2 C_I\|\nabla x\mathbf{v}_h\|_\Omega^2 \leq C_n^2\|\mathbf{v}_h\|_{\epsilon,h}^2 \tag{7.18}$$

□

We use similar techniques as Burman and Hansbo [4] and use the Oswald interpolant to control the CIP term.

Lemma 7.3. ***Property of the Oswald interpolant.***

There exist an interpolation operator $\mathcal{I} : \left[H^2(\mathcal{T})\right]^d \longrightarrow [V_h]^d$, *such that for any piecewise constant function* $\phi \geq 0$, *the following hold,*

$$\|\phi^{\frac{1}{2}}\left(\nabla p_h - \mathcal{I}\left(\nabla p_h\right)\right)\|_\Omega^2 \leq C_c \sum_{T\in\mathcal{T}} h_T\phi\left(\left[\partial_{\mathbf{n}}p_h\right], \left[\partial_{\mathbf{n}}p_h\right]\right)_{\partial T\setminus\partial\Omega} \quad \forall\, \nabla p_h \in V_h \tag{7.19}$$

Proof. Proof can be found in both [32] and [33]. □

We set $\phi = \frac{h^2}{\epsilon^2+h^2}$ throughout this chapter.

Lemma 7.4. ***Scott–Zhang interpolants.***

There exist an interpolation operator $\pi_h : \left[H^1(\Omega)\right]^d \longrightarrow V_h$, *such that*

$$\sum_{T\in\mathcal{T}} h_T^{-2}\|\mathbf{v} - \pi_h\mathbf{v}\|_T^2 \leq C\|\nabla\mathbf{v}\|^2, \tag{7.20}$$

$$\|\pi_h\mathbf{v}\| \leq C\|\mathbf{v}\| \tag{7.21}$$

and

$$\|\nabla\pi_h\mathbf{v}\| \leq C\|\nabla\mathbf{v}\| \tag{7.22}$$

holds. There also exist an L^2*–projection operator* $\tilde{\pi}_h : \left[L^2(\Omega)\right]^d \longrightarrow V_h$, *such that*

$$\|\tilde{\pi}_h\left(\nabla p_h\right)\| \leq \|\nabla p_h\| \quad \forall\, p_h \in Q_h. \tag{7.23}$$

Proof. Proof can be found in [21] and [34]. □

Lemma 7.5. ***Interpolation error estimates.***

Let $\pi_h : \left[H^2(\Omega)\right]^d \longrightarrow Q_h$ *be a Scott–Zhang interpolation operator, then the following error estimates hold.*

$$\|q - \pi_h q\|_{r,T} \leq Ch^{s-r}|q|_{s,\omega(T)}, \qquad 0 \leq r \leq s \leq 2 \quad \forall\, T \in \mathcal{T}, \tag{7.24}$$

$$\|q - \pi_h q\|_{r,F} \leq Ch^{s-r-1/2}|q|_{s,\omega(T)}, \qquad 0 \leq r \leq s \leq 2 \quad \forall\, F \in \partial_i\mathcal{T}, \tag{7.25}$$

where $\omega(T)$ *is the patch of neighbors of element* T*; that is, the union of elements sharing a vertex with* T*.*

Proof. Proof can be found in [35]. □

Theorem 7.1. ***Interpolation error estimate for the ϵ-norm.***

Let π_h be the interpolation operator in Lemma 7.5, there is a constants $C > 0$ such that for all $\mathbf{v} \in \left[H^2(\Omega)\right]^d$ and all $q \in H^2(\Omega)$:

$$\|\mathbf{v} - \pi_h\mathbf{v}\|_\epsilon \leq Ch|\mathbf{v}|_{2,\Omega}, \tag{7.26}$$

$$|||q - \pi_h q|||_{\epsilon,h} \leq C\phi^{\frac{1}{2}}h|q|_{2,\Omega}, \tag{7.27}$$

$$|||q - \pi_h q|||_{\epsilon} \leq Ch|q|_{2,\Omega}, \tag{7.28}$$

$$\|\left(\mathbf{v} - \pi_h\mathbf{v}, q - \pi_h q\right)\|_\epsilon \leq Ch\left(|\mathbf{v}|_{2,\Omega} + |q|_{2,\Omega}\right). \tag{7.29}$$

Proof. First note that from Equation (7.24), we get

$$\|q - \pi_h q\|_{r,\Omega} \leq Ch^{s-r}|q|_{s,\Omega}, \tag{7.30}$$

by letting C contain the number of maximum neighboring elements to any $T \in \mathcal{T}$. We now prove Equation (7.26) term by term using Equation (7.30).

$$\begin{aligned}
\|\mathbf{v} - \pi_h\mathbf{v}\| &\leq Ch^2|\mathbf{v}|_{2,\Omega} && \leq Ch|\mathbf{v}|_{2,\Omega}, \\
\epsilon\|\nabla\left(\mathbf{v} - \pi_h\mathbf{v}\right)\| &\leq \epsilon Ch|\mathbf{v}|_{2,\Omega} && \leq Ch|\mathbf{v}|_{2,\Omega}, \\
\epsilon\sqrt{\gamma}\|h^{-\frac{1}{2}}\left(\mathbf{v} - \pi_h\mathbf{v}\right)\|_\Gamma &\leq \epsilon\sqrt{\gamma}C_T\left(h^{-1}\|\mathbf{v} - \pi_h\mathbf{v}\| + \|\nabla\left(\mathbf{v} - \pi_h\mathbf{v}\right)\|\right) && \leq Ch|\mathbf{v}|_{2,\Omega}, \\
\epsilon\|h^{\frac{1}{2}}\partial_\mathbf{n}\left(\mathbf{v} - \pi_h\mathbf{v}\right)\|_\Gamma &\leq \epsilon\|h^{\frac{1}{2}}\underbrace{\nabla\left(\mathbf{v} - \pi_h\mathbf{v}\right)}_{:=\mathbf{w}}\|_\Gamma \leq \epsilon C\left(\|\mathbf{w}\| + h\|\nabla\mathbf{w}\|\right) && \leq Ch|\mathbf{v}|_{2,\Omega}.
\end{aligned}$$

We used the trace inequality (7.16) for the two last inequality.

Equation (7.27) follow from applying Equation (7.30). We show Equation (7.28). Since $q \in H^2(\Omega)$, we can write $b\left(q - \pi_h q, \mathbf{v}\right) = \left(\nabla\left(q - \pi_h q\right), \mathbf{v}\right)_\Omega$. Using Cauchy–Schwarz's inequality (A.9) we get,

$$|||q - \pi_h q|||_\epsilon = \sup_{\mathbf{v}\in V}\frac{\left(\nabla\left(q - \pi_h q\right), \mathbf{v}\right)_\Omega}{\|\mathbf{v}\|_{\epsilon,h}} \leq \|\nabla\left(q - \pi_h q\right)\|\sup_{\mathbf{v}\in V}\frac{\|\mathbf{v}\|}{\|\mathbf{v}\|_{\epsilon,h}} \leq Ch|q|_{2,\Omega}. \tag{7.31}$$

For the last inequality we used Equation (7.30) and the fact that $\|\mathbf{v}\|/\|\mathbf{v}\|_{\epsilon,h} \leq 1$. Equation (7.29) follows from Equation (7.28) and (7.26). □

Lemma 7.6. ***Special case of Theorem 7.1***

$$\|\mathbf{v} - \pi_h\mathbf{v}\|_\epsilon \leq Ch|\mathbf{v}|_{1,\Omega}, \quad \textit{for } \epsilon = 0, \tag{7.32}$$

$$|||q - \pi_h q|||_{\epsilon,h} \leq Ch|q|_{1,\Omega}, \quad \textit{for } \epsilon = 1, \tag{7.33}$$

$$|||q - \pi_h q|||_{\epsilon} \leq Ch|q|_{1,\Omega}, \quad \textit{for } \epsilon = 1. \tag{7.34}$$

Proof. Equation (7.32) follows from that fact that $\|\mathbf{v}\|_\epsilon = \|\mathbf{v}\|$ for $\epsilon = 0$ and Equation (7.33) since $\phi^{\frac{1}{2}} = Ch$ for $\epsilon = 1$. The last inequality can be shown by applying Cauchy–Schwarz's inequality and Equation (7.24) and (7.25),

$$|||q - \pi_h q|||_\epsilon := \sup_{\mathbf{v}\in V} \frac{-\left((q-\pi_h q), \nabla\cdot\mathbf{v}\right)_\Omega + \left((q-\pi_h q), \mathbf{v}\cdot\mathbf{n}\right)_\Gamma}{\|\mathbf{v}\|_{\epsilon,h}}, \tag{7.35}$$

$$\leq \sup_{\mathbf{v}\in V} \frac{\|q-\pi_h q\|\|\nabla\mathbf{v}\| + \frac{1}{\gamma}h^{\frac{1}{2}}\|q-\pi_h q\|_\Gamma\|\gamma h^{-\frac{1}{2}}\mathbf{v}\|_\Gamma}{\|\mathbf{v}\|_{\epsilon,h}}, \tag{7.36}$$

$$\leq Ch|q|_{1,\Omega} \sup_{\mathbf{v}\in V} \frac{\|\nabla\mathbf{v}\| + \|\gamma h^{-\frac{1}{2}}\mathbf{v}\|_\Gamma}{\|\mathbf{v}\|_{\epsilon,h}} \leq Ch|q|_{1,\Omega}. \tag{7.37}$$

□

Lemma 7.7.

$$\sum_{T\in\mathcal{T}} \left(\frac{\epsilon^2 + h_T^2}{h_T^2}\right)^{\frac{1}{2}} \|\pi_h\mathbf{v} - \mathbf{v}\|_T \leq C_z\|\mathbf{v}\|_\epsilon \tag{7.38}$$

Proof. Follows from Lemma 7.4.

$$\sum_{T\in\mathcal{T}} \frac{\epsilon^2 + h_T^2}{h_T^2}\|\pi_h\mathbf{v} - \mathbf{v}\|_T^2 = \sum_{T\in\mathcal{T}} \left(\frac{\epsilon^2}{h_T^2} + 1\right)\|\pi_h\mathbf{v} - \mathbf{v}\|_T^2, \tag{7.39}$$

$$\leq C\left(\epsilon^2\|\nabla\mathbf{v}\|^2 + \|\pi_h\mathbf{v} - \mathbf{v}\|^2\right), \tag{7.40}$$

$$\leq C\left(\epsilon^2\|\nabla\mathbf{v}\|^2 + \|\mathbf{v}\|^2\right), \tag{7.41}$$

$$\leq C_z\|\mathbf{v}\|_\epsilon^2. \tag{7.42}$$

□

Lemma 7.8. ***Galerkin orthogonality.***

Let $(\mathbf{u}, p) \in V\times Q$ *be solutions of the Brinkman equation in Problem 4.3 and let* $(\mathbf{u}_h, p_h) \in V_h \times Q_h$ *be the discrete solution of Problem 7.1. Then*

$$A\left((\mathbf{u} - \mathbf{u}_h, p - p_h), (\mathbf{v}_h, q_h)\right) = 0 \quad \forall\ (\mathbf{v}_h, q_h) \in V_h \times Q_h. \tag{7.43}$$

Proof. Follows from the fact that $(\mathbf{u}, p)$ satisfy

$$A\left((\mathbf{u}, p), (\mathbf{v}_h, q_h)\right) = L\left(\mathbf{v}_h, q_h\right) \tag{7.44}$$

and then subtracting

$$A\left((\mathbf{u}_h, p_h), (\mathbf{v}_h, q_h)\right) = L\left(\mathbf{v}_h, q_h\right). \tag{7.45}$$

□

7.2 Brezzi's conditions

Theorem 7.2. ***Boundedness of*** $a(\cdot,\cdot)$.

$$a(\mathbf{u},\mathbf{v}_h) \leq C_1\|\mathbf{u}\|_\epsilon\|\mathbf{v}_h\|_{\epsilon,h} \quad \forall\,\mathbf{u}\in H^2 \textit{ and } \mathbf{v}_h\in V_h. \tag{7.46}$$

Proof. The proof can be found in [36], it follows from Cauchy–Schwarz's inequality (A.9) and (A.10), and the inverse estimate (7.15):

$$\begin{aligned} a(\mathbf{u},\mathbf{v}_h) &= (\mathbf{u},\mathbf{v}_h)_\Omega + \epsilon^2(\nabla\mathbf{u},\nabla\mathbf{v}_h)_\Omega + \epsilon^2\gamma\left(h^{-1}\mathbf{u},\mathbf{v}_h\right)_\Gamma \\ &\quad - \epsilon^2(\partial_\mathbf{n}\mathbf{u},\mathbf{v}_h)_\Gamma - \epsilon^2(\partial_\mathbf{n}\mathbf{v}_h,\mathbf{u})_\Gamma\,, \\ &\leq \|\mathbf{u}\|\|\mathbf{v}_h\| + \epsilon^2\|\nabla\mathbf{u}\|\|\nabla\mathbf{v}_h\| + \epsilon^2\gamma\|h^{-\frac{1}{2}}\mathbf{u}\|_\Gamma\|h^{-\frac{1}{2}}\mathbf{v}_h\|_\Gamma \\ &\quad + \epsilon^2\|h^{\frac{1}{2}}\partial_\mathbf{n}\mathbf{u}\|_\Gamma\|h^{-\frac{1}{2}}\mathbf{v}_h\|_\Gamma + \epsilon^2\|h^{\frac{1}{2}}\partial_\mathbf{n}\mathbf{v}_h\|_\Gamma\|h^{-\frac{1}{2}}\mathbf{u}\|_\Gamma, \\ &\leq \|\mathbf{u}\|\|\mathbf{v}_h\| + \epsilon^2\|\nabla\mathbf{u}\|\|\nabla\mathbf{v}_h\| + \epsilon^2\gamma\|h^{-\frac{1}{2}}\mathbf{u}\|_\Gamma\|h^{-\frac{1}{2}}\mathbf{v}_h\|_\Gamma \\ &\quad + \epsilon^2\|h^{\frac{1}{2}}\partial_\mathbf{n}\mathbf{u}\|_\Gamma\|h^{-\frac{1}{2}}\mathbf{v}_h\|_\Gamma + \epsilon^2\sqrt{C_I}\|\nabla\mathbf{v}_h\|\|h^{-\frac{1}{2}}\mathbf{u}\|_\Gamma, \\ &\leq C_1\|\mathbf{u}\|_\epsilon\|\mathbf{v}_h\|_{\epsilon,h}. \end{aligned} \tag{7.47}$$

We can also show a modified coercivity on the form $a(\mathbf{u}_h,\mathbf{v}_h) \leq C_1\|\mathbf{u}_h\|_\epsilon\|\mathbf{v}_h\|_\epsilon$ for all $\mathbf{u}_h$, $\mathbf{v}_h\in V_h$. □

Theorem 7.3. ***Boundedness of*** $b(\cdot,\cdot)$.

$$b(q,\mathbf{v}) \leq C_2\|\mathbf{v}\|_{\epsilon,h}|||q|||_\epsilon \quad \forall\,\mathbf{v}\in V,\ q\in Q \tag{7.48}$$

Proof. Follows from the definition in Equation (7.10). □

Theorem 7.4. ***Modified coercivity of*** $b(\cdot,\cdot)$.

Assuming quasi uniform mesh, the following holds.

$$\sup_{\mathbf{v}_h\in V_h}\frac{b(p_h,\mathbf{v}_h)}{\|\mathbf{v}_h\|_{\epsilon,h}} + (c(p_h,p_h))^{\frac{1}{2}} \geq C_4|||p_h|||_{\epsilon,h} \quad \forall\,p_h\in Q_h \tag{7.49}$$

Proof. First, we note that the bilinear form $b(p_h,\mathbf{v}_h)$, Equation (7.6), for $\epsilon > 0$ and $\epsilon = 0$ are equivalent for the discrete case for P_1 elements. We therefore consider only $b(p_h,\mathbf{v}_h) = (\mathbf{v}_h,\nabla p_h)_\Omega$. We use the same $\mathbf{v}_h$ as in the proof of Lemma 7.57,

$$\mathbf{v}_h = \phi\,\tilde{\pi}_h(\nabla p_h) \text{ and } \phi = \frac{h^2}{\epsilon^2+h^2}, \tag{7.50}$$

where $\tilde{\pi}_h$ is the projection in Lemma 7.4.

$$|||p_h|||_{\epsilon,h}^2 = \left(\phi\,\tilde{\pi}_h\left(\nabla p_h\right), \nabla p_h\right)_\Omega + \left(\phi\,\left(\nabla p_h - \tilde{\pi}_h\left(\nabla p_h\right)\right), \nabla p_h\right)_\Omega, \tag{7.51}$$

$$= b\left(p_h, \mathbf{v}_h\right) + \left(\phi^{\frac{1}{2}}\left(\nabla p_h - \tilde{\pi}_h\left(\nabla p_h\right)\right), \phi^{\frac{1}{2}}\left(\nabla p_h - \mathcal{I}\left(\nabla p_h\right)\right)\right)_\Omega . \tag{7.52}$$

Here, $\mathcal{I}\left(\nabla p_h\right)$ denotes the Oswald interpolation from Lemma 7.3. The last equality come from the fact that $\mathcal{I}\left(\nabla p_h\right) \in V_h$ and that $\tilde{\pi}_h$ is a L^2–projection. Further, we use Cauchy–Schwarz's inequality (A.9) and Lemma 7.3.

$$|||p_h|||_{\epsilon,h}^2 \leq b\left(p_h, \mathbf{v}_h\right) + \phi^{\frac{1}{2}}\|\nabla p_h - \tilde{\pi}_h\left(\nabla p_h\right)\| C_c\ \left(c\left(p_h, p_h\right)\right)^{\frac{1}{2}}, \tag{7.53}$$

$$\leq b\left(p_h, \mathbf{v}_h\right) + C_c\,\phi^{\frac{1}{2}}\left(\|\nabla p_h\| + \|\tilde{\pi}_h\left(\nabla p_h\right)\|\right)\left(c\left(p_h, p_h\right)\right)^{\frac{1}{2}}, \tag{7.54}$$

$$\leq b\left(p_h, \mathbf{v}_h\right) + 2C_c\,\phi^{\frac{1}{2}}\|\nabla p_h\|\ \left(c\left(p_h, p_h\right)\right)^{\frac{1}{2}}, \tag{7.55}$$

$$= b\left(p_h, \mathbf{v}_h\right) + 2C_c|||p_h|||_{\epsilon,h}\ \left(c\left(p_h, p_h\right)\right)^{\frac{1}{2}} \tag{7.56}$$

For the last inequality we used Lemma 7.4. The proof is completed by dividing Equation (7.56) by $|||p_h|||_{\epsilon,h}$ and showing that

$$\|\mathbf{v}_h\|_{\epsilon,h} \leq C|||p_h|||_{\epsilon,h}. \tag{7.57}$$

We show this term by term for $\|\mathbf{v}_h\|_{\epsilon,h}$:

$$\|\mathbf{v}_h\| = \|\phi\,\tilde{\pi}_h\left(\nabla p_h\right)\| \lesssim \phi\|\nabla p_h\| = \phi^{\frac{1}{2}}|||p_h|||_{\epsilon,h} \lesssim |||p_h|||_{\epsilon,h} \tag{7.58}$$

The first inequality comes from Lemma 7.4, the second from the fact that $\phi^{\frac{1}{2}} \leq 1$.

$$\epsilon\|\nabla\mathbf{v}_h\| = \epsilon\|\phi\,\nabla\left(\tilde{\pi}_h\left(\nabla p_h\right)\right)\| \leq C_{I,1}\frac{\epsilon}{h}\phi\|\tilde{\pi}_h\left(\nabla p_h\right)\| \leq C_{I,1}\frac{\epsilon}{h}\phi^{\frac{1}{2}}|||p_h|||_{\epsilon,h} \lesssim |||p_h|||_{\epsilon,h} \tag{7.59}$$

The first inequality comes from the inverse estimate (7.15), the second inequality we used Lemma 7.4 and the last inequality comes from the fact that $\frac{\epsilon}{h}\phi^{\frac{1}{2}} < 1$.

$$\epsilon\|h^{-\frac{1}{2}}\mathbf{v}_h\|_\Gamma \leq C_T\left(\frac{\epsilon}{h}\|\mathbf{v}_h\| + \epsilon\|\nabla\mathbf{v}_h\|\right) \lesssim |||p_h|||_{\epsilon,h} \tag{7.60}$$

The first inequality comes from the trace inequality (7.16). For the first term on the second inequality we use Equation (7.58) and the fact that $\frac{\epsilon}{h}\phi^{\frac{1}{2}} < 1$. For the second term on the second inequality we use Equation (7.59). □

Theorem 7.5. ***Coercivity of*** $a\left(\cdot,\cdot\right)$***.***

$$a\left(\mathbf{u}_h, \mathbf{u}_h\right) \geq C_3\|\mathbf{u}_h\|_{\epsilon,h}^2 \quad \forall\,\mathbf{u}_h \in V_h. \tag{7.61}$$

Proof. The proof can be found in both [36] and [37]. We also prove it in this thesis:

$$\begin{aligned} a\left(\mathbf{u}_h, \mathbf{u}_h\right) &= \|\mathbf{u}_h\|^2 + \epsilon^2\|\nabla \mathbf{u}_h\|^2 - 2\epsilon^2 \left(\partial_{\mathbf{n}}\mathbf{u}_h, \mathbf{u}_h\right)_\Gamma + \epsilon^2\gamma\|h^{-\frac{1}{2}}\mathbf{u}_h\|_\Gamma^2, \\ &\geq \|\mathbf{u}_h\|^2 + \epsilon^2\|\nabla \mathbf{u}_h\|^2 - \frac{\epsilon^2}{\delta}\|h^{\frac{1}{2}}\partial_{\mathbf{n}}\mathbf{u}_h\|_\Gamma^2 + \epsilon^2\left(\gamma - \delta\right)\|h^{-\frac{1}{2}}\mathbf{u}_h\|_\Gamma^2, \\ &\geq \|\mathbf{u}_h\|^2 + \epsilon^2\|\nabla \mathbf{u}_h\|^2 - \epsilon^2\frac{C_I}{\delta}\|\nabla \mathbf{u}_h\|^2 + \epsilon^2\left(\gamma - \delta\right)\|h^{-\frac{1}{2}}\mathbf{u}_h\|_\Gamma^2. \end{aligned}$$

To obtain the first inequality, we used Cauchy's inequality with δ (A.8). The second inequality comes from the inverse estimate 7.1 with similar summation trick as in Theorem 7.2. $\frac{C_I}{\delta}$ must be less then 1, thus we choose δ such that $\delta > C_I$. Also γ must be chosen such that $\gamma > \delta > C_I$. □

Note that C_3 is dependent on C_I as mentioned in Section 5.2.

Theorem 7.6. ***The inf-sup stability of*** $A\left(\cdot, \cdot\right)$***.***

There is a constant $C > 0$ *such that*

$$\sup_{(\mathbf{v}_h, q_h) \in V_h \times Q_h} \frac{A\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right)}{\|\left(\mathbf{v}_h, q_h\right)\|_{\epsilon,h}} > C\|\left(\mathbf{u}_h, p_h\right)\|_{\epsilon,h} \quad \forall\left(\mathbf{u}_h, p_h\right) \in V_h \times Q_h \tag{7.62}$$

Proof.

Part 1: Let $(\mathbf{v}_h, q_h) = (\mathbf{u}_h, -p_h)$, then we obtain

$$A\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{u}_h, -p_h\right)\right) = a\left(\mathbf{u}_h, \mathbf{u}_h\right) + b\left(p_h, \mathbf{u}_h\right) + b\left(-p_h, \mathbf{u}_h\right) + c\left(p_h, p_h\right), \tag{7.63}$$

$$\geq C_3\|\mathbf{u}_h\|_{\epsilon,h}^2 + c\left(p_h, p_h\right), \tag{7.64}$$

from the coercivity Theorem 7.5.

Part 2: Let $(\mathbf{v}_h, q_h) = \left(\phi\,\tilde{\pi}_h\left(\nabla p_h\right), 0\right)$, where $\tilde{\pi}_h$ is the projection in Lemma 7.4. From Theorem 7.4 we have,

$$b\left(p_h, \pi_h\phi\,\tilde{\pi_h}\left(\nabla p_h\right)\right) \geq C_4\;|||p_h|||_{\epsilon,h}\|\mathbf{v}_h\|_{\epsilon,h} - \left(c\left(p_h, p_h\right)\right)^{\frac{1}{2}}\|\mathbf{v}_h\|_{\epsilon,h}, \tag{7.65}$$

$$\geq C_4\;|||p_h|||_{\epsilon,h}\|\mathbf{v}_h\|_{\epsilon,h} - \delta\|\mathbf{v}_h\|_{\epsilon,h}^2 - \frac{1}{4\delta}c\left(p_h, p_h\right). \tag{7.66}$$

In the last inequality we used Cauchy–Schwarz's inequality with δ (A.8). Using inequality 7.66 and recalling that $v_h = \phi\, \tilde{\pi_h}\left(\nabla p_h\right)$, we get

$$
\begin{aligned}
A\left(\left(\mathbf{u}_h, p_h\right), \left(\phi\, \tilde{\pi_h}\left(\nabla p_h\right), 0\right)\right) &= a\left(\mathbf{u}_h, \mathbf{v}_h\right) + b\left(p_h, \mathbf{v}_h\right), && (7.67)\\
&\geq -C_1 \|\mathbf{u}_h\|_\epsilon \|\mathbf{v}_h\|_\epsilon + C_4 |||p_h|||_{\epsilon,h} \|\mathbf{v}_h\|_{\epsilon,h} && (7.68)\\
&\quad - \delta \|\mathbf{v}_h\|^2_{\epsilon,h} - \frac{1}{4\delta} c\left(p_h, p_h\right), && (7.69)\\
&\geq -\frac{C_1}{4\tilde{\delta}} \|\mathbf{u}_h\|^2_\epsilon - \tilde{\delta} C_1 \|\mathbf{v}_h\|^2_\epsilon + C_4 |||p_h|||_{\epsilon,h} \|\mathbf{v}_h\|_{\epsilon,h} && (7.70)\\
&\quad - \delta \|\mathbf{v}_h\|^2_{\epsilon,h} - \frac{1}{4\delta} c\left(p_h, p_h\right). && (7.71)
\end{aligned}
$$

We rescale $\|\mathbf{v}_h\|_{\epsilon,h} := |||p_h|||_{\epsilon,h}$, use Lemma 7.2 and chose δ and $\tilde{\delta}$ such that $C_\delta := C_4 - C_1 C_n^2 \tilde{\delta} - \delta$ is positive.

$$
\begin{aligned}
A\left(\left(\mathbf{u}_h, p_h\right), \left(\phi\, \tilde{\pi_h}\left(\nabla p_h\right), 0\right)\right) &= a\left(\mathbf{u}_h, \mathbf{v}_h\right) + b\left(p_h, \mathbf{v}_h\right), && (7.72)\\
&\geq C_\delta |||p_h|||_{\epsilon,h} \|\mathbf{v}_h\|_{\epsilon,h} - \frac{C_1 C_n^2}{4\tilde{\delta}} \|\mathbf{u}_h\|^2_{\epsilon,h} - \frac{1}{4\delta} c\left(p_h, p_h\right). && (7.73)
\end{aligned}
$$

Part 3: Finally we combine part 1 and 2, $\left(\mathbf{v}_h, q_h\right) = \left(\mathbf{u}_h, -p_h\right) + \alpha\left(\phi\, \tilde{\pi_h}\left(\nabla p_h\right), 0\right)$ for a sufficiently small α and use inequality (7.64) and (7.73).

$$
\begin{aligned}
A\left(\left(\mathbf{u}_h, p_h\right), \left(\mathbf{v}_h, q_h\right)\right) &\geq C_3 \|\mathbf{u}_h\|^2_{\epsilon,h} + c\left(p_h, p_h\right) && (7.74)\\
&\quad + \alpha\left(C_\delta |||p_h|||_{\epsilon,h} \|\mathbf{v}_h\|_{\epsilon,h} - \frac{C_1 C_n^2}{4\tilde{\delta}} \|\mathbf{u}_h\|^2_{\epsilon,h} - \frac{1}{4\delta} c\left(p_h, p_h\right)\right), && (7.75)\\
&\geq \left(C_3 - \alpha \frac{C_1 C_n^2}{4\tilde{\delta}}\right) \|\mathbf{u}_h\|^2_{\epsilon,h} + \alpha C_\delta |||p_h|||^2_{\epsilon,h}, && (7.76)\\
&\geq \min\left(\left(C_3 - \alpha \frac{C_1 C_n^2}{4\tilde{\delta}}\right), \alpha C_\delta\right) \|\left(\mathbf{v}_h, q_h\right)\|^2_{\epsilon,h}, && (7.77)
\end{aligned}
$$

and the proof is complete. Note that α have to be chosen such that $C_3 - \alpha \frac{C_1 C_n^2}{4\tilde{\delta}}$ and $1 - \alpha \frac{1}{4\delta}$ are positive. □

7.3 A priori error estimate

Theorem 7.7. ***A priori error estimate.***

Let $(\mathbf{u}, p) \in \left[H^2(\Omega)\right]^d \times H^2(\Omega)$ *be the solution of the Brinkman Problem 4.3, and let* $(\mathbf{u}_h, p_h)$ *be the discrete solution of corresponding CIP stabilized Nitsche formulation, Problem 7.1. Then, there is a constant* $C > 0$ *such that*

$$\|\mathbf{u}-\mathbf{u}_h\|_\epsilon + |||p-p_h|||_{\epsilon,h} \leq Ch\left(|\mathbf{u}|_{2,\Omega} + |p|_{2,\Omega}\right). \tag{7.78}$$

Proof. Using the triangle inequality and Lemma 7.2 we obtain

$$\begin{aligned}\|\mathbf{u}-\mathbf{u}_h\|_\epsilon + |||p-p_h|||_{\epsilon,h} \leq{}& \|\mathbf{u}-\pi_h\mathbf{u}\|_\epsilon + |||p-\pi_h p|||_{\epsilon,h} \\ &+ C_n\|\pi_h\mathbf{u}-\mathbf{u}_h\|_{\epsilon,h} + |||\pi_h p - p_h|||_{\epsilon,h}.\end{aligned} \tag{7.79}$$

The desired bound for the first two terms on the right-hand side of Equation (7.79) are obtained by applying Theorem 7.1. We need to handle the last two terms on the right-hand side of Equation (7.79). By using Cauchy's inequality (A.7) and Theorem 7.6 we obtain

$$C_n\|\pi_h\mathbf{u}-\mathbf{u}_h\|_{\epsilon,h} + |||\pi_h p - p_h|||_{\epsilon,h} \leq \frac{A\left((\mathbf{u}_h-\pi_h\mathbf{u}, p_h-\pi_h p), (\mathbf{v}_h, q_h)\right)}{\|(\mathbf{v}_h, q_h)\|_{\epsilon,h}}, \tag{7.80}$$

for some $\mathbf{v}_h, p_h \in V_h \times Q_h$. Next, we use Galerkin orthogonality (Lemma 7.8),

$$A\left((\mathbf{u}_h-\pi_h\mathbf{u}, p_h-\pi_h p), (\mathbf{v}_h, q_h)\right) = A\left((\mathbf{u}-\pi_h\mathbf{u}, p-\pi_h p), (\mathbf{v}_h, q_h)\right). \tag{7.81}$$

From the definition of A, we have

$$\begin{aligned}A\left((\mathbf{u}-\pi_h\mathbf{u}, p-\pi_h p), (\mathbf{v}_h, q_h)\right) ={}& a\left(\mathbf{u}-\pi_h\mathbf{u}, \mathbf{v}_h\right) + b\left(\mathbf{u}-\pi_h\mathbf{u}, q_h\right) \\ &+ b\left(p-\pi_h p, \mathbf{v}_h\right) - c\left(p-\pi_h p, q_h\right).\end{aligned} \tag{7.82}$$

We handle these term separately. Using Theorem 7.2 we obtain

$$a\left(\mathbf{u}-\pi_h\mathbf{u}, \mathbf{v}_h\right) \leq C_1\|\mathbf{u}-\pi_h\mathbf{u}\|_\epsilon\|\mathbf{v}_h\|_{\epsilon,h} \leq C_1\|\mathbf{u}-\pi_h\mathbf{u}\|_\epsilon\|\mathbf{v}_h\|_{\epsilon,h}. \tag{7.83}$$

Using Cauchy–Schwarz's inequality,

$$b\left(\mathbf{u}-\pi_h\mathbf{u}, q_h\right) = \left(\phi^{-\frac{1}{2}}\left(\mathbf{u}-\pi_h\mathbf{u}\right), \phi^{\frac{1}{2}}\nabla q_h\right) \leq \sqrt{\frac{\epsilon^2+h^2}{h^2}}\|\mathbf{u}-\pi_h\mathbf{u}\| |||q_h|||_{\epsilon,h}. \tag{7.84}$$

Next, we use Theorem 7.3,

$$b\left(p-\pi_h p, \mathbf{v}_h\right) \leq C_2|||p-\pi_h p|||_\epsilon\|\mathbf{v}_h\|_{\epsilon,h}. \tag{7.85}$$

The CIP stabilization term can be estimated by combining the inverse estimate Equation (7.14), Equation (7.22), Equation (7.30), and the trace inequality Equation (7.16).

$$\begin{aligned} -c\left(p-\pi_h p, q_h\right) &= \beta_s \sum_{T\in\mathcal{T}} \frac{h^3}{\epsilon^2+h^2}\left(\left[\partial_{\mathbf{n}}\pi_h p\right], \left[\partial_{\mathbf{n}} q_h\right]\right)_{\partial T} \\ &\leq C\phi \sum_{T\in\mathcal{T}} \|h^{\frac{1}{2}}\partial_{\mathbf{n}}\left(\pi_h p - p\right)\|_{\partial T}\|h^{\frac{1}{2}}\partial_{\mathbf{n}} q_h\|_{\partial T} \\ &\leq C\phi\left(\|\left(\pi_h p-p\right)\| + h|p|_{2,\Omega}\right)\|\nabla q_h\| \leq C\phi^{\frac{1}{2}} h|p|_{2,\Omega}|||q_h|||_{\epsilon,h}. \end{aligned} \tag{7.86}$$

Now we use on Equation (7.26) on (7.83) and (7.84), where we gain an extra h since Equation (7.84) only contains the L^2-norm. We use Equation (7.28) on (7.85). Collecting the term, we get

$$\begin{aligned} A\left(\left(\mathbf{u}-\pi_h\mathbf{u}, p-\pi_h p\right), \left(\mathbf{v}_h, q_h\right)\right) \leq C \Big(& h|\mathbf{u}|_{2,\Omega}\|\mathbf{v}_h\|_{\epsilon,h} + h\sqrt{\epsilon^2+h^2}|\mathbf{u}|_{2,\Omega}|||q_h|||_{\epsilon,h} \\ & + h|p|_{2,\Omega}\|\mathbf{v}_h\|_{\epsilon,h} + \phi^{\frac{1}{2}} h|p|_{2,\Omega}|||q_h|||_{\epsilon,h}\Big). \end{aligned} \tag{7.87}$$

We apply the discrete Cauchy's inequality to the equation above to obtain

$$A\left(\left(\mathbf{u}-\pi_h\mathbf{u}, p-\pi_h p\right), \left(\mathbf{v}_h, q_h\right)\right) \leq Ch\left(|\mathbf{u}|_{2,\Omega} + |p|_{2,\Omega}\right)\|\left(\mathbf{v}_h, q_h\right)\|_{\epsilon,h}. \tag{7.88}$$

Combining Equation (7.80) and (7.88) completes the proof. □

Lemma 7.9. ***Special case of the a priori error estimate***

$$\|\mathbf{u}-\mathbf{u}_h\|_{\epsilon} + |||p-p_h|||_{\epsilon,h} \leq Ch\left(|\mathbf{u}|_{2,\Omega} + |p|_{1,\Omega}\right), \quad \textit{for } \epsilon = 1, \tag{7.89}$$

$$\|\mathbf{u}-\mathbf{u}_h\|_{\epsilon} + |||p-p_h|||_{\epsilon,h} \leq Ch\left(|\mathbf{u}|_{1,\Omega} + |p|_{2,\Omega}\right), \quad \textit{for } \epsilon = 0. \tag{7.90}$$

Proof. Can be shown by modifying the proof of Theorem 7.7 by applying Lemma 7.6. □

Chapter 8

Preconditioning

8.1 Iterative methods

The finite element method transforms a PDE into a system of equations,

$$A\mathbf{u} = \mathbf{b}. \tag{8.1}$$

The matrix $A \in \mathbb{R}^{N,N}$ is typically large and sparse (meaning that it contains only $\mathcal{O}(N)$ nonzero entries). Direct methods for solving the linear system, like Gaussian elimination or LU-factorization, require between $\mathcal{O}(N^2)$ and $\mathcal{O}(N^3)$ floating point operations and $\mathcal{O}(N^2)$ of storage. For realistic simulation N is large, typically in the range 10^6–10^9, the direct methods are insufficient. Instead, we use iterative methods with a good preconditioner. An iterative method (assuming that it convergent), solves a linear system in $\mathcal{O}(N)$ operations.

The iterative method we use is the Krylov subspace method called *Minimal Residual* (MinRes). It requires that the matrix A is symmetric, in contrast to the more common *Conjugated gradient method* which requires that A is both symmetric and positive definite.

The convergence of the iterative methods is related to the condition number of A. A high condition number gives a slow convergence rate. For discretized PDEs, the condition number grows as $1/h^2$; in other words, the condition number of A is not bounded. Hence the convergence might be extremely slow or it might not converge at all. To fix the problem we multiply Equation (8.1) with a preconditioner B and instead solve the system:

$$BA\mathbf{u} = B\mathbf{b}. \tag{8.2}$$

The preconditioner B should have the following properties:

1. The condition number of BA is bounded.

2. B only uses $\mathcal{O}(N)$ flops for evaluation.

3. B require only $\mathcal{O}(N)$ of storage.

If B is chosen to be A^{-1}, the first property is satisfied. However, the second and third property is not satisfied since A^{-1} is dense.

Finding good preconditioners is challenging. We explore some possible preconditioners for unstabilized and stabilized Brinkman problems in the view of operator preconditioning. The rest of this chapter follows the theory from [2, 38, 10] and [11]. We will focus only on the fitted case and simplify the notation by removing the domain Ω and the brackets to the spaces; that is, $H^1 = \left[H^1(\Omega)\right]^d$.

8.2 Operator preconditioning of the unstabilized Brinkman equation

We reformulate the (strong) Brinkman Equation (3.19) as a matrix equation,

$$\mathcal{A}_\epsilon \begin{pmatrix} \mathbf{u} \\ p \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0 \end{pmatrix}, \tag{8.3}$$

where $\mathcal{A}_\epsilon$ is the differential operator

$$\mathcal{A}_\epsilon = \begin{pmatrix} I - \epsilon^2\Delta & \text{grad} \\ \text{div} & 0 \end{pmatrix}. \tag{8.4}$$

For each fixed $\epsilon > 0$, the operator $\mathcal{A}_\epsilon$ is an isomorphism mapping $H^1 \times L^2_0$ onto $H^{-1} \times L^2_0$, where H^{-1} is the dual space of H^1 [38].

However, we would like to vary ϵ and have the operator norm bounded independently of ϵ. To achieve this we will need to introduce ϵ dependent spaces and norms.

As mentioned in the end of Section 4.2, the Darcy case has two possible solution spaces, namely $L^2_g \times H^1 \cap L^2_0$ and $H(\text{div}) \times L^2_0$. We have two possible solution spaces X_ϵ for the Brinkman Equation (8.3):

$$X_\epsilon = \left(L^2 \cap \epsilon H^1\right) \times \left(\left(H^1 \cap L^2_0\right) + \epsilon^{-1} L^2_0\right) \tag{8.5}$$

$$X_\epsilon = \left(H(\text{div}) \cap \epsilon H^1\right) \times L^2_0 \tag{8.6}$$

The corresponding dual spaces with respect to the L^2–inner product are[1]

$$X^*_\epsilon = \left(L^2 + \epsilon^{-1} H^{-1}\right) \times \left(H^{-1}_0 \cap \epsilon L^2_0\right), \tag{8.7}$$

$$X^*_\epsilon = \left(H^{-1}(\text{rot}) + \epsilon^{-1} H^{-1}\right) \times L^2_0, \tag{8.8}$$

[1] The asterisk "*" denotes the dual space and not the background mesh.

where H_0^{-1} corresponds to the dual space of $H^1 \cap L_0^2$ [38, 2].

In Section 6.2 we found that the solution space in Problem 4.5, 5.1 and 5.2 reduces to $H\left(\text{div}\right) \times L_0^2$ when $\epsilon \to 0$. However, for constructing preconditions we chose to use the solution space (8.5) and dual space (8.7).

Mardal et al. [38] show that $\mathcal{A}_\epsilon$ is an isomorphism mapping X_ϵ onto X_ϵ^*. We want an operator $\mathcal{B}_\epsilon$ that maps X_ϵ^* onto X_ϵ.

$$\mathcal{B}_\epsilon = \begin{pmatrix} \left(I - \epsilon^2 \Delta\right)^{-1} & 0 \\ 0 & \epsilon^2 I + \left(-\Delta\right)^{-1} \end{pmatrix} \tag{8.9}$$

has this property. Combining $\mathcal{B}_\epsilon$ and $\mathcal{A}_\epsilon$ gives:

$$\mathcal{B}_\epsilon \mathcal{A}_\epsilon : \quad X_\epsilon \xrightarrow{\mathcal{A}_\epsilon} X_\epsilon^* \xrightarrow{\mathcal{B}_\epsilon} X_\epsilon; \tag{8.10}$$

that is, $\mathcal{B}_\epsilon \mathcal{A}_\epsilon$ maps X_ϵ onto itself. Furthermore, the condition number $\kappa\left(\mathcal{B}_\epsilon \mathcal{A}_\epsilon\right)$ is bounded independently of ϵ.

Remark: If we instead want to use the solution space (8.6) with the corresponding dual space (8.8), the preconditioner can be chosen as

$$\mathcal{B}_\epsilon = \begin{pmatrix} \left(I - \text{grad div} - \epsilon^2 \Delta\right)^{-1} & 0 \\ 0 & I \end{pmatrix}, \tag{8.11}$$

which maps (8.8) onto (8.6) [11].

8.3 Discrete preconditioning of the unstabilized Brinkman equation

We need to transform the operators $\mathcal{A}_\epsilon$ and $\mathcal{B}_\epsilon$ into discrete matrices $\mathcal{A}_{\epsilon,h}$ and $\mathcal{B}_{\epsilon,h}$. For $\mathcal{A}_{\epsilon,h}$ we use the finite element formulation in Problem 4.5 or 5.1 (depending on if we want to use Nitsche method or not), for a suitable finite element without stabilization. This approach however, will be inefficient for $\mathcal{B}_{\epsilon,h}$ since the discrete operator $\mathcal{B}_{\epsilon,h}$ is dense. Instead, we use a matrix $\tilde{\mathcal{B}}_{\epsilon,h}$ which is spectral equivalent to $\mathcal{B}_{\epsilon,h}$ and cheap to both evaluate and store.

For the computations we use a multigrid preconditioner $\tilde{\mathcal{B}}_{\epsilon,h}$ which uses a standard V–cycle operator and four sweeps of symmetric Gauss–Seidel as smoother [39]. For the stop criteria we use $\left(\mathbf{r}_k^T A \mathbf{r}_k < \text{tolerance}\right)^{1/2}$, where $\mathbf{r}_k$ is the residual, $\mathbf{r}_k := \mathbf{b}_k - A\mathbf{u}_k$. We use the library cbc.block [40] for block operations and to construct preconditioners, and we use the software Octave [41] to find exact eigenvalues and condition numbers.

$\epsilon \backslash N$	4×4	8×8	16×16
1	15.8	17.3	17.8
2^{-2}	12.8	15.6	17.0
2^{-4}	6.85	10.2	13.5
2^{-8}	3.30	3.46	3.80
0	3.33	3.38	3.40

Table 8.1: Condition number for $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ for the *mini*–element (without Nitsche).

$\epsilon \backslash N$	4×4	8×8	16×16
1	1.14e+04	4.54e+04	1.82e+05
2^{-2}	9.22e+01	3.56e+02	1.41e+03
2^{-4}	3.30e+01	6.45e+01	2.21e+02
2^{-8}	3.41e+02	1.34e+03	3.75e+03
0	3.54e+02	1.57e+03	6.245+03

Table 8.2: Condition number for $\mathcal{A}_{\epsilon,h}$ for the *mini*–element (without Nitsche).

8.3.1 Condition number of $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ and $\mathcal{A}_{\epsilon,h}$

Definition 8.1. ***Condition number.***

The condition number of a matrix A is the largest absolute eigenvalue divided by the smallest absolute eigenvalue,

$$\kappa(A) = \frac{|\lambda_{max}|}{|\lambda_{min}|}. \tag{8.12}$$

We remove the zero eigenvalue that comes because the pressure is only defined uniquely up to a constant.

Table 8.1 displays the exact condition number for preconditioned system $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ for the *mini*–element without the Nitsche method for various ϵ and N. Table 8.2 displays the exact condition number for the matrix $\mathcal{A}_{\epsilon,h}$ for the *mini*–element without the Nitsche method. Table 8.1 shows that the condition number increases a bit, but seems to even out for high values of N, these results are similar to what [42] and [38] showed. Table 8.2 shows that the condition number grows rapidly as N increases. For the lower right entry of $\mathcal{B}_{\epsilon,h}$ we used $(I - \Delta)^{-1}$ instead of $(-\Delta)^{-1}$, since $(-\Delta)^{-1}$ gives a singular matrix.

8.3.2 Number of iterations for the preconditioned system

Table 8.3 shows the number of iterations $\tilde{\mathcal{B}}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ for the *mini*–element without Nitsche method. We used 10^{-8} as tolerance and random values as initial vector. From the table we see that the number of iterations grows as N increases, for $\epsilon > 0$. From the theory we expect that the number of iterations would be bounded. An explanation might be that the bubble function of the *mini*–element is only supposed to occur on the finest grid. The program might use bubbles on the coarse level in the multigrid.

We also find the number of iterations for the lowest order of the Taylor–Hood elements which consist of P_2 elements for the velocity and P_1 elements for the pressure. Table 8.4 shows the result. We see that the number of iterations increases only a bit as N increases for $\epsilon > 0$, this is expected and within reasonable limits. These numbers of iterations

$\epsilon\backslash N$	16	32	64	128	256
1	60	76	120	227	467
2^{-2}	53	65	98	189	384
2^{-4}	48	53	67	108	199
2^{-8}	24	28	39	48	54
0	21	21	22	22	23

Table 8.3: Number of iterations needed to solve the system $\tilde{\mathcal{B}}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ for the *mini*–elements (without Nitsche).

$\epsilon\backslash N$	16	32	64	128	256
1	52	65	82	95	115
2^{-2}	45	55	69	85	101
2^{-4}	37	42	52	63	76
2^{-8}	26	28	31	33	37
0	26	26	26	27	27

Table 8.4: Number of iterations needed to solve the system $\tilde{\mathcal{B}}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ for the *Taylor–Hood* elements (without Nitsche).

are a bit lower than [11] have. The reason might be that we use four smoothing sweeps, which more then normally used.

8.3.3 Condition number with and without Nitsche method

In Chapter 5 and 6 we saw that the Nitsche method seemed to have a stabilizing effect for P_1–P_1 elements. Table 5.1 and 5.2 show good convergence rates for P_1–P_1 elements without any stabilization for the matching case. Similar results the unfitted case can be found in Table 6.7 and 6.8, where we only use ghost–penalties. However, plots showed small oscillations for the pressure.

We explore the stabilizing effect of the Nitsche method further by finding the condition number for the system $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$, both with and without the Nitsche method for Taylor–Hood element, *mini*–element and P_1–P_1 elements without stabilization. Table 8.5 and 8.6 show the condition number when using the Taylor–Hood elements with and without the Nitsche method, respectively. The condition number increases slightly as N increases and is a bit lower when using the Nitsche method. Table 8.5 have similar values as in [42], [38], and [11]. For Table 8.6 we do not have any comparison. Table 8.7 shows the condition number when using the *mini*–elements with the Nitsche method (without Nitsche is displayed i Table 8.1). Here, the condition number is a bit higher when using the Nitsche method. Finally, Table 8.8 shows the condition number for P_1–P_1 element with Nitsche method. When we did not use the Nitsche method, the condition number was between 10^{15} and 10^{30}. Table 8.8 shows that the condition number starts reasonably low and increases somewhere between first and second order as N increases. These results also indicate that the Nitsche method has a stabilizing effect for P_1–P_1 elements, but that the system is not completely stabilized.

$\epsilon \backslash N$	4×4	8×8	16×16
1	13.25	13.44	13.52
2^{-2}	12.18	12.92	13.27
2^{-4}	9.00	10.94	12.24
2^{-8}	5.94	6.13	6.53
0	5.90	6.00	6.05

Table 8.5: Condition number for $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ with Taylor–Hood element.

$\epsilon \backslash N$	4×4	8×8	16×16
1	11.17	11.81	12.17
2^{-2}	8.74	10.18	11.23
2^{-4}	4.56	6.47	8.52
2^{-8}	3.19	3.26	3.32
0	3.19	3.25	3.28

Table 8.6: Condition number for $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ with Taylor–Hood elements with the Nitsche method.

$\epsilon \backslash N$	4×4	8×8	16×16
1	20.05	21.65	22.25
2^{-2}	16.67	19.95	21.50
2^{-4}	7.64	13.26	17.76
2^{-8}	3.33	3.44	3.78
0	3.31	3.37	3.39

Table 8.7: Condition number for $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ with *mini*–elements with the Nitsche method.

$\epsilon \backslash N$	4×4	8×8	16×16
1	16.48	52.60	165.93
2^{-2}	16.40	52.60	165.86
2^{-4}	15.00	51.72	165.16
2^{-8}	12.44	43.81	150.83
0	12.42	43.62	149.65

Table 8.8: Condition number for $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ with P_1–P_1 elements with the Nitsche method.

8.4 Preconditioning the stabilized Brinkman equation

For the stabilized Brinkman equation, the differential operator $\mathcal{A}_\epsilon$ is not the same as (8.4). We write the new differential operator as

$$\mathcal{A}_\epsilon^s = \begin{pmatrix} I - \epsilon^2 \Delta & \text{grad} \\ \text{div} & C \end{pmatrix}, \tag{8.13}$$

where C is the stability term and the superscript s means the stabilized Brinkman equation. In this formulation we only stabilized the pressure. Since $\mathcal{A}_\epsilon$ has changed to $\mathcal{A}_\epsilon^s$ it is reasonable to believe that we have to find a new preconditioner $\mathcal{B}_\epsilon^s$. Table 8.9 shows the condition number for $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}^s$ and Table 8.10 shows the number of iterations for solving $\tilde{\mathcal{B}}_{\epsilon,h}\mathcal{A}_{\epsilon,h}^s$ when using CIP stabilization, where we used the preconditioner in (8.9). We do not consider the Nitsche method in this section, but rather consider the test case (4.47). From Table 8.9 and 8.10 we see that the condition number and number of iterations are large and increasing as N increase for the Stoke case. When ϵ is either close to zero or zero, the values are low and almost not increasing. This indicates that the preconditioner (8.9) works well for (very) low ϵ, but is insufficient for $\epsilon = 1$.

$\epsilon \backslash N$	4×4	8×8	16×16
1	161.2	198.9	236.1
2^{-2}	51.79	113.6	174.7
2^{-4}	8.018	18.41	50.07
2^{-8}	6.652	6.670	6.645
0	6.666	6.720	6.837

Table 8.9: Condition number for $\mathcal{B}_{\epsilon,h}\mathcal{A}^s_{\epsilon,h}$ with CIP stabilization.

$\epsilon \backslash N$	16	32	64	128	256
1	293	347	427	457	493
2^{-2}	256	317	382	419	447
2^{-4}	148	227	289	345	385
2^{-8}	46	49	66	98	166
0	44	46	46	48	50

Table 8.10: Number of iterations needed to solve the system $\tilde{\mathcal{B}}_{\epsilon,h}\mathcal{A}^s_{\epsilon,h}$ with CIP stabilization.

$\epsilon \backslash N$	4×4	8×8	16×16
1	5.05	6.45	7.77
2^{-2}	4.61	5.41	6.48
2^{-4}	5.55	4.93	4.96
2^{-8}	7.73	7.70	7.63
0	7.76	7.80	7.99

Table 8.11: Condition number for $\mathcal{B}^s_{\epsilon,h}\mathcal{A}^s_{\epsilon,h}$ with CIP stabilization.

$\epsilon \backslash N$	16	32	64	128	256
1	47	54	62	67	70
2^{-2}	46	51	57	62	67
2^{-4}	47	50	51	57	64
2^{-8}	37	43	54	67	76
0	39	41	43	48	56

Table 8.12: Number of iterations needed to solve the system $\tilde{\mathcal{B}}^s_{\epsilon,h}\mathcal{A}^s_{\epsilon,h}$ with CIP stabilization.

Our suggestion for a new preconditioner is

$$\mathcal{B}^s_\epsilon = \begin{pmatrix} \left(I-\epsilon^2\Delta\right)^{-1} & 0 \\ 0 & \left(\epsilon^{-2}I - C\right)^{-1} + \left(-\Delta - C\right)^{-1} \end{pmatrix}. \tag{8.14}$$

For the case $\epsilon = 0$ we remove the term $\left(\epsilon^{-2}I - C\right)^{-1}$. We test the preconditioner (8.14) by finding the condition number and number of iterations for both CIP and PSPG stabilization. Table 8.11 and 8.13 show the condition number for CIP and PSPG stabilization, respectively. We see that the numbers are low for both stabilization methods and not growing to much. The number of iterations are shown in Table 8.12 and 8.14. The numbers are low and barely increasing. The results in Table 8.11, 8.12, 8.13 and 8.14 indicate that the preconditioner (8.14) is good.

$\epsilon \backslash N$	4×4	8×8	16×16
1	4.10	5.80	7.08
2^{-2}	4.56	4.95	5.87
2^{-4}	9.13	5.82	4.85
2^{-8}	15.80	16.11	15.71
0	15.87	16.31	16.49

Table 8.13: Condition number for $\mathcal{B}^s_{\epsilon,h}\mathcal{A}^s_{\epsilon,h}$ with PSPG stabilization.

$\epsilon \backslash N$	16	32	64	128	256
1	42	48	56	60	64
2^{-2}	39	42	49	52	57
2^{-4}	32	36	39	44	49
2^{-8}	58	57	48	37	35
0	60	62	64	66	68

Table 8.14: Number of iterations needed to solve the system $\tilde{\mathcal{B}}^s_{\epsilon,h}\mathcal{A}^s_{\epsilon,h}$ with PSPG stabilization.

Chapter 9

Discussion

9.1 The Brinkman equation

The Brinkman equation arises in many different problems. We obtained it from discretizing the Navier–Stokes equations in a certain fashion. Similarly, we can obtain the Brinkman equation by discretizing the time-dependent Stokes equation implicitly in time. The Brinkman equation can be used to model flow through viscous and porous media. The Brinkman equation can also be viewed as a singular perturbation problem when $\epsilon < h$.

One of the challenges is that the solution spaces are different in the Stokes case and the Darcy case. We approach this by letting the solution space depend on ϵ. The weak formulation has two possible solution spaces and two corresponding dual spaces.

$$\begin{aligned} X_\epsilon &= \left(L^2 \cap \epsilon H^1\right) \times \left(\left(H^1 \cap L_0^2\right) + \epsilon^{-1} L_0^2\right), \\ X_\epsilon^* &= \left(L^2 + \epsilon^{-1} H^{-1}\right) \times \left(H_0^{-1} \cap \epsilon L_0^2\right) \end{aligned} \tag{9.1}$$

and

$$\begin{aligned} X_\epsilon &= \left(H\,(\mathrm{div}) \cap \epsilon H^1\right) \times L_0^2, \\ X_\epsilon^* &= \left(H^{-1}\,(\mathrm{rot}) + \epsilon^{-1} H^{-1}\right) \times L_0^2. \end{aligned} \tag{9.2}$$

For our analysis in this thesis, we looked at the (9.1) spaces. Our programs follows the (9.2) spaces. To change these to the (9.1) spaces we add an *if-test* using the formulation in Problem 6.1 for $\epsilon = 0$.

Finding elements which are uniformly stable; that is, stable for all $\epsilon \in [0, 1]$, is challenging. Mardal et al. [2] points out that many of the elements which are stable for the Stokes case are not stable for the Darcy case, and vice versa. Examples of such elements are the Crouzeix–Raviart element, the P_2–P_0 element and the Raviart-–Thomas element. The article also points out that the *mini*–element is uniformly stable for the (9.1) spaces and shows that the Mardal–Tai–Winther element is uniformly stable for the (9.2) spaces. We refer to Chapter 3 in [20] for the definitions of the elements mentioned in this section. Unfortunately the software we used, does not have the Mardal–Tai–Winther element implemented.

9.1.1 Stabilization methods

A different approach than using uniformly stable elements is to use stabilization methods. The software used only supported P_1 and P_0 cut elements, and since implementation of higher order elements was deemed out of scope, we chose to focus on stabilization methods. There are many good reasons to use stabilization methods over uniformly stable elements, one being that uniformly stable elements often have more degrees of freedom which requires more memory and floating point operations to solve.

We have considered the PSPG method and the CIP method. Juntunen and Stenberg [3] show stability and both a priori and a posteriori error estimates for the *mini*–element and PSPG method for the (9.1) spaces. The CIP method for the Brinkman problem were presented and analyzed by Burman and Hansbo [4]. We mainly considered the CIP method and used the PSPG method as comparison.

The PSPG method is a well known method which has been widely used in practice with good results. The method does however have some disadvantages [33], one being that mass lumping is impossible. The CIP method overcomes these disadvantages with no additional unknowns added. A practical advantage in our case is that we use the same similar terms for the CIP stabilization as for the ghost penalties. The results in Tables 4.3, 4.4, 4.5 and 4.6 do not show any noticeable differences between the two methods. Other stabilization methods and their analysis can be found in [43].

9.1.2 Nitsche method

The need to weakly impose boundary conditions has made the Nitsche method popular in recent years. The method was mentioned and discussed by Nitsche [9] in 1971. Juntunen and Stenberg [26] consider the Nitsche method for general boundary conditions for the Poisson equation. The Nitsche method was used in fictitious domain methods with cut elements in [5] for the Poisson equation and extended to the Stokes equation in [6]. The Nitsche method can also be used for domain decomposition problems where there is a jump on the boundary interacting the domains, and for overlapping meshes methods [44, 45]. An alternative to the Nitsche method is to use Lagrange multiplier defined as element-wise constant on the cut elements [46, 47]. The Nitsche method is however preferred since it does not introduce multipliers.

Finding a lower bound for the penalty parameter γ is of practical interest and we discuss it for the fitted case. In the proof of the coercivity Theorem 7.5 we find that γ must be larger than C_I, where C_I is the constant in the inverse estimate (7.1). C_I can be found by finding the largest eigenvalue, λ_{max}, in the eigenproblem [37]:

Find $\mathbf{u}_h \in V_h$ and $\lambda \in \mathbb{R}$ such that

$$\left(h_T^{1/2}\partial_{\mathbf{n}}\mathbf{u}_h, \partial_{\mathbf{n}}\mathbf{v}_h\right)_{\partial T\cap\Gamma} = \lambda\left(\nabla\mathbf{u}_h, \nabla\mathbf{v}_h\right)_T, \quad \mathbf{v}_h \in V_h. \tag{9.3}$$

For P_1 we can show

$$C_I = \frac{h_T \operatorname{meas}(\partial T\cap\Gamma)}{\operatorname{meas}(T)}. \tag{9.4}$$

For the unit square test case in Section 4.5, $C_I = 2\sqrt{2}$ ($C_I = 4\sqrt{2}$ for two of the corner elements).

Our numerical results indicate that the Nitsche method has a stabilizing effect. Table 5.1 and 5.2 show errors and convergence rates for P_1–P_1 elements without stabilization for matching meshes and Table 6.7 and 6.8 show for the unfitted mesh. These tables show low errors and good convergence rates. Figure 5.2 reveals small oscillations for the pressure. In Chapter 8 we explored the Nitsche methods further by finding the condition number of $\mathcal{B}_{\epsilon,h}\mathcal{A}_{\epsilon,h}$ (Table 8.8). These condition numbers are small compared to when we don't use the Nitsche method, where the condition number was between $1.0e+15$ and $1.0e+30$. The condition numbers in Table 8.8 are increasing with an order between 1.5 and 2 as N increases. The results in the tables and plots indicate that the Nitsche method has a stabilizing effect, but that it is not stable in term of the inf-sup condition.

9.1.3 Fictitious domain finite element methods

Classical fictitious domain methods like [48], extend the equation to the fictitious domain Ω^* and set the source function to zero in $\Omega^*\backslash\Omega$. The normal derivative across the boundary is not well approximated and there is a loss of accuracy. To address this we use fictitious domain with cut elements. The problem is that there is no lower bound on the size of $T\cap\Omega$ and no upper bound on the condition number of the system matrix [5]. One consequence is that the denominator in Equation (9.4) is not bounded from below and stability fails on the boundary. We use ghost-penalties to stabilize in the interface zone which extends the coercivity to the fictitious domain [7].

One challenge is to find scaling functions $s_{\mathbf{u}}$ and s_p for the ghost-elements for the Brinkman equation. Proposition 5.1 was used to suggest scaling functions for $\epsilon = 1$ and $\epsilon = 0$. However, for $\epsilon = 0$ we suggested $s_{\mathbf{u}} = h_T^3$, but numerical results from Table 6.1 shows that this did not achieve the expected convergence rates. Instead we use $s_{\mathbf{u}} = h_T^{-1}$ for $\epsilon = 0$, which gave good convergence rates. The scaling functions are

$$s_{\mathbf{u}}(h_T) = \frac{h_T}{\epsilon^2 + h_T^2} \quad \text{and} \quad s_p(h_T) = \frac{h_T^3}{\epsilon^2 + h_T^2}. \tag{9.5}$$

Tables 6.3 to 6.10 show results using the scaling functions above. Common for the tables is that the convergence rates are good for $\epsilon = 1$ and $\epsilon = 0$, but the convergence rates for

ϵ in between, are not as good. This might indicate that the scaling functions (9.5) are insufficient or that the scaling functions can be improved.

Table 6.9 and 6.10 show results when using "bad" cut elements. The convergence rates are affected. However, this is a unlikely test case where every cut element are "bad". This test case shows, to a certain extent, the robustness of the method.

9.1.4 A priori error estimate

In Chapter 7 we proved an a priori error estimate for the Brinkman equation with the Nitsche method and CIP stabilization on a matching mesh. We considered the spaces in (9.1). Our spaces and norms are very similar to those employed by Juntunen and Stenberg [3] use. Juntunen and Stenberg [3] prove both an a priori and an a posteriori error estimate for the *mini*–element and P_1–P_1 with PSPG stabilization, and at the end they discuss the Nitsche method. Burman and Hansbo [4] prove an a priori error estimate for the CIP stabilized Brinkman equation (without Nitsche), however in the other spaces (9.2). Techniques from both [4] and [3] were used to prove Theorem 7.7. A natural extension of our error estimate would be to prove it for the fictitious domain method presented in Chapter 5. Massing et al. [8] prove an a priori error estimate for the PSPG stabilized Stokes problem for the same fictitious domain method we considered.

9.1.5 Preconditioner for the fictitious domain method

In Chapter 8 we discussed preconditioners for the unstabilized Brinkman equation. We used the theory from [10] and [11]. The preconditioner was tested by counting the number of iterations and finding the exact condition number (to machine precision). We also suggested a preconditioner (Equation (8.14)) for the stabilized Brinkman equation and performed the same test with good results.

In the same chapter we only considered matching meshes and not the fictitious domain method from Chapter 5. To extend to the fictitious domain method we could use the same preconditioner (8.14) and include the ghost penalties for the pressure in C and the ghost penalties for the velocity in $\left(I - \epsilon^2\Delta\right)^{-1}$. However, this have not been tested.

9.1.6 Limitations of the numerical results

Most of our numerical results are produced from the manufactured solution Equation (4.47) and (6.1). This is one of infinitely many possible solutions. Our solution might not contain the whole spectrum of eigenfunctions; that is, it might be that our solution only test low frequent solutions and not high frequent solutions. To thoroughly test the problem, manufactured solutions with different properties should be used. An example of another manufactured solution we could use, is $\mathbf{u} = \epsilon\nabla \times e^{-xy/\epsilon}$ and $p = \epsilon\, e^{x/\epsilon}$. This solution is ϵ-dependent and its behavior is typical for solutions of singular perturbation problems [2]. Also this solution has inhomogeneous boundary conditions, in contrast to the solution in Equation (4.47) and (6.1) where $\mathbf{g} = 0$ on Γ. The numerical results should be produced for higher mesh resolutions, especially for Table 6.10. If we study the table closely we can see that the convergence rate between $N = 64$ and $N = 128$, are low. We would like to know if it is low for $N = 128$ and $N = 256$ as well.

The condition number tests in Chapter 8, done for $\mathcal{A}_{\epsilon,h}$ and $\mathcal{A}^{s}_{\epsilon,h}$, only depend on $V_h(\mathcal{T}) \times Q_h(\mathcal{T})$ and not the source term $\mathbf{f}(\mathbf{x})$. These tests do not have the same problem as the manufactured solutions might have and can be a good supplement to manufactured solutions. The condition numbers are exact to machine precision and we are only able to find them using a coarse grids (low N).

9.2 Conclusions and further work

We have studied a fictitious domain method with cut elements, which allows the domain to be independent of the mesh. This is a desired property since meshing and mesh analysis can be difficult and time consuming for complex domains. The Nitsche method is useful to weakly impose Dirichlet boundary conditions and we also observed some unexpected stabilizing effects.

The parameter dependent Brinkman equation covers a family of problems, ranging from the Stokes flow to the Darcy flow. Two stabilization techniques were considered; the more classical PSPG method and the CIP method. Both methods have similar numerical results, however the CIP method is preferred because of its practicality (with respect to ghost penalties) and because it allows mass lumping [33]. An a priori error estimate for the CIP stabilized Brinkman equation with the Nitsche method on matching meshes was proved.

Suggestions for scaling functions (9.5) for the ghost penalties were given. They showed good results for $\epsilon = 0$ and $\epsilon = 1$, but could be improved for intermediate ϵ values.

Preconditioners have been studied for the fitted case and suggestions on how to extend to the unfitted case are given. This thesis lays some of the foundation for application on large problems. In particular blood flow trough aneurysms with a stent inserted, since the Navier–Stokes equations can be discretized to the Brinkman equation and a level set function for the stent is provided (see B.2).

9.2.1 Further work

The field of non-matching mesh methods is large and expanding. Further research and development of the methodology is needed, as well as finding suitable preconditioners.

In relation to this thesis we mention only a couple of things that is subject for further work:

- Explore the stabilization effect of the Nitsche method.
- Improve the scaling functions $s_{\mathbf{u}}$ and s_p.
- Extend the a priori error estimate to fictitious domain formulations.
- Explore the preconditioner for fictitious domain formulations.
- Further software development; extend to higher order elements.
- Simulation on an aneurysm with a stent inserted.

Appendix A

Mathematical definitions and formulas

Definition A.1. *L^2 inner product, $(\cdot,\cdot)_\Omega$, over the domain, $\Omega \subset \mathbb{R}^d$*
Let A and B be two $n \times m$ matrices. There inner product is,

$$(A, B)_\Omega = \int\limits_\Omega A : B \, d\mathbf{x} = \int\limits_\Omega \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} B_{ij} \, d\mathbf{x}. \tag{A.1}$$

Assuming that the entries $A_{ij}B_{ij}$ are integrable.

Definition A.2. *Weak derivative*
Suppose $u, v \in L^1_{loc}(\Omega)$, and that α is a multi-index. We say that v is the α^{th}-weak partial derivative of u, written

$$D^\alpha u = v,$$

provided

$$\int\limits_\Omega u \, D^\alpha \phi \, d\mathbf{x} = (-1)^{|\alpha|} \int\limits_\Omega v \, \phi \, d\mathbf{x} \quad \forall \, \phi \in C_0^\infty (\Omega) \, . \tag{A.2}$$

Definition A.3. *The Sobolev space*

$$H^k(\Omega)$$

consists of all locally summable functions $u : \Omega \to \mathbb{R}$ such that for each multi-index α with $|\alpha| \leq k$, $D^\alpha u$ exists in the weak sense and belongs to $L^2(\Omega)$.

Definition A.4. *The $H^k(\Omega)$ norm*
If $u \in H^k(\Omega)$, we define the norm to be

$$\|u\|_{H^k(\Omega)} = \|u\|_{k,\Omega} = \left(\sum_{|\alpha| \leq k} \int\limits_\Omega |D^\alpha u|^2 d\mathbf{x} \right)^{\frac{1}{2}} . \tag{A.3}$$

Similarly, we define a semi to be

$$|u|_{H^k(\Omega)} = |u|_{k,\Omega} = \left(\sum_{|\alpha|=k} \int\limits_{\Omega} |D^\alpha u|^2 d\mathbf{x} \right)^{\frac{1}{2}}. \tag{A.4}$$

Definition A.5. *The Sobolev space*

$$H\,(div;\,\Omega)$$

consists of all locally summable functions $\mathbf{u} : \Omega \to \mathbb{R}^d$ *in* $\left[L^2\,(\Omega)\right]^d$ *such that the divergence exists in the weak sense and belongs to* $L^2\,(\Omega)$. *The* $H\,(div;\,\Omega)$ *norm is defined as*

$$\|\mathbf{u}\|_{H(div;\,\Omega)} = \left(\int\limits_{\Omega} |\mathbf{u}|^2 d\mathbf{x} + \int\limits_{\Omega} |\nabla \cdot \mathbf{u}|^2 d\mathbf{x} \right)^{\frac{1}{2}}. \tag{A.5}$$

Definition A.6. *Equivalent operators*
Let $\mathcal{B}$ *and* $\tilde{\mathcal{B}}$ *be two operators or matrices for* $v \in V$. *They are said to be equivalent if there exits constants* c_0, $c_1 > 0$ *such that*

$$c_0 \left(\tilde{\mathcal{B}} v, v \right) \leq (\mathcal{B} v, v) \leq c_1 \left(\tilde{\mathcal{B}} v, v \right) \quad \forall\, v \in V. \tag{A.6}$$

There is a similar definition for norms.

Theorem A.1. *Cauchy's inequality*

$$2ab \leq a^2 + b^2 \tag{A.7}$$

Proof. $0 \leq (a-b)^2 = a^2 - 2ab + b^2$. □

Theorem A.2. *Cauchy's inequality with* δ
Let $\delta > 0$, *then*

$$2ab \leq \frac{1}{\delta} a^2 + \delta b^2 \tag{A.8}$$

Proof. Write

$$2ab = \frac{a^2}{\delta^{1/2}}\, \delta^{1/2} b^2$$

and apply the Cauchy's inequality. □

Theorem A.3. *Cauchy–Schwarz's inequality*
If $u, v \in L^2\,(\Omega)$, *we have*

$$\int\limits_{\Omega} |uv|\, dx \leq \|u\|_{L^2(\Omega)} \|v\|_{L^2(\Omega)}. \tag{A.9}$$

A discrete version: If $a = (a_1, \dots a_n,)$, $b = (b_1, \dots b_n) \in \mathbb{R}^n$, *we have*

$$|\sum_{k=1}^{n} a_k b_k| \leq \left(\sum_{k=1}^{n} |a_k|^2\right)^{\frac{1}{2}} \left(\sum_{k=1}^{n} |b_k|^2\right)^{\frac{1}{2}}. \tag{A.10}$$

Proof. Proof can be found in [49]. □

Theorem A.4. *If* X *and* Y *are Hilbert spaces, both continuously contained in some larger Hilbert spaces, then the intersection* $X \cap Y$ *and the sum* $X + Y$ *are themselves Hilbert spaces with the norms*

$$\|q\|_{X\cap Y} = \left(\|q\|_X^2 + \|q\|_Y^2\right)^{1/2} \tag{A.11}$$

and

$$\|q\|_{X+Y} = \inf_{q=x+y,\, x\in X,\, y\in Y} \left(\|x\|_X^2 + \|q\|_y^2\right)^{1/2}. \tag{A.12}$$

Furthermore if $X \cap Y$ *is dense in both* X *and* Y *then*

$$(X \cap Y)^* = X^* + Y^* \tag{A.13}$$

and

$$(X + Y)^* = X^* \cap Y^* \tag{A.14}$$

Proof. Proof can be found in Chapter 2 in [50]. □

Appendix B

Source code

B.1 Brinkman problem without Nitsche

Python code

```python
#-*- coding:utf-8 -*-
from dolfin import *
import numpy, sys

#Default parameters
use_mini = False; use_pspg = False;
use_cip  = False; epsilon = 1.0;
beta = 0.1; N = 6;

#Read in command line arguments
for arg in sys.argv[1:]:
  exec(arg)
  print arg,

mesh = UnitSquareMesh(N,N)
h_T = CellSize(mesh); h = avg(h_T)
eps2 = Constant(epsilon**2); n = FacetNormal(mesh)

#Define the function spaces
V = VectorFunctionSpace(mesh, ’CG’, 1)
Q = FunctionSpace(mesh, ’CG’, 1)
R = FunctionSpace(mesh,"R", 0)
if use_mini:
  print "using MINI element" #RM
  B =VectorFunctionSpace(mesh, "Bubble", 3)
  V = V + B
W = MixedFunctionSpace([V,Q,R]);

u, p, r = TrialFunctions(W);
v, q, s = TestFunctions(W);
```

```
#Source function
f = Expression((" pi*sin(2*pi*x[1])*pow(sin(pi*x[0]),2)
    +ep*ep*2*pi*pi*pi*sin(2*pi*x[1])*(2*pow(sin(pi*x[0]),2)-
    cos(2*pi*x[0])) - 2*pi*cos(2*pi*x[0])",
    "-pi*sin(2*pi*x[0])*pow(sin(pi*x[1]),2)
    -ep*ep*2*pi*pi*pi*sin(2*pi*x[0])*(2*pow(sin(pi*x[1]),2)-
    cos(2*pi*x[1]))"),ep=epsilon)
#Analytical solution
u_exact = Expression(("pi*sin(2*pi*x[1])*pow(sin(pi*x[0]),2)",
                      "-pi*sin(2*pi*x[0])*pow(sin(pi*x[1]),2)"))
p0 = Expression("-sin(2*pi*x[0])")

#Standard form of a and L
a = inner(u,v)*dx + eps2*inner(grad(u),grad(v))*dx;
b = -div(v)*p*dx + q*r*dx;
b+= -div(u)*q*dx + p*s*dx;

L = dot(f,v)*dx + p0*s*dx

if use_cip:
  s_stab = -avg(beta*h_T**3/(epsilon**2 + h_T**2))
  c = s_stab*dot(jump(grad(p),n),jump(grad(q),n))*dS;
elif use_pspg:
  s_stab = -beta*(h_T**2/(epsilon**2 + h_T**2))
  a += s_stab*dot(u,v)*dx;
  b += s_stab*dot(grad(p),v)*dx + s_stab*dot(u,grad(q))*dx;
  c  = s_stab*dot(grad(p),grad(q))*dx ;
  L += s_stab*dot(f,v)*dx + s_stab*dot(f,grad(q))*dx;
else:
  c = 0

#Define boundary and boundary condition
def boundary(x):
    return x[0] < DOLFIN_EPS or x[0] > 1.0 - DOLFIN_EPS or \
        x[1] < DOLFIN_EPS or x[1] > 1.0 - DOLFIN_EPS
bcs = DirichletBC(W.sub(0), u_exact, boundary)

A, B =  assemble_system(a+b+c, L, bcs)
w = Function(W) #Solution functions
solve(A, w.vector(), B)

U, Pr, Rr = w.split()
U = Function(V,U); Pr= Function(Q,Pr);

#Save solution for post processing
mesh_to_file = File("results/boxmesh_N"+str(N)+".xml.gz")
u_to_file = File("results/velocity_N"+str(N)+".xml")
p_to_file = File("results/pressure_N"+str(N)+".xml")
mesh_to_file << mesh
u_to_file << U
p_to_file << Pr
```

B.2 Level set function for a stent

C++ code

```
//Define Constants
double r_s = 2.5; //blood vessel radius, mm
double   H = 8.0;//blood vessel height, mm
double h_t = 0.050;//stent tread thickness mm
int     w  = 1;//number of twists
int    tr  = 6;//number of treads in each directions

double level_set(double x1, double y1, double z1)
{ //Change to cylinder coordinates r, theta, z
  double  r1 = sqrt(x1*x1 +y1*y1); //Radius
  double   b = h_t;                  //semi-minor axis
  double   a = h_t*sqrt(pow(pi*2*r_s,2) +
      ((H*H)/(w*w)))/(2*pi*r_s); //semi-major axis
  double level_set_value, theta, theta2, z1_t;
  //Translation to the center of the stent line
  double r1_t= r1-r_s ;
  double delta_angle = 2*pi/tr; //angle between two stent lines

  for ( int k = 0; k < tr; k++) //check each tread
    { //Angle
      theta  = atan2(y1,x1) + pi - delta_angle*k;
      theta2 = -theta;//Angle opposite direction

      //Check level set value in first direction
      if (theta < 0){theta = theta + 2*pi;}
      z1_t= z1-theta*H/(w*2*pi);
      level_set_value = pow(r1_t/b,2) + pow(z1_t/a,2);
      if (level_set_value < 1){return level_set_value;}

      //Check level set value in  other direction
      if (theta2 < 0){theta2 = theta2 + 2*pi;}
      z1_t= z1-theta2*H/(w*2*pi);
      level_set_value = pow(r1_t/b,2) + pow(z1_t/a,2);
      if (level_set_value < 1){return level_set_value;}
    }
  // level_set_value < 1: The point is in the stent line
  // level_set_value = 1: The point is on the stent line surface
  // level_set_value > 1: The point is outside.
  return level_set_value;}
```

B.3 UFL code for the fictitious domain simulation

Python code

```python
from ufl import *
from ffc import *

cell = tetrahedron

#Function space
V = VectorElement("CG", cell, 1)
Q = FiniteElement("CG", cell, 1)
R = FiniteElement("R", cell, 0)
W = MixedElement([V,Q,R])

# Assume h_T is cell diameter = 2 circumradius
h_T = 2*cell.circumradius; h_F = avg(h_T)
n = cell.n; I = Identity(3)
# Normal vector on Surface
n_I = Coefficient(VectorElement("DG", cell, 0))

# Source terms
F = VectorElement("CG", cell, 2);
P = FiniteElement("CG", cell, 2);
f = Coefficient(F); g = Coefficient(F)
p_exact = Coefficient(P)

#Kinematic viscosity (epsilon*epsilon)
eps = Constant(cell)

# Ghost penalty parameters and Nitsche penalty parameter
beta_u = Constant(cell); beta_p = Constant(cell)
beta_CIP = Constant(cell); gamma = Constant(cell)

# Define test and trial functions
(u, p, r) = TrialFunctions(W)
(v, q, s) = TestFunctions(W)

# Standard form used for standard and cut elements
a_h = inner(u,v)*dx + eps*inner(grad(u), grad(v))*dx;
# The x-y plane, quasi 2D
a_h_surf_fitted = inner(dot(v, n), p)*ds(10) + inner(dot(u, n),
    q)*ds(10);

# Nitsche boundary terms for the velocity
a_h_surface = -eps*inner(grad(u)*n_I, v)*ds - \
    eps*inner(grad(v)*n_I, u)*ds + eps*gamma*inv(h_T)*dot(u, v)*ds

# dS(0): CIP inf-sup stabilization
# dS(1): Ghost-penalty dS(1)
a_h_ghost= avg(beta_u*h_T/(eps + h_T*h_T)) * \
    dot(jump(grad(u), n), jump(grad(v), n))*dS(1)
b_h_stab  = -avg(beta_CIP*h_T**3/(eps + h_T**2)) * \
    dot(jump(grad(p),n),jump(grad(q),n))*dS(0)
b_h_stab += -avg(beta_p*h_T**3/(eps + h_T**2)) * \
    dot(jump(grad(p),n),jump(grad(q),n))*dS(1)

# Standard form used for standard and cut elements
b_h = - div(v)*p*dx - div(u)*q*dx
# Nitsche boundary terms related to the pressure
b_h_surface = inner(dot(v, n_I), p)*ds + inner(dot(u, n_I), q)*ds

# Construct overall forms
A_h = a_h + b_h
A_h+= p*s*dx + q*r*dx
J_h = a_h_ghost + b_h_stab
A_h_surf = a_h_surface + b_h_surface

# Right-hand side
L_h = inner(f,v)*dx
L_h+= p_exact*s*dx
L_h_surf = -inner(eps*grad(v)*n_I,g)*ds  + inner(q*I*n_I, g)*ds
L_h_surf+= eps*gamma*inv(h_T)*dot(g, v)*ds

#------------------------------------------------------------
# Compile forms
#------------------------------------------------------------
params = default_parameters()
params["format"] = "dolfin"
params["optimize"] = True
params["split"]=True
object_names = {id(beta_u): "beta_u", id(beta_p): "beta_p",
                id(beta_CIP): "beta_CIP", id(eps): "eps",
                id(p_exact): "p_exact",  id(gamma): "gamma",
                id(n_I): "n_I", id(f): "f", id(g): "g"}

# Compile form for non.cut elements including ghost-penalty
compile_form([A_h+J_h+a_h_surf_fitted, L_h], parameters=params,
             prefix="FidoBrinkmanP1P1", object_names=object_names)

compile_element([V], prefix="V1", parameters=params)
compile_element([Q], prefix="Q1", parameters=params)

# Compile form for cut elements by changing "representation"
params["representation"] = "physical"
params["optimize"] = False
compile_form([A_h, L_h], prefix="FidoBrinkmanP1P1Cut",
             parameters=params, object_names=object_names)

# Compile form for integration on surface ("physical")
compile_form([A_h_surf,L_h_surf], prefix="FidoBrinkmanP1P1Surf",
             parameters=params, object_names=object_names)
```

Bibliography

[1] D Braess. *Finite elements: Theory, fast solvers, and applications in solid mechanics.* Cambridge University Press, 2007.

[2] K-A Mardal, X-C Tai, and R Winther. A Robust Finite Element Method for Darcy–Stokes Flow. *SIAM Journal on Numerical Analysis*, 40(5):1605–1631, 2002.

[3] M Juntunen and R Stenberg. Analysis of finite element methods for the Brinkman problem. *Calcolo*, 47(3):129–147, 2010.

[4] E Burman and P Hansbo. Edge stabilization for the generalized Stokes problem: a continuous interior penalty method. *Computer methods in applied mechanics and engineering*, 195(19):2393–2410, 2006.

[5] E Burman and P Hansbo. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Applied Numerical Mathematics*, 62(4): 328–341, 2012.

[6] E Burman and P Hansbo. Fictitious domain methods using cut elements: III. A stabilized nitsche method for Stokes' problem. Technical report, School of Engineering, Jönköping University, 2011.

[7] E Burman. Ghost penalty. *Comptes Rendus Mathematique*, 348(21):1217–1220, 2010.

[8] A Massing, M G Larson, A Logg, and M E Rognes. A stabilized Nitsche fictitious domain method for the Stokes problem. *arXiv preprint arXiv:1206.1933*, 2012.

[9] J Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer, 1971.

[10] K-A Mardal and R Winther. Preconditioning discretizations of systems of partial differential equations. *Numerical Linear Algebra with Applications*, 18(1):1–40, 2011.

[11] K-A Mardal and R Winther. Construction of Preconditioners By Mapping Proper-

ties for Systems of Partial Differential Equations. *Efficient Preconditioned Solution Methods for Elliptic Partial Differential Equations*, pages 66–86, 2011.

[12] J R Cebral and M Raschi. Suggested Connections Between Risk Factors of Intracranial Aneurysms: A Review. *Annals of biomedical engineering*, pages 1–18, 2012.

[13] J Isaksen, A Egge, K Waterloo, B Romner, and T Ingebrigtsen. Risk factors for aneurysmal subarachnoid haemorrhage: the Tromsø study. *Journal of Neurology, Neurosurgery & Psychiatry*, 73(2):185–187, 2002.

[14] N K de Rooij, F H H Linn, J A van der Plas, A Algra, and G J E Rinkel. Incidence of subarachnoid haemorrhage: a systematic review with emphasis on region, age, gender and time trends. *Journal of Neurology, Neurosurgery & Psychiatry*, 78(12): 1365–1372, 2007.

[15] J G Isaksen. *Simulation of hemodynamics and biomechanics in intracranial aneurysms and the Circle of Willis.* PhD thesis, University of Tromsø, 2012.

[16] J R Cebral, M A Castro, J E Burgess, R S Pergolizzi, M J Sheridan, and C M Putman. Characterization of cerebral aneurysms for assessing risk of rupture by using patient-specific computational hemodynamics models. *American Journal of Neuroradiology*, 26(10):2550–2559, 2005.

[17] J R Cebral, F Mut, M Raschi, E Scrivano, R Ceratto, P Lylyk, and C M Putman. Aneurysm rupture following treatment with flow-diverting stents: computational hemodynamics analysis of treatment. *American journal of neuroradiology*, 32(1): 27–33, 2011.

[18] P K Kundu, I M Cohen, and D R Dowling. *Fluid Mechanics.* Academic Press, 5. edition, 2012.

[19] H P Langtangen, K-A Mardal, and R Winther. Numerical Methods for Incompressible Viscous Flow. *Advances in Water Resources*, 25(8):1125–1146, 2002.

[20] A Logg, K-A Mardal, and G Wells. *Automated Solution of Differential Equations by the Finite Element Method: The fenics book*, volume 84. Springer, 2012.

[21] S C Brenner and L R Scott. *The Mathematical Theory of Finite Element Methods*, volume 15. Springer, 3. edition, 2008.

[22] F Brezzi and M Fortin. *Mixed and hybrid finite element methods.* Springer-Verlag New York, Inc., 1991.

[23] D N Arnold, F Brezzi, and M Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344, 1984.

[24] T J R Hughes, L P Franca, and M Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A

stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59 (1):85–99, 1986.

[25] FEniCS Project. `http://www.fenicsproject.org/`. URL `http://www.fenicsproject.org/`.

[26] M Juntunen and R Stenberg. Nitsche's method for general boundary conditions. *Mathematics of computation*, 78(267):1353–1374, 2009.

[27] A Massing, M G L, and A Logg. Efficient Implementation of Finite Element Methods on Nonmatching and Overlapping Meshes in Three Dimensions. *SIAM Journal on Scientific Computing*, 35(1):C23–C47, 2013.

[28] N Sukumar, N Moës, B Moran, and T Belytschko. Extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering*, 48(11):1549–1570, 2000.

[29] B Mirtich. Fast and accurate computation of polyhedral mass properties. *Journal of graphics tools*, 1(2):31–50, 1996.

[30] M S Alnæs. UFL: a finite element form language. In *Automated Solution of Differential Equations by the Finite Element Method*, pages 303–338. Springer, 2012.

[31] A Hansbo and P Hansbo. An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer methods in applied mechanics and engineering*, 191(47):5537–5552, 2002.

[32] E Burman. A unified analysis for conforming and nonconforming stabilized finite element methods using interior penalty. *SIAM journal on numerical analysis*, 43 (5):2012–2033, 2005.

[33] E Burman, M A Fernández, and P Hansbo. Continuous interior penalty finite element method for Oseen's equations. *SIAM journal on numerical analysis*, 44(3): 1248–1274, 2006.

[34] P Clément. Approximation by finite element functions using local regularization. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 9(R2):77–84, 1975.

[35] L R Scott and S Zhang. Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation*, 54(190):483–493, 1990.

[36] E Burman and P Zunino. Numerical approximation of large contrast problems with the unfitted Nitsche method. In *Frontiers in Numerical Analysis-Durham 2010*, pages 227–282. Springer, 2012.

[37] P Hansbo. Nitsche's method for interface problems in computational mechanics. *GAMM-Mitteilungen*, 28(2):183–206, 2005.

[38] K-A Mardal, J Schöberl, and R Winther. A uniform inf–sup condition with applications to preconditioning. *arXiv preprint arXiv:1201.1513*, 2012.

[39] A Logg and K-A Mardal. Lectures on the finite element method. Lecture notes in INF5650 "Numerical methods for partial differential equations 2", 2013.

[40] cbc.block. `https://bitbucket.org/fenics-apps/cbc.block`. URL `https://bitbucket.org/fenics-apps/cbc.block`.

[41] Octave. `http://www.gnu.org/software/octave/`. URL `http://www.gnu.org/software/octave/`.

[42] K-A Mardal and R Winther. Uniform preconditioners for the time dependent Stokes problem. *Numerische Mathematik*, 98(2):305–327, 2004.

[43] E Burman and M A Fernández. Galerkin finite element methods with symmetric pressure stabilization for the transient Stokes equations: stability and convergence analysis. *SIAM Journal on Numerical Analysis*, 47(1):409–439, 2008.

[44] A Massing, M G L, A Logg, and M E Rognes. A stabilized Nitsche overlapping mesh method for the Stokes problem. *Numerische Mathematik*, pages 1–29, 2014. ISSN 0029-599X. doi: 10.1007/s00211-013-0603-z. URL `http://dx.doi.org/10.1007/s00211-013-0603-z`.

[45] E Burman and M A Fernández. An unfitted Nitsche method for incompressible fluid-structure interaction using overlapping meshes. Rapport de recherche RR-8424, INRIA, December 2013. URL `http://hal.inria.fr/hal-00918272`.

[46] E Burman and P Hansbo. Fictitious domain finite element methods using cut elements: I. A stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 199(41):2680–2686, 2010.

[47] I Babuška. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.

[48] V Girault and R Glowinski. Error analysis of a fictitious domain method applied to a Dirichlet problem. *Japan Journal of Industrial and Applied Mathematics*, 12 (3):487–514, 1995.

[49] L C Evans. *Partial Differential Equations*, volume 19. American Mathematical Society, 2010.

[50] J B-J Lofstrom and J Bergh. Interpolation spaces. *SpringereVerlag, Newe*, 1976.