Divergence Conforming Discontinuous Galerkin FEM for the Stokes Problem

Written by
Tobias Rohner

Supervised by
Prof. Dr. Ralf Hiptmair
Seminar for Applied Mathematics
ETH Zürich

September 1, 2019

Abstract

This thesis discusses the Interior Penalty Discontinuous Galerkin discretization for the incompressible Stokes equations on general divergence conforming finite element spaces and provides a C++ implementation of the zeroth order case. The rate of convergence is determined empirically by means of numerical experiments. We also demonstrate the limitations of said low order implementation.

Contents

1   Introduction

In this thesis we derive the Interior Penalty Discontinuous Galerkin (IPDG) discretization for the Stokes problem and provide a zeroth order implementation based on the LehrFEM++ framework developed at the Seminar for Applied Mathematics at ETH Zürich [10]. The code of the implementation can be found at `https://gitlab.ethz.ch/torohner/ipdg-fem-stokes`.

The thesis is split into four main parts. The first part in section 2 discusses the mathematical derivation of the method. We first provide a description of the Stokes Problem itself in subsection 2.1. This is followed by a description of the function space used and a thorough derivation of the IPDG formulation in the subsequent subsections.

The second part, section 3, deals with the discretization of the problem formulation by discretizing the function space with the Raviart-Thomas finite element space RT_p. Because of the discontinuous nature of the functions, problems arise when attempting to construct a system matrix suitable for local assembly. We discuss the solution for this local assembly problem for each term in the variational formulation. The last subsection deals with the way boundary conditions are imposed differentiating between those having no fluid flow and those that do have fluid flow over the domain boundary.

The implementation of the discretized problem for a zeroth order finite element space is then discussed in the third part found in section 4. We derive the specific structure of the element matrices and vectors and the algorithm to find the offset function for imposing boundary conditions. This is followed by a discussion of the implementation challenges arising when using higher order finite element spaces and solutions thereof are briefly examined.

The fourth part found in section 5 contains convergence studies in the form of analytical results and multiple numerical experiments. Each one explores a possible pitfall originating from the specific choices made during the implementation step. Some experiments demonstrating the limitations of the given zeroth order implementation can be found in subsection 5.2 and subsection 5.6. The lid driven cavity experiment in subsection 5.3 demonstrates the solver for boundary conditions with no fluid flow over the domain boundaries while the experiments conducted in subsection 5.4 and subsection 5.5 make use of the solver involving flows over the domain boundary in the form of fluid flowing through various pipes.

2 Mathematical Derivation

2.1 The Stokes Equations

Fluid flow in d-dimensions is generally described by the Navier-Stokes equations [7, Chapter 4] in terms of the flow velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^d$, the density $\rho \in \mathbb{R}$, the dynamic viscosity $\mu \in \mathbb{R}$ and the volumetric forces $\boldsymbol{f} : \Omega \to \mathbb{R}^d$:

$$\rho \frac{\mathrm{D}\boldsymbol{u}}{\mathrm{D}t} = -\nabla p + \mu \Delta \boldsymbol{u} + \rho \boldsymbol{f} \qquad \text{for } x \in \Omega \qquad (2.1a)$$

$$0 = \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) \qquad \text{for } x \in \Omega \qquad (2.1b)$$

$$\boldsymbol{u} = \boldsymbol{g}_D \qquad \text{for } x \in \partial\Omega \qquad (2.1c)$$

where the Laplacian of $\boldsymbol{u}$ is taken component wise, the Dirichlet boundary conditions $\boldsymbol{g}_D$ describe the flow at the domain boundary and the material derivative $\frac{\mathrm{D}\phi}{\mathrm{D}t}$ is defined as

$$\frac{\mathrm{D}\phi}{\mathrm{D}t} := \frac{\partial \phi}{\partial t} + (\boldsymbol{u} \cdot \nabla) \phi$$

The Stokes equations develop under the assumption that the Reynolds number is small [7, Chapter 4.1.3] which can be associated with very slow flow velocities or a high viscosity of the fluid.

$$1 \gg \mathrm{Re} = \frac{\rho u L}{\mu}$$

When assuming a constant characteristic length L and density ρ, it immediately follows that

$$u \ll \mu$$

and thus the inertial forces $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ are much smaller than the viscous forces $\mu \Delta \boldsymbol{u}$ and can be neglected [7, Chapter 4.1.3]. This leads to the instationary Stokes equations

$$\rho \frac{\partial \boldsymbol{u}}{\partial t} = -\nabla p + \mu \Delta \boldsymbol{u} + \rho \boldsymbol{f} \qquad \text{for } x \in \Omega \qquad (2.2a)$$

$$0 = \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) \qquad \text{for } x \in \Omega \qquad (2.2b)$$

$$\boldsymbol{u} = \boldsymbol{g}_D \qquad \text{for } x \in \partial\Omega \qquad (2.2c)$$

It can be seen that the resulting equations are linear and thus easily solvable by standard finite element methods.

In the context of this thesis, only stationary incompressible solutions are sought. Thus we have that $\frac{\partial \boldsymbol{u}}{\partial t} = 0$ and $\rho = \text{const}$. This simplifies the equations further to

$$0 = -\nabla p + \mu \Delta \boldsymbol{u} + \rho \boldsymbol{f} \qquad \text{for } x \in \Omega \qquad (2.3a)$$

$$0 = \nabla \cdot \boldsymbol{u} \qquad \text{for } x \in \Omega \qquad (2.3b)$$

$$\boldsymbol{u} = \boldsymbol{g}_D \qquad \text{for } x \in \partial\Omega \qquad (2.3c)$$

2.2 Function Space

Our function space is intrinsically linked to some possibly nonconforming mesh $\mathcal{M}$. For the discretization of our variational formulation, we will however restrict ourselves to conforming meshes [8, Def. 3.5.2]. Furthermore, we will need the notion of the edge set $\mathcal{E}(\mathcal{M})$, the interior edge set $\mathcal{E}^o(\mathcal{M})$ and the boundary edge set $\mathcal{E}^\partial(\mathcal{M})$.

As the name Discontinuous Galerkin already implies, our finite element space does not consist of continuous functions as in the normal Lagrangian FEM case but rather of piecewise H^1 functions

with discontinuities on the edges of a mesh $\mathcal{M}$. This broken Sobolev space is denoted by $H^1(\mathcal{M})$ which is defined as [13, Eq. 8]:

$$H^1(\mathcal{M}) := \left\{ v \in L^2(\Omega) \mid v|_K \in H^1(K) \; \forall K \in \mathcal{M} \right\} \tag{2.4}$$

The incompressibility constraint in Equation 2.3b is also incorporated into our function space. This greatly simplifies the system of equations corresponding to the discretized PDE. With the proposed finite element space, our problem reduces to finding $\boldsymbol{u} \in V$ such that

$$\begin{cases} 0 = -\nabla p + \mu \Delta \boldsymbol{u} + \rho \boldsymbol{f} & x \in \Omega \\ \boldsymbol{u} = \boldsymbol{g}_D & x \in \partial\Omega \end{cases} \tag{2.5}$$

where V is defined as the intersection of the space of all divergence free functions and $\left(H^1(\mathcal{M})\right)^2$

$$V = \left\{ \boldsymbol{v} \in \left(H^1(\mathcal{M})\right)^2 \mid \nabla \cdot \boldsymbol{v} = 0 \right\} \tag{2.6}$$

Because certainly $\nabla \cdot \boldsymbol{v} = 0 \in L^2(\Omega)$, we know that our function space is a subspace of $H(\mathrm{div}; \Omega) := \left\{ \boldsymbol{v} \in L^2(\Omega; \mathbb{R}^d) \mid \nabla \cdot \boldsymbol{v} \in L^2(\Omega) \right\}$ [12, Def. 20.1] with $d = 2$ and thus the normal component of the functions on the edge set $\mathcal{E}(\mathcal{M})$ is necessarily continuous [3, p. 3].

We also set the fluid flow over the domain boundaries to be equal to zero. Solutions with flow over the boundary will later be computed with the offset function technique described in subsubsection 3.6.2. Consequently, our finite element space is given by

$$V_0 := \left\{ \boldsymbol{v} \in \left(H^1(\mathcal{M})\right)^2 \mid \nabla \cdot \boldsymbol{v} = 0 \text{ in } \Omega \; \wedge \; \boldsymbol{v} \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \right\} \tag{2.7}$$

2.3 Variational Formulation

For the variational formulation we consider only the momentum conservation law Equation 2.3a. The incompressibility constraint is enforced either by using Lagrangian multipliers or by incorporating the constraint into the basis of our function space in which we seek our solution.

Before starting with the derivation of the variational formulation, we introduce a few notations that will make the equations more compact and hopefully readable. For $\boldsymbol{u} : \mathbb{R}^2 \to \mathbb{R}^2$ we define

$$\nabla \boldsymbol{u} = \begin{bmatrix} \partial_x u_1 & \partial_x u_2 \\ \partial_y u_1 & \partial_y u_2 \end{bmatrix}$$

For $\boldsymbol{T}, \boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{n}, \boldsymbol{v} \in \mathbb{R}^2$ we define

$$\boldsymbol{T} : \boldsymbol{\Sigma} = \sum_{i,j=1}^{2} T_{ij}\Sigma_{ij} \quad \text{and} \quad \boldsymbol{n} \cdot \boldsymbol{\Sigma} \cdot \boldsymbol{v} = \sum_{i,j=1}^{2} n_i \Sigma_{ij} v_j$$

For $\boldsymbol{v} \in \mathbb{R}^2$ and $\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\sigma}_1 & \boldsymbol{\sigma}_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}$ we write

$$\boldsymbol{v} \cdot \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{v} \cdot \boldsymbol{\sigma}_1 & \boldsymbol{v} \cdot \boldsymbol{\sigma}_2 \end{bmatrix} \tag{2.8}$$

The challenge when computing the variational formulation of our problem is that the function space V_0 defined in Equation 2.7 contains discontinuous functions. This means that the gradient of the functions in V_0 is not in $L^2(\Omega)$ and thus it is impossible to define the variational formulation of a second order PDE over this function space as it would require evaluating an inner product of two gradients. We solve this problem by performing a local Galerkin discretization on the cells of the mesh without any coupling between neighboring cells. This coupling is later introduced via a penalty term that forces the solution to be continuous [1, p. 1759].

To derive the variational formulation, we start by multiplying Equation 2.3a with a test function $\boldsymbol{v} \in V_0$ and splitting up the integral into a sum of integrals over the elements $K \in \mathcal{M}$.

$$0 = \sum_{K \in \mathcal{M}} \left(-\int_K \nabla p \cdot \boldsymbol{v} \, \mathrm{d}x + \mu \int_K \Delta \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}x + \rho \int_K \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}x \right) \quad \forall \boldsymbol{v} \in V_0 \tag{2.9}$$

This works because the functions are continuous on the mesh cells by the choice of our function space V_0. Next, we partially integrate the first integral involving the pressure term. We obtain that

$$\int_K \nabla p \cdot \boldsymbol{v} \, \mathrm{d}x = -\int_K (\nabla \cdot \boldsymbol{v}) p \, \mathrm{d}x + \int_{\partial K} (\boldsymbol{v} \cdot \boldsymbol{n}) p \, \mathrm{d}x \stackrel{\nabla \cdot \boldsymbol{v} = 0}{=} \int_{\partial K} (\boldsymbol{v} \cdot \boldsymbol{n}) p \, \mathrm{d}x \tag{2.10}$$

The integral over K vanishes because our function space only contains divergence free functions.

The second integral is the normal Laplacian except that it is vector valued. By applying Green's first identity component wise and making use of the notation introduced earlier, we arrive at

$$\int_K \Delta \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}x = -\int_K \nabla \boldsymbol{u} : \nabla \boldsymbol{v} \, \mathrm{d}x + \int_{\partial K} \boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}S \tag{2.11}$$

Inserting these new identities into Equation 2.9 we get the following formulation of the problem:

$$0 = \sum_{K \in \mathcal{M}} \left(\int_K ((-\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho \boldsymbol{f} \cdot \boldsymbol{v}) \, \mathrm{d}x + \int_{\partial K} (\mu \boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{v} - (\boldsymbol{v} \cdot \boldsymbol{n}) p) \, \mathrm{d}x \right) \tag{2.12}$$

We can see that one part of the pressure term already vanished but the other part persists making the equation underdetermined. By rearranging the equation it will however be obvious that the remaining term is also equal to zero. To show this, we can convert the sum over the integrals containing the pressure term to a sum of integrals over the edge set $\mathcal{E}(\mathcal{M})$:

$$\begin{aligned}
\sum_{K \in \mathcal{M}} \int_{\partial K} (\boldsymbol{v} \cdot \boldsymbol{n}) p \, \mathrm{d}x = &\sum_{e \in \mathcal{E}^o(\mathcal{M})} \int_e \left((\boldsymbol{v}^+ \cdot \boldsymbol{n}^+) p^+ - (\boldsymbol{v}^- \cdot \boldsymbol{n}^-) p^- \right) \mathrm{d}x \\
&+ \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \int_e (\boldsymbol{v} \cdot \boldsymbol{n}) p \, \mathrm{d}x
\end{aligned} \tag{2.13}$$

where the superscript + and - denote the value of the variables on the left and right neighboring cells. Because the normal component of $\boldsymbol{v} \in V_0$ is continuous and the exact solution for p itself is also continuous, we get that the sum over the interior edge set $\mathcal{E}^o(\mathcal{M})$ vanishes. This only leaves the integral over the boundary edge set:

$$\sum_{e \in \mathcal{E}^o(\mathcal{M})} \int_e (\boldsymbol{v} \cdot \boldsymbol{n}) p \, \mathrm{d}x = \int_{\partial \Omega} (\boldsymbol{v} \cdot \boldsymbol{n}) p \, \mathrm{d}x = 0 \tag{2.14}$$

where the last equality comes from the fact that by the choice of our finite element space, the flow over the domain boundaries is zero at every point.

All pressure related terms are now removed and the equation reduces to finding $\boldsymbol{u} \in V_0$ such that it holds for all $\boldsymbol{v} \in V_0$ that

$$0 = \sum_{K \in \mathcal{M}} \left(\int_K ((-\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho \boldsymbol{f} \cdot \boldsymbol{v}) \, \mathrm{d}x + \int_{\partial K} \mu \boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}S \right) \tag{2.15}$$

To apply some transformations when introducing the symmetric interior penalty method in subsection 2.4, we want to reformulate the integral over the cell boundaries to an integral over the edge set $\mathcal{E}(\mathcal{M})$. The integral over the boundary of the K can be split into two integrals over the interior edge set $\mathcal{E}^o(\mathcal{M})$ and the boundary edge set $\mathcal{E}^\partial(\mathcal{M})$. For some interior edge $e \in \mathcal{E}^o(\mathcal{M})$, the integral looks as follows

$$\int_e \left(\mu \boldsymbol{n}^+ \cdot \nabla \boldsymbol{u}^+ \cdot \boldsymbol{v}^+ - \mu \boldsymbol{n}^- \cdot \nabla \boldsymbol{u}^- \cdot \boldsymbol{v}^- \right) \mathrm{d}x$$

In the case of a boundary edge e, the integral lacks the contributions from one side of the edge and thus has the following structure

$$\int_e \mu \boldsymbol{n}^+ \cdot \nabla \boldsymbol{u}^+ \cdot \boldsymbol{v}^+ \, \mathrm{d}x$$

This can be expressed more concisely by introducing the jump $[\![\cdot]\!]$ and average $\{\!\{\cdot\}\!\}$ operators. For ϕ some scalar, vector or matrix valued function, we define the jump and average operators over some edge $e \in \mathcal{E}(\mathcal{M})$ similarly to [4, p. 1072] and [1, p. 1756] with the difference that we omit the normal vector in the jump term as this will make the operator more flexible and applicable to our use case.

$$[\![\phi]\!] = \begin{cases} \phi^+ - \phi^- & \text{for } e \in \mathcal{E}^o(\mathcal{M}) \\ \phi^+ & \text{for } e \in \mathcal{E}^\partial(\mathcal{M}) \end{cases} \tag{2.16a}$$

$$\{\!\{\phi\}\!\} = \begin{cases} \frac{\phi^+ + \phi^-}{2} & \text{for } e \in \mathcal{E}^o(\mathcal{M}) \\ \phi^+ & \text{for } e \in \mathcal{E}^\partial(\mathcal{M}) \end{cases} \tag{2.16b}$$

Using these operators, we can rewrite the integral over the cell boundaries of the mesh in Equation 2.15 as a sum over integrals over all edges

$$\sum_{K \in \mathcal{M}} \int_{\partial K} \boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{v} \, \mathrm{d}x = \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e [\![\boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{v}]\!] \, \mathrm{d}x \tag{2.17}$$

Inserting this identity into Equation 2.15, we get that we have to find $\boldsymbol{u} \in V_0$ such that it holds for all $\boldsymbol{v} \in V_0$ that

$$0 = \sum_{K \in \mathcal{M}} \int_K (-\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho \boldsymbol{f} \cdot \boldsymbol{v}) \, \mathrm{d}x + \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \mu [\![\boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot v]\!] \, \mathrm{d}x \tag{2.18}$$

Next, we split the jump of the products into multiple simpler expressions such that each jump or average operator only contains an expression dependent on either the flow velocity $\boldsymbol{u}$ or the test function $\boldsymbol{v}$ but not both at the same time. This will be of great use when deriving the system matrix for the discretized problem. A simple calculation [1, Eq. 3.3] shows that

$$[\![\boldsymbol{n} \cdot \nabla \boldsymbol{u} \cdot \boldsymbol{v}]\!] = [\![\boldsymbol{n} \cdot \nabla \boldsymbol{u}]\!] \cdot \{\!\{\boldsymbol{v}\}\!\} + \{\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{u}\}\!\} \cdot [\![\boldsymbol{v}]\!] \tag{2.19}$$

We apply Equation 2.19 to the formulation of our problem and get

$$\begin{aligned} 0 = & \sum_{K \in \mathcal{M}} \int_K (-\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho \boldsymbol{f} \cdot \boldsymbol{v}) \, \mathrm{d}x \\ & + \mu \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e ([\![\boldsymbol{n} \cdot \nabla \boldsymbol{u}]\!] \cdot \{\!\{\boldsymbol{v}\}\!\} + \{\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{u}\}\!\} \cdot [\![\boldsymbol{v}]\!]) \, \mathrm{d}x \end{aligned} \tag{2.20}$$

If we assume that $p \in H^1(\Omega)$ and $\boldsymbol{f} \in L^2(\Omega)$ it follows from the elliptic lifting theorem on convex domains [8, Thm. 5.4.2] that $\boldsymbol{u} \in (H^2(\Omega))^2$ causing the gradient of the exact solution of $\boldsymbol{u}$ to be continuous. The variational formulation further reduces to finding $\boldsymbol{u} \in V_0$ such that for all $\boldsymbol{v} \in V_0$ it holds that

$$0 = \sum_{K \in \mathcal{M}} \int_K (-\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho \boldsymbol{f} \cdot \boldsymbol{v}) \, \mathrm{d}x + \mu \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \{\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{u}\}\!\} \cdot [\![\boldsymbol{v}]\!] \, \mathrm{d}x \tag{2.21}$$

2.4 Symmetric Interior Penalty Discontinuous Galerkin

Note that we can add arbitrary terms to Equation 2.21 without affecting the solution given that the added terms are zero for the exact solution of the PDE. For the Symmetric Interior Penalty Discontinuous Galerkin method, we add terms to make the bilinear form symmetric and additionally add a penalty term for the stabilization of the problem [1, p. 1759]:

$$
- \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!] [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x + \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \int_e \frac{\sigma}{|e|} (\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e) [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \tag{2.22}
$$

where $\boldsymbol{\tau}_e$ denotes the tangential vector of unit length onto the edge e. This penalty term is zero for all $\boldsymbol{v} \in V_0$ if and only if the function $\boldsymbol{u} \in V_0$ is continuous and fulfills the Dirichlet boundary conditions. We only need to penalize tangential discontinuities because our functions are in $H(\mathrm{div}; \Omega)$ and thus the normal component is already continuous. This penalty term also introduces a coupling between the different cells of the mesh by being nonzero for all solutions in the kernel of the non-stabilized variational formulation except the exact one.

The only term causing the variational formulation to not be symmetric is the integral over $\{\!\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{u}\}\!\!\} \cdot [\![\boldsymbol{v}]\!]$. We know that for the exact solution $\boldsymbol{u} \in H^1(\Omega) \subset \mathcal{C}^0(\Omega)$. This results in the jump of $\boldsymbol{u}$ being zero everywhere and we can add $\{\!\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{v}\}\!\!\} \cdot [\![\boldsymbol{u}]\!]$ to the non-symmetric term without altering the solution of the variational formulation. By doing this, we arrive at the symmetric interior penalty variational formulation of our PDE:

$$
\begin{aligned}
0 = & \sum_{K \in \mathcal{M}} \int_K (-\mu \nabla \boldsymbol{u} : \nabla \boldsymbol{v} + \rho \boldsymbol{f} \cdot \boldsymbol{v}) \, \mathrm{d}x \\
& + \mu \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e (\{\!\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{u}\}\!\!\} \cdot [\![\boldsymbol{v}]\!] + \{\!\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{v}\}\!\!\} \cdot [\![\boldsymbol{u}]\!]) \, \mathrm{d}x \\
& - \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!] [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \\
& + \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \frac{\sigma}{|e|} \int_e (\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e) [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x
\end{aligned} \tag{2.23}
$$

We can write this more compactly as

$$
\boldsymbol{u} \in V_0 : \ a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{u}, \boldsymbol{v}) + \kappa(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}) + k(\boldsymbol{v}) \ \ \forall \boldsymbol{v} \in V_0 \tag{2.24}
$$

where $a(\boldsymbol{u}, \boldsymbol{v})$, $b(\boldsymbol{u}, \boldsymbol{v})$, $\kappa(\boldsymbol{u}, \boldsymbol{v})$, $l(\boldsymbol{v})$ and $k(\boldsymbol{v})$ are defined as

$$
a(\boldsymbol{u}, \boldsymbol{v}) := \mu \sum_{K \in \mathcal{M}} \int_K \nabla \boldsymbol{u} : \nabla \boldsymbol{v} \, \mathrm{d}x \tag{2.25a}
$$

$$
b(\boldsymbol{u}, \boldsymbol{v}) := -\mu \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e (\{\!\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{u}\}\!\!\} \cdot [\![\boldsymbol{v}]\!] + \{\!\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{v}\}\!\!\} \cdot [\![\boldsymbol{u}]\!]) \, \mathrm{d}x \tag{2.25b}
$$

$$
\kappa(\boldsymbol{u}, \boldsymbol{v}) := \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!] [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \tag{2.25c}
$$

$$
l(\boldsymbol{v}) := \rho \sum_{K \in \mathcal{M}} \int_K \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}x \tag{2.25d}
$$

$$
k(\boldsymbol{v}) := \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \int_e \frac{\sigma}{|e|} (\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e) [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \tag{2.25e}
$$

3 Discretization

When discretizing the variational formulation naively, the jump and average terms for a cell would need the function values and DOFs on the neighboring cells. This makes the local assembly of the system matrix impossible. The challenge for the discretization step is to construct the system matrix in such a way that local assembly is still possible. In this section, we derive a structure for the global system matrix which is suited for local assembly.

3.1 Local Assembly

When performing local assembly, the cells of the mesh must be able to generate element matrices without needing function values nor contribute to DOFs from neighboring cells. This makes it impossible to incorporate the jump and average terms over the edges directly into the system matrix. The solution is to introduce artificial degrees of freedom on the edges of the mesh containing information from which the jump terms can be computed. The system matrix consequently is extended by the additional DOFs and the final system of equations has the following structure:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B}^T & \boldsymbol{K}^T \\ \boldsymbol{B} & \tilde{\boldsymbol{B}} & \boldsymbol{0} \\ \boldsymbol{K} & \boldsymbol{0} & \tilde{\boldsymbol{K}} \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{u}_b \\ \boldsymbol{u}_\kappa \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi} \\ \boldsymbol{\phi}_b \\ \boldsymbol{\phi}_\kappa \end{bmatrix} \tag{3.1}$$

where the submatrix $\boldsymbol{A}$ corresponds to the bilinear form $a(\boldsymbol{u}, \boldsymbol{v})$, the submatrices $\boldsymbol{B}$, $\boldsymbol{B}^T$ and $\tilde{\boldsymbol{B}}$ are used to compute the viscous term in the bilinear form $b(\boldsymbol{u}, \boldsymbol{v})$ and the submatrices $\boldsymbol{K}$, $\boldsymbol{K}^T$ and $\tilde{\boldsymbol{K}}$ will add the penalty term $\kappa(\boldsymbol{u}, \boldsymbol{v})$.

3.2 Finite Element Space

For simplicity, the PDE is only solved on triangular meshes $\mathcal{M}$. As a basis for the finite element space we chose the Raviart-Thomas basis functions $\mathrm{RT}_p(\mathcal{M})$ which are vector valued divergence conforming finite elements such that their divergence is a polynomial of degree p [9, Section 3.4.1]. They also have some useful properties which leads to considerably smaller systems of equations in comparison to using fully discontinuous function spaces. These properties are discussed in the rest of this subsection.

Because the divergence of functions in RT_p are polynomials of order p, the function space itself is spanned by polynomials of order $p + 1$. The d-dimensional local basis functions for the Raviart-Thomas elements on a triangle T are defined by [9, Eq. 3.19]

$$\mathrm{RT}_p(T) = \mathcal{P}_p(T)^d + \boldsymbol{x}\mathcal{P}_p(T) \tag{3.2}$$

where $\mathcal{P}_p$ denotes the space of all polynomials of order up to and including p. They have the useful property of being $H(\mathrm{div})$-conforming finite elements and thus have a continuous normal component on the mesh edges while not necessarily having a continuous tangential component [9, Section 3.4].
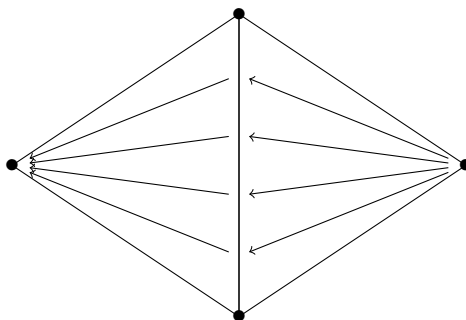

Figure 1: A basis function of RT_0 with its DOF on the center edge

3.3 Local Contributions

The local contributions $a(\boldsymbol{u}, \boldsymbol{v})$ and $l(\boldsymbol{v})$ given in Equation 2.25a and Equation 2.25d originate from the standard Galerkin discretization on the mesh cells. Therefore we can discretize them the same way as in continuous Galerkin. We define the matrix elements of $\boldsymbol{A}$ as

$$A_{ij} = \mu \sum_{K \in \mathcal{M}} \int_K \nabla \boldsymbol{b}_i : \nabla \boldsymbol{b}_j \, \mathrm{d}x \tag{3.3}$$

And similarly, we define the elements of the vector $\boldsymbol{\phi}$ as

$$\phi_n = \rho \sum_{K \in \mathcal{M}} \int_K \boldsymbol{f} \cdot \boldsymbol{b}_n \, \mathrm{d}x \tag{3.4}$$

3.4 Penalty Term

We incorporate the penalty term $\kappa(\boldsymbol{u}, \boldsymbol{v}) - k(\boldsymbol{v})$ given in Equation 2.25c and Equation 2.25e into the system matrix. by making use of the artificial degrees of freedom on the mesh and transforming the penalty term in such a way that it can be written as a linear system of equations in matrix form.

The penalty term for some test function $\boldsymbol{v}$ is given by

$$\kappa(\boldsymbol{u}, \boldsymbol{v}) - k(\boldsymbol{v}) = \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!][\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x - \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \int_e \frac{\sigma}{|e|} (\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e) [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \tag{3.5}$$

The traces of $\boldsymbol{u}$ and $\boldsymbol{v}$ are polynomials of order $p+1$ where p is the order of our Raviart-Thomas basis functions. Thus, the integral over the n-th edge e_n can be replaced by a quadrature rule of sufficiently high order with N quadrature points $\{x_i^n\}_{i=1}^N$ and weights $\{w_i^n\}_{i=1}^N$:

$$\kappa(\boldsymbol{u}, \boldsymbol{v}) = \sum_{n=1}^{|\mathcal{E}(\mathcal{M})|} \sum_{i=1}^N \frac{\sigma}{|e_n|} w_i^n [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_{e_n}]\!]\Big|_{x_i^n} [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_{e_n}]\!]\Big|_{x_i^n} \tag{3.6a}$$

$$k(\boldsymbol{v}) = \sum_{n=1}^{|\mathcal{E}^\partial(\mathcal{M})|} \sum_{i=1}^N \frac{\sigma}{|e_n|} w_i^n (\boldsymbol{g}_D \cdot \boldsymbol{\tau}_{e_n})\Big|_{x_i^n} [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_{e_n}]\!]\Big|_{x_i^n} \tag{3.6b}$$

This reduces the expression for the penalty term to a linear combination of $N|\mathcal{E}(\mathcal{M})|$ scalar values which is equivalent to the inner product of two vectors. We thus introduce N auxiliary DOFs per edge of the mesh which contain the jump terms of each edge evaluated at the quadrature points x_i^n.

By the choice of our system matrix in subsection 3.1, we know that for any solution vector $\boldsymbol{u}$ to the discretized problem, the penalty term added is forced to be

$$[\kappa(\boldsymbol{u}, \boldsymbol{b}_i) - k(\boldsymbol{b}_i)]_{i=1}^N = \boldsymbol{K}^T \underbrace{\left(-\tilde{\boldsymbol{K}}^{-1} \left(\boldsymbol{K}\boldsymbol{u} - \boldsymbol{\phi}_\kappa \right) \right)}_{\boldsymbol{u}_\kappa} \tag{3.7}$$

where N is the number of basis functions. This can be seen by applying one step of a block Gaussian elimination to the system matrix Equation 3.1.

The terms in Equation 3.6a involving the test function $\boldsymbol{v}$ are a part of the system matrix while the other terms involving $\boldsymbol{u}$ are part of the solution vector. For this reason, we extend the i-th row of the system matrix corresponding to the test function $\boldsymbol{v} = \boldsymbol{b}_i$ with the following terms:

$$\left(\boldsymbol{K}^T \right)_{ij} = \left[\!\left[\boldsymbol{b}_i \cdot \boldsymbol{\tau}_{e_{\lceil j/N \rceil}} \right]\!\right]\Big|_{x_{j\%N}^{\lceil j/N \rceil}} \tag{3.8}$$

We know that $\boldsymbol{K}$ maps vectors of basis function coefficients to vectors corresponding to the jumps over the edges at the quadrature nodes. We can thus extend the system matrix $\boldsymbol{A}$ with the matrix

$\boldsymbol{K}$ and a suitable choice of a lower right block to force the solution vector components to be the jumps of $\boldsymbol{u}$ multiplied with the factor $-\frac{\sigma}{|e_n|}w_i^n$. This is achieved by choosing the elements of matrix $\tilde{\boldsymbol{K}}$ as

$$\tilde{K}_{ij} = -\frac{|e_{\lceil i/N \rceil}|}{\sigma w_{i\%N}^{\lceil i/N \rceil}}\delta_{ij} \tag{3.9}$$

where δ_{ij} is the Kronecker delta.

The remaining term is the linear part $k(\boldsymbol{v})$ in Equation 3.6b of the penalty term. For a specific test function $\boldsymbol{v}$, this linear part causes the penalty term in Equation 3.6a to be affine in $\boldsymbol{u}$ as opposed to linear. For any given $\boldsymbol{v} \in V_0$ it acts as an offset to the bilinear form $\kappa(\boldsymbol{u}, \boldsymbol{v})$. Thus we can identify it with $\boldsymbol{\phi}_\kappa$ in Equation 3.7 and choose it as

$$(\boldsymbol{\phi}_\kappa)_i = \begin{cases} \left(\boldsymbol{g}_D \cdot \boldsymbol{\tau}_{e_{\lceil i/N \rceil}}\right)\Big|_{x_{i\%N}} & \text{for } e_{\lceil i/N \rceil} \in \mathcal{E}^\partial(\mathcal{M}) \\ 0 & \text{otherwise} \end{cases} \tag{3.10}$$

3.5 Viscous Term

The viscous term given by

$$b(\boldsymbol{u}, \boldsymbol{v}) = -\mu \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e (\{\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{u}\}\!\} \cdot [\![\boldsymbol{v}]\!] + \{\!\{\boldsymbol{n} \cdot \nabla \boldsymbol{v}\}\!\} \cdot [\![\boldsymbol{u}]\!]) \, \mathrm{d}x \tag{3.11}$$

in Equation 2.25b is incorporated into the system matrix analogously to the penalty term.

First, we eliminate the integral in Equation 3.11. This is achieved by replacing it with a numerical quadrature of sufficiently high order:

$$b(\boldsymbol{u}, \boldsymbol{v}) = -\mu \sum_{n=1}^{|\mathcal{E}(\mathcal{M})|} \sum_{i=1}^{N} w_i^n \left(\{\!\{\boldsymbol{n}_{e_n} \cdot \nabla \boldsymbol{u}\}\!\}\big|_{x_i^n} \cdot [\![\boldsymbol{v}]\!]\big|_{x_i^n} + \{\!\{\boldsymbol{n}_{e_n} \cdot \nabla \boldsymbol{v}\}\!\}\big|_{x_i^n} \cdot [\![\boldsymbol{u}]\!]\big|_{x_i^n}\right) \tag{3.12}$$

Equation 3.12 is now a linear combination of finitely many terms and can thus be written in matrix form.

We will build the matrices $\boldsymbol{B}$ and $\tilde{\boldsymbol{B}}$ from submatrices called $\boldsymbol{J}$, $\boldsymbol{M}$ and $\boldsymbol{T}$ where $\boldsymbol{J}$ is responsible for the jump terms, $\boldsymbol{M}$ is responsible for the average terms and $\boldsymbol{T}$ is responsible for combining the two

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{J} \\ \boldsymbol{M} \end{bmatrix} \tag{3.13a}$$

$$\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{T}^T \\ \boldsymbol{T} & \boldsymbol{0} \end{bmatrix} \tag{3.13b}$$

The matrix $\boldsymbol{J}$ is built in such a way that the $2n$-th and $(2n+1)$-th rows map the basis function coefficients to the x- respectively the y-component of the jump at edge $e_{\lceil n/N \rceil}$ and quadrature node $x_{n\%N}^{\lceil n/N \rceil}$. This means that

$$J_{2i:2i+1,j} = [\![\boldsymbol{b}_i]\!]\big|_{x_{j\%N}^{\lceil j/N \rceil}} \tag{3.14}$$

The matrix $\boldsymbol{M}$ is built similarly except that the average expression is taken instead of the jump expression:

$$M_{2i:2i+1,j} = \left\{\!\!\left\{\boldsymbol{n}_{e_{\lceil i/2N \rceil}} \cdot \nabla \boldsymbol{b}_i\right\}\!\!\right\}\Big|_{x_{j\%N}^{\lceil j/N \rceil}} \tag{3.15}$$

The matrix $\boldsymbol{T}$ is defined similarly as the matrix $\tilde{\boldsymbol{K}}$ in the discretization of the penalty term $\kappa(\boldsymbol{u}, \boldsymbol{v})$ and thus is also a matrix with the quadrature weights on the main diagonal:

$$T_{ij} = \frac{1}{w_{i\%2N}^{\lceil i/2N \rceil}}\delta_{ij} \tag{3.16}$$

The viscous term in Equation 3.12 is linear in $\boldsymbol{u}$. Therefore, the right hand side vector $\boldsymbol{\phi}_b$ becomes

$$\boldsymbol{\phi}_b = 0 \tag{3.17}$$

3.6 Boundary Conditions

The solver differentiates between two different boundary conditions: No-Flow where the normal component of the flow velocity is zero on the domain boundary and In-Out-Flow where fluid can flow in and out of the domain. Theoretically, the solver for the In-Out-Flow boundary conditions could also solve the problems with No-Flow boundary conditions. Those can however be implemented more efficiently as shown in the following subsections.

3.6.1 No-Flow

Regardless of whether we enforce the incompressibility constraint using Lagrangian multipliers or incorporate it directly into our function space basis, the basis functions have the property that only basis functions associated with entities on the mesh boundary contribute to the flux over $\partial\Omega$. Thus, to enforce a normal component of zero over the mesh boundaries it is sufficient to set all DOFs on the mesh boundary to zero. This of course does not include the auxiliary DOFs on the boundary as those are used to weakly impose the tangential component of the Dirichlet boundary conditions.

3.6.2 In-Out-Flow

For fluid flows over the boundary of our domain, we employ the offset function technique described in [8, Section 3.7.6]. We denote by V_0 our finite element space with the normal component set to zero over the whole domain boundary given in Equation 2.7. This is the same space we use for the no-flow boundary conditions.

We now define a new affine space as $V = \boldsymbol{u}_0 + V_0$ where $\boldsymbol{u}_0$ is some arbitrary function having the same normal component over the domain boundary as the Dirichlet boundary conditions $\boldsymbol{g}_D$. The variational problem then has the following form

$$\boldsymbol{u} \in V_0: \ a(\boldsymbol{u} + \boldsymbol{u}_0, \boldsymbol{v}) = l(\boldsymbol{v}) \ \forall \boldsymbol{v} \in V_0 \tag{3.18}$$

Due to the bilinearity of a, we can split it into the two components $a(\boldsymbol{u}, \boldsymbol{v})$ and $a(\boldsymbol{u}_0, \boldsymbol{v})$. The latter is only dependent on the test function $\boldsymbol{v}$ and can thus be incorporated into the linear form yielding the variational problem

$$\boldsymbol{u} \in V_0: \ a(\boldsymbol{u}, \boldsymbol{v}) = l(\boldsymbol{v}) - a(\boldsymbol{u}_0, \boldsymbol{v}) \ \forall \boldsymbol{v} \in V_0 \tag{3.19}$$

This can now be solved using the same system matrix as for the no-flow case.

4 Implementation

Our implementation uses the C++ finite element library LehrFEM++ [10]. The use of this library reduces our programming effort to only having to provide an element matrix and vector provider as well as some auxiliary functions to compute the offset function and analyze the obtained solutions. The implementations of these are discussed in the rest of this section.

4.1 Finite Element Space

For the implementation we choose to incorporate the incompressibility constraint directly into the basis of our discretized function space instead of using Lagrangian multipliers and thus get a different set of basis functions than the Raviart-Thomas basis functions

$$\mathrm{RT}_p \cap \left\{ \boldsymbol{v} \in L^2(\Omega) \mid \nabla \cdot \boldsymbol{v} = 0 \text{ in } \Omega \ \wedge \ \boldsymbol{v} \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \right\} = \nabla \times \mathcal{S}_{p+1,0}^0 \tag{4.1}$$

where the space $\mathcal{S}_{p,0}^0$ is defined as the space of all C^0-splines of degree p [8, Section 3.4.2]

$$\mathcal{S}_{p,0}^0 := \left\{ f \in C^0(\Omega) \mid f\big|_K \in \mathcal{P}_p, K \in \mathcal{M} \ \wedge \ f\big|_{\partial\Omega} \equiv 0 \right\}$$

and the curl denotes the 2D-curl which is for some function $\Psi : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R}$ defined as the rotated gradient

$$\nabla \times \Psi := \begin{bmatrix} \partial_y \Psi \\ -\partial_x \Psi \end{bmatrix} \tag{4.2}$$

An example for a basis function of this space for the case $p = 0$ is drawn in Figure 2. Note that it is in fact discontinuous as it is piecewise constant and nonzero on the triangles making up the support of the function. This implies a jump of the function on the boundary of the support.
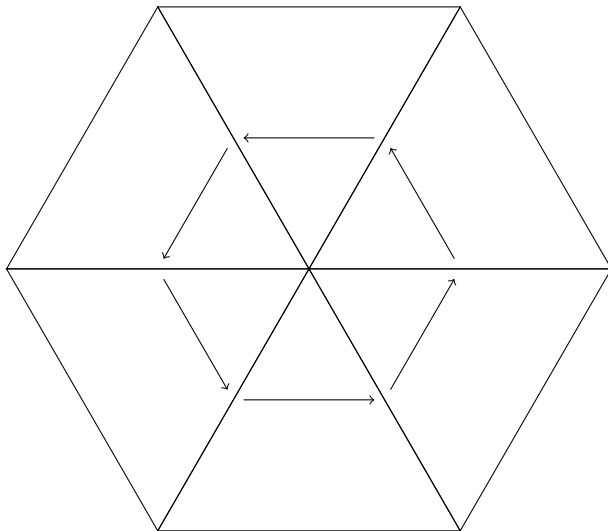
Figure 2: A basis function of $\nabla \times \mathcal{S}_{1,0}^0$ with its DOF on the center vertex. One can see the discontinuities of the tangential component across each edge.

These basis functions inherit the useful properties from the Raviart-Thomas elements because - as they are in RT_p - they are just linear combinations of Raviart-Thomas elements. Consequently, the normal component of the functions in our finite element space over the edges of the mesh is still continuous. And in particular, the flow over the boundary of the support of a basis function is equal to zero. However, they have the major downside that the basis is only valid for simply connected domains. This will be demonstrated and proven in subsection 5.6.

For simplicity, our implementation only supports the zeroth order FE-space given by

$$V_{0,h} = \mathrm{RT}_0 \cap \left\{ \boldsymbol{v} \in L^2(\Omega) \mid \nabla \cdot \boldsymbol{v} = 0 \text{ in } \Omega \ \wedge \ \boldsymbol{v} \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega \right\} = \nabla \times \mathcal{S}_{1,0}^0 \tag{4.3}$$

The local basis functions are consequently implemented by taking the rotated gradient of the first order linear Lagrangian finite elements already implemented in the LehrFEM++ class `lf::uscalfe::FeLagrangeO1Tria<double>`.

4.2 Element Matrices

As the finite element space is piecewise constant, all terms involving the gradient of the basis functions are equal to zero reducing our variational formulation to

$$\boldsymbol{u}_h \in V_{0,h}: \ \kappa(\boldsymbol{u}_h, \boldsymbol{v}_h) = l(\boldsymbol{v}_h) + k(\boldsymbol{v}_h) - \kappa(\boldsymbol{u}_0, \boldsymbol{v}_h) \ \ \forall \boldsymbol{v}_h \in V_{0,h} \tag{4.4}$$

where $\boldsymbol{u}_0$ is our offset function. Consequently, we can use a quadrature rule with a single point for the penalty term and the parts of the LSE corresponding to the viscous terms vanish. This reduces the system matrix to

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{K}^T \\ \boldsymbol{K} & \tilde{\boldsymbol{K}} \end{bmatrix} \tag{4.5}$$

11

The element matrices are therefore also divided into four blocks with each of them contributing to the corresponding block in the full system matrix.

$$\tilde{A}_e = \begin{bmatrix} A_e & K_e^T \\ K_e & \tilde{K}_e \end{bmatrix} \tag{4.6}$$

As the basis functions are piecewise constant, the upper left block of the element matrix corresponding to the Laplacian is equal to zero.

The jump operator is linear and can thus easily be assembled from the jump of the function on the triangle to some reference value provided we can somehow let the tangential vectors to an edge viewed from the two adjacent cells point in opposite directions. An illustration of this assembly is provided in Figure 3. This is implemented by taking the rotated outward pointing normals as tangential vectors.
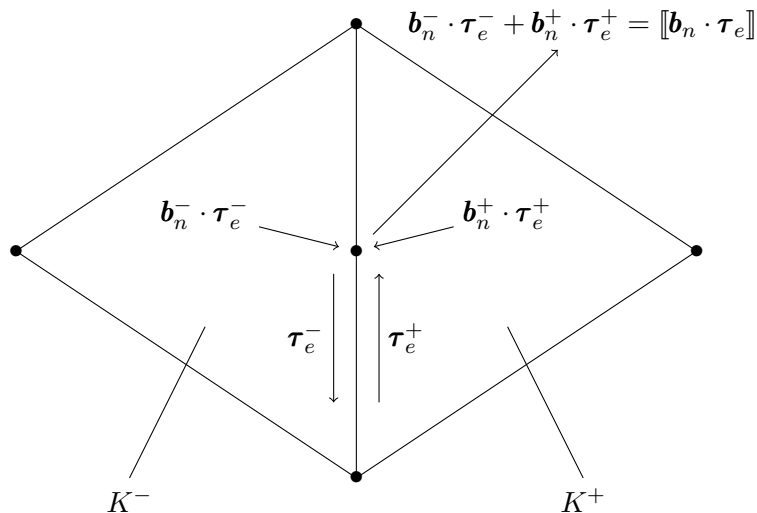
Figure 3: Assembly of the jump term on the auxiliary DOF between K^+ and K^-

According to Equation 3.9, the i-th diagonal element of the lower right block of the full system matrix must equal $-\frac{1}{\sigma}$. To accomplish that, we need to differentiate between boundary edges and inner edges, as the former have only one adjacent triangle instead of two. The diagonal elements of $\tilde{K}$ are the sum of diagonal elements of two $\tilde{K}_e$ for interior edges and they are equal to the diagonal elements of some $\tilde{K}_e$ for boundary edges.

In summary, the four blocks of the element matrices are given by

$$\begin{aligned} A_e &= 0 \\ (K_e)_{ij} &= b_i \cdot \tau_j \\ \left(\tilde{K}_e\right)_{ii} &= \begin{cases} -\frac{1}{\sigma} & \text{for boundary edges} \\ -\frac{1}{2\sigma} & \text{for interior edges} \end{cases} \end{aligned} \tag{4.7}$$

where b_i is the i-th local basis function, e_i is the i-th edge of the current element and τ_i is the local tangential vector onto the i-th edge of the triangle.

4.3 Element Vectors

The structure of the element vectors is determined by our structural choice of the system matrix

$$\phi_K = \begin{bmatrix} \phi_{K,A} \\ \phi_{K,\kappa} \end{bmatrix} \tag{4.8}$$

Contrary to the system matrix, the element vectors need no special cases due to auxiliary DOFs or boundary edges. Their elements are therefore given by

$$(\boldsymbol{\phi}_{K,A})_i = \int_K \boldsymbol{f} \cdot \boldsymbol{b}_i \, \mathrm{d}x \tag{4.9a}$$

$$(\boldsymbol{\phi}_{K,\kappa})_i = \begin{cases} \left(\boldsymbol{g}_D \cdot \boldsymbol{\tau}_{e_{\lceil i/N \rceil}}\right)\Big|_{x_{i\%N}} \left[\!\left[\boldsymbol{b}_i \cdot \boldsymbol{\tau}_{e_{\lceil i/N \rceil}}\right]\!\right]\Big|_{x_{i\%N}} & \text{for } e_{\lceil i/N \rceil} \in \mathcal{E}^\partial(\mathcal{M}) \\ 0 & \text{otherwise} \end{cases} \tag{4.9b}$$

The reason this formula for the local contribution is exactly the same as the one for the global contribution is that the auxiliary DOFs on the boundary only have a single cell contributing to them.

4.4 Offset Function

The offset function is represented by a vector $\boldsymbol{u}_0$ of basis function coefficients such that the normal component of the flow over the domain boundary is equal to the normal component of the Dirichlet boundary conditions $\boldsymbol{g}_D \cdot \boldsymbol{n}$.

We denote by $\{\boldsymbol{b}_1^\partial, \boldsymbol{b}_2^\partial, \cdots, \boldsymbol{b}_K^\partial\}$ the set of all basis functions contributing to fluid flow over the boundary. The matrix $\boldsymbol{F} \in \mathbb{R}^{K \times K}$ mapping basis function coefficients to the normal component of the flow over the domain boundaries is then given by

$$F_{ij} = \boldsymbol{b}_j^\partial \cdot \boldsymbol{n}_{e_i} \tag{4.10}$$

By defining the vector $\tilde{\boldsymbol{g}} \in \mathbb{R}^K$ as

$$\tilde{g}_i = \boldsymbol{g}_D \cdot \boldsymbol{n}_{e_i} \tag{4.11}$$

we are able to solve for the basis function coefficients $\boldsymbol{u}_0$ fulfilling the Dirichlet conditions $\boldsymbol{g}_D$ by computing

$$\boldsymbol{u}_0 = \boldsymbol{F}^{-1} \tilde{\boldsymbol{g}}_D \tag{4.12}$$

To build the matrix $\boldsymbol{F}$, we construct two `lf::assemble::DynamicFEDofHandler`. One has one DOF for each boundary node while the other has one DOF for each boundary edge. The assembly of $\boldsymbol{F}$ is now very similar to the assembly of the matrix $\boldsymbol{K}$ in the system matrix except one has to pay attention to which nodes are boundary nodes in order to assemble the element matrices correctly.

Once we have assembled this matrix, we only need to invert it to get the basis function coefficients for our offset function.

4.5 Considerations for Higher Order Implementations

As soon as we have at least linear basis functions, the quadrature rules need more than a single quadrature point and consequently we have more than one auxiliary DOF per edge. As a consequence, we need a global ordering of DOFs on the edges in order for the jump and average terms to be assembled correctly. This need for a global ordering is depicted in Figure 4 where the indices at the corners denote the local indexing of the vertices and the indices next to the auxiliary DOFs on the edge denote their local indexing. When assembling the system matrix, the n-th term of the sum for the numerical quadrature is associated with the n-th auxiliary DOF on the edge. However, in the demonstrated case of an inconsistent DOF numbering, the contributions from the two triangles will be mixed and thus provide incorrect results.

To solve this, we can make use of the `lf::mesh::Entity::RelativeOrientations()` method provided by LehrFEM++ and permute the local auxiliary DOFs in the element matrix in such a way that we get a consistent global ordering.
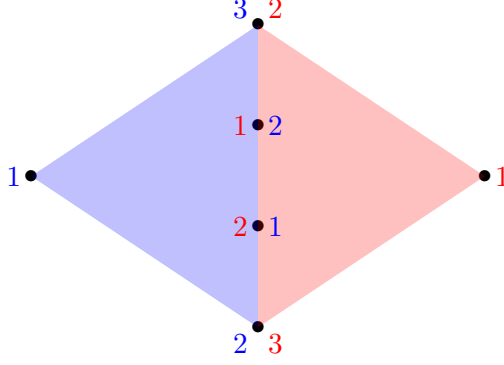
Figure 4: Auxiliary DOF numbering demonstrating the problem occuring with higher order basis functions. If this inconsistent DOF numbering is not handled, the values of the edge DOFs are potentially assembled from two contributions to different auxiliary DOFs.

5 Convergence Studies

In this section, we state some analytical results on the convergence properties of our algorithm. These are then verified by five numerical experiments. The experiments are purposefully designed to demonstrate the limitations of using $\nabla \times \mathcal{S}_{1,0}^0$ as a finite element space and empirically determine the convergence rate in the case of h-refinement. The meshes are either generated by mesh builders in LehrFEM++ or by GMSH [6]. As we found in subsection 5.6 that the numerical solution to the PDE seems to be dependent on the mesh used, we additionally provide all convergence plots with a modified penalty term given by

$$\kappa'(\boldsymbol{u}, \boldsymbol{v}) := \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \sigma [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!] [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \tag{5.1a}$$

$$k'(\boldsymbol{v}) := \sum_{e \in \mathcal{E}^{\partial}(\mathcal{M})} \int_e \sigma (\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e) [\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \tag{5.1b}$$

as an attempt to narrow down the cause of this mesh dependency.

5.1 Analytical Results

The DG-norm is defined as [5, p. 69]

$$||\boldsymbol{v}||_{1,h}^2 := \sum_{K \in \mathcal{M}} ||\nabla \boldsymbol{v}||_{0,K}^2 + \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{1}{|e|} ||[\![\boldsymbol{v}]\!]||_{0,e}^2 \, \mathrm{d}x \tag{5.2}$$

Proposition 1 (Convergence in the DG norm). *The error in the DG-norm for functions in $\nabla \times \mathcal{S}_{1,0}^0$ can be bounded from below by the H^1-seminorm of the exact solution.*

Proof.

$$
\begin{aligned}
||\boldsymbol{u} - \boldsymbol{u}_h||_{1,h}^2 &= \sum_{K \in \mathcal{M}} ||\nabla (\boldsymbol{u} - \boldsymbol{u}_h)||_{0,K}^2 + \sum_{e \in \mathcal{E}(\mathcal{M})} \frac{1}{|e|} ||[\![\boldsymbol{u} - \boldsymbol{u}_h]\!]||_{0,e}^2 \\
&= \sum_{K \in \mathcal{M}} ||\nabla \boldsymbol{u} - \underbrace{\nabla \boldsymbol{u}_h}_{=0}||_{0,K}^2 + \sum_{e \in \mathcal{E}(\mathcal{M})} \frac{1}{|e|} ||\underbrace{[\![\boldsymbol{u}]\!]}_{=0} - [\![\boldsymbol{u}_h]\!]||_{0,e}^2 \\
&= \sum_{K \in \mathcal{M}} ||\nabla \boldsymbol{u}||_{0,K}^2 + \sum_{e \in \mathcal{E}(\mathcal{M})} \frac{1}{e} ||[\![\boldsymbol{u}_h]\!]||_{0,e}^2 \\
&\geq \sum_{K \in \mathcal{M}} ||\nabla \boldsymbol{u}||_{0,K}^2 \\
&= |\boldsymbol{u}|_1
\end{aligned}
\tag{5.3}
$$

14

As the remaining terms are not dependent on the mesh resolution, this concludes our proof. □

Proposition 2 (Existence of Solutions). *Let $\Omega \subset \mathbb{R}^2$ be a compact domain which is not simply connected. Let $\boldsymbol{u} \in RT_p$ be the solution of the PDE. Then it does not necessarily hold that $\boldsymbol{u} \in V_h = \nabla \times \mathcal{S}_{p+1,0}^0$.*

Proof. Because $V_h = \nabla \times \mathcal{S}_{p+1,0}^0$, we know that there exists a potential $\Psi \in \mathcal{S}_{p+1,0}^0$ for all $\boldsymbol{u} \in V_h$ such that $\boldsymbol{u} = \nabla \times \Psi$. Due to this choice of our function space, the potential is zero at the domain boundaries. Let $\boldsymbol{\gamma} : [0,1] \to \Omega$ be a curve such that $\boldsymbol{\gamma}(0) \in \partial\Omega$ and $\boldsymbol{\gamma}(1) \in \partial\Omega$. Let the solution to the PDE $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ be the 2D curl of the potential $\Psi : \Omega \to \mathbb{R}$ with $\Psi(\boldsymbol{x}) = 0 \; \forall \boldsymbol{x} \in \partial\Omega$. We get that the net flow across $\boldsymbol{\gamma}$ is given by

$$
\begin{aligned}
\int_{\boldsymbol{\gamma}} \boldsymbol{u}(\boldsymbol{\xi}) \cdot \mathrm{d}\boldsymbol{\xi}^t &= \int_{\boldsymbol{\gamma}} \nabla \times \Psi(\boldsymbol{\xi}) \cdot \mathrm{d}\boldsymbol{\xi}^t = \int_0^1 \begin{bmatrix} \partial_y \Psi(\boldsymbol{\gamma}(t)) \\ -\partial_x \Psi(\boldsymbol{\gamma}(t)) \end{bmatrix} \cdot \begin{bmatrix} \gamma_y'(t)) \\ -\gamma_x'(t)) \end{bmatrix} \mathrm{d}t \\
&= \int_0^1 \begin{bmatrix} \partial_x \Psi(\boldsymbol{\gamma}(t)) \\ \partial_y \Psi(\boldsymbol{\gamma}(t)) \end{bmatrix} \cdot \begin{bmatrix} \gamma_x'(t)) \\ \gamma_y'(t)) \end{bmatrix} \mathrm{d}t = \int_{\boldsymbol{\gamma}} \nabla \Psi(\boldsymbol{\xi}) \cdot \mathrm{d}\boldsymbol{\xi} \\
&= \Psi(\boldsymbol{\gamma}(1)) - \Psi(\boldsymbol{\gamma}(0)) = 0
\end{aligned}
\tag{5.4}
$$

This means that between any two points on the domain boundary, the total fluid flow of a solution in $\nabla \times \mathcal{S}_{p+1,0}^0$ must always be exactly zero. This restriction however is not part of our original function space V_0 defined in subsection 2.2. □

Proposition 3 (Consistency). *The proposed numerical scheme for piecewise constant basis functions is consistent only for right hand sides $\boldsymbol{f} \equiv \boldsymbol{0}$.*

Proof. Let $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ be the exact solution to our PDE and let V_h be our discretized function space we seek our solution in. Then we have for all $\boldsymbol{v}_h \in V_h$ that

$$
\begin{aligned}
\kappa(\boldsymbol{u}, \boldsymbol{v}_h) &= \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!] [\![\boldsymbol{v}_h \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \\
&= \sum_{e \in \mathcal{E}^o(\mathcal{M})} \int_e \frac{\sigma}{|e|} \underbrace{[\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!]}_{=0} [\![\boldsymbol{v}_h \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x + \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e]\!] [\![\boldsymbol{v}_h \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \\
&= \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e]\!] [\![\boldsymbol{v}_h \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \\
&= \sum_{e \in \mathcal{E}^\partial(\mathcal{M})} \int_e \frac{\sigma}{|e|} (\boldsymbol{g}_D \cdot \boldsymbol{\tau}_e) [\![\boldsymbol{v}_h \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x \\
&= k(\boldsymbol{v}_h)
\end{aligned}
\tag{5.5}
$$

which is equal to the right hand side $k(\boldsymbol{v}_h) + l(\boldsymbol{v}_h)$ of the variational formulation for all $\boldsymbol{v}_h \in V_h$ if and only if $\boldsymbol{f} \equiv 0$. □

5.2 Manufactured Solution

For the manufactured solution for the Stokes problem, we define the flow velocity $\boldsymbol{u}$ on the unit disk as

$$
\begin{aligned}
\boldsymbol{u}_r(r, \theta) &= 0 \\
\boldsymbol{u}_\theta(r, \theta) &= 1 - \cos(2n\pi r)
\end{aligned}
\tag{5.6}
$$

Notice that the flow velocity does not change with θ. This simplifies the Navier-Stokes equations in polar coordinates to
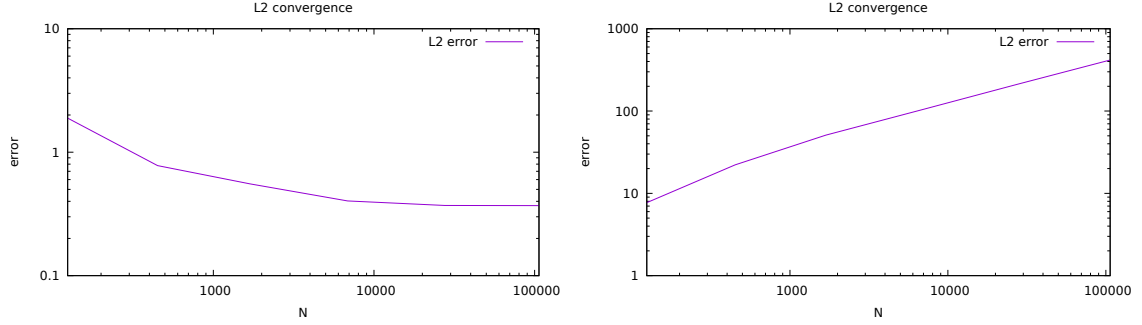
$$
\rho \frac{\boldsymbol{u}_\theta}{r} = \frac{\partial p}{\partial r}
\tag{5.7a}
$$

$$
0 = \mu \left(\frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial}{\partial r} (r\boldsymbol{u}_\theta) \right) \right) + \rho \boldsymbol{f}_\theta
\tag{5.7b}
$$

15

By plugging in $\boldsymbol{u}_\theta$ and solving for the volumetric forces $\boldsymbol{f}_\theta$ assuming that the kinematic viscosity $\nu = \frac{\mu}{\rho} = 1$ we obtain that

$$\boldsymbol{f}_\theta(r) = \frac{1 - (1 + 4\pi^2 n^2 r^2)\cos(2\pi nr) - 2\pi nr \sin(2\pi nr)}{r^2} \tag{5.8}$$
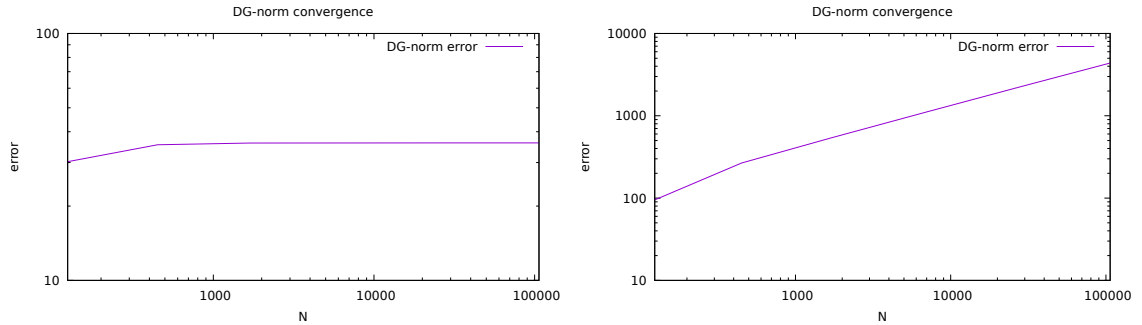
When measuring the error in the L^2-norm plotted in Figure 5, convergence fails as predicted in Proposition 3.

(a) The convergence of of the manufactured solution in L^2-norm with the original penalty term

(b) The convergence of of the manufactured solution in L^2-norm with the modified penalty term

Figure 5: Plots of the L^2-norm for the manufactured solution. We can see that the convergence fails for both the original and the modified penalty term. This is a consequence of the inconsistency of our method for nonzero volumetric forces.

Similarly, we do not observe any convergence in the DG-norm plotted in Figure 6. This is however expected due to Proposition 1.

(a) The convergence of of the manufactured solution in the DG-norm with the original penalty term

(b) The convergence of of the manufactured solution in the DG-norm with the modified penalty term

Figure 6: Plots of the error in the DG-norm for the manufactured solution. We do not observe any convergence as predicted in Proposition 1.

When visualizing the analytical and numerical solution in Figure 7, we see that the only difference between the two solutions is a constant factor. This can be explained by the fact that the only remaining terms of the variational formulation when using piecewise constant basis functions are given by

$$\boldsymbol{u} \in V_0 : \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{\sigma}{|e|} [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!][\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x = \rho \sum_{K \in \mathcal{M}} \int_K \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}x \quad \forall \boldsymbol{v} \in V_0 \tag{5.9}$$

The stabilization constant does not depend on any variable and can thus be taken out of the integral
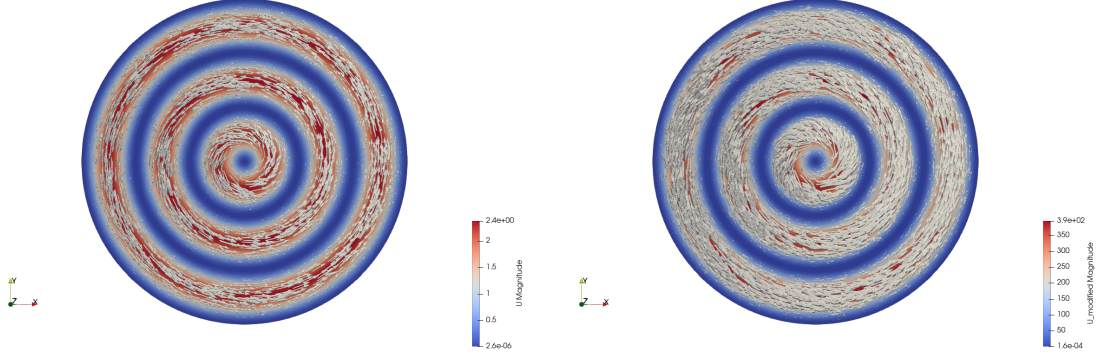
(a) The numerical solution to the manufactured problem with the original penalty term

(b) The numerical solution to the manufactured problem with the modified penalty term
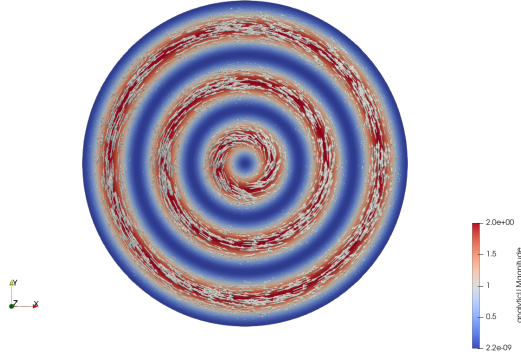
(c) The analytical solution to the manufactured problem

Figure 7: The flow velocities of the unscaled solutions to the manufactured solution problem. We can see that the only difference between them is a multiplicative scaling factor. This is due to the fact that all viscous terms vanish because we are using zeroth order basis functions. A theoretical explanation can be found in Equation 5.10.

and the sum and put on the right hand side:

$$\boldsymbol{u} \in V_0: \quad \sum_{e \in \mathcal{E}(\mathcal{M})} \int_e \frac{1}{|e|} [\![\boldsymbol{u} \cdot \boldsymbol{\tau}_e]\!][\![\boldsymbol{v} \cdot \boldsymbol{\tau}_e]\!] \, \mathrm{d}x = \frac{\rho}{\sigma} \sum_{K \in \mathcal{M}} \int_K \boldsymbol{f} \cdot \boldsymbol{v} \, \mathrm{d}x \quad \forall \boldsymbol{v} \in V_0 \tag{5.10}$$

We can now see that varying σ has the same effect as varying ρ which explains why the difference between the analytical and the numerical solution is only a constant factor.

To estimate this factor, we approximate the analytical solution as a piecewise constant function on the mesh cells and minimize the L^2-norm of the difference between the analytical solution and the numerical solution scaled with α:

$$\alpha = \operatorname*{argmin}_{\alpha' \in \mathbb{R}} \sum_{K \in \mathcal{M}} |K| \left(||\boldsymbol{u}|| - \alpha' ||\boldsymbol{u}_h|| \right)^2 = \frac{\sum_{K \in \mathcal{M}} |K| ||\boldsymbol{u}|| ||\boldsymbol{u}_h||}{\sum_{K \in \mathcal{M}} |K| ||\boldsymbol{u}_h||^2} \tag{5.11}$$

Multiplying the numerical solution by this estimate and plotting the error results in the plots Figure 8 and Figure 9 where we can clearly observe an algebraic convergence with rate $\frac{1}{2}$ in the L^2-norm.
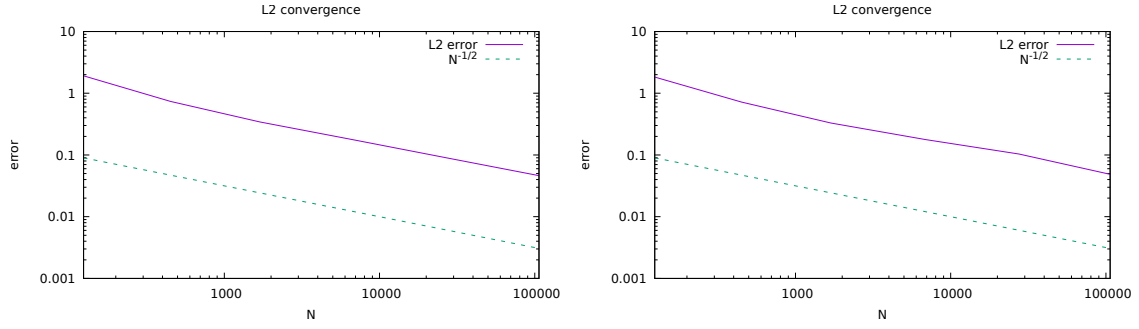
(a) The convergence of of the scaled manu-
factured solution in L^2-norm with the original
penalty term

(b) The convergence of of the scaled manufac-
tured solution in L^2-norm with the modified
penalty term

Figure 8: Plots of the convergence in the L^2-norm for the scaled manufactured solution. Both the original and the modified penalty term seem to converge algebraically with rate $\frac{1}{2}$.
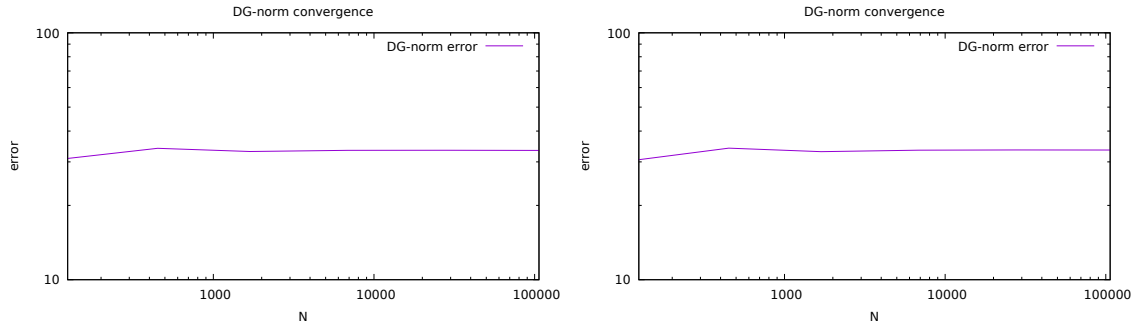
(a) The convergence of the scaled manufac-
tured solution in the DG-norm with the original
penalty term

(b) The convergence of the scaled manufac-
tured solution in the DG-norm with the modified
penalty term

Figure 9: Plots of the error in the DG-norm for the scaled manufactured solution. As shown in Proposition 1, we do not observe any convergence.

18

5.3 Lid Driven Cavity

The lid driven cavity experiment consists of a box where the top lid is driven with a constant velocity v. In our case we take $\Omega = [0, 1]^2$ and $v = 1$.

The PDE is solved on a refinement hierarchy of tensor product meshes and to measure the convergence rate, we substitute the analytical solution for the solution on the finest grid. A few of these meshes are drawn in Figure 10.
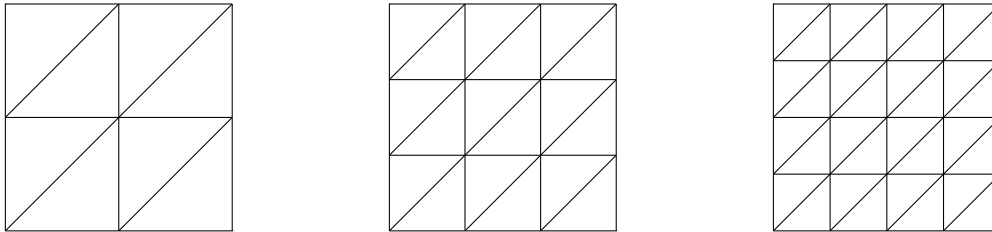
Figure 10: Examples of meshes the Lid Driven Cavity was solved on

Solving the PDE on the finest mesh with 66049 degrees of freedom resulted in the solution shown in Figure 11a. One can nicely see the Moffat Vortices in the bottom right corner of the cavity [2]. Theoretically, there is a sequence of infinitely many vortices becoming exponentially smaller. The grid resolution in this experiment however was only sufficient to see the first two. Increasing the grid resolution even further will lead to rounding errors when exporting and importing the mesh in ASCII format due to only a few floating point digits being saved. Locally refining the mesh is also not an option, as we see in Figure 11b and Figure 11c that the solution obtained is distorted regardless of whether we use the original or modified penalty term. This mesh dependence is further demonstrated and investigated in the following sections.
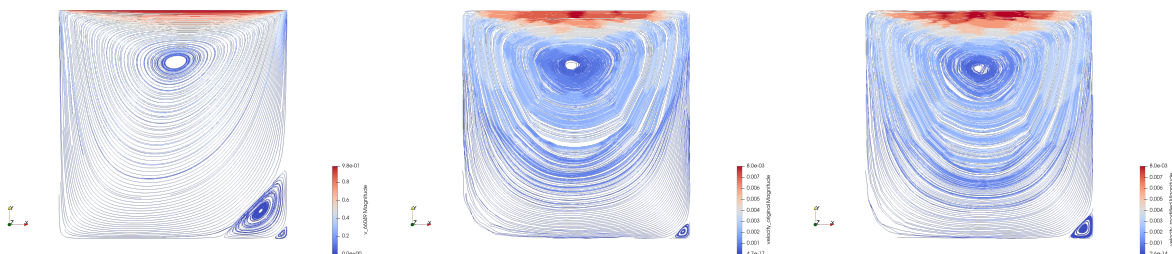
(a) The flow lines for a lid driven cavity experiment with driving velocity $v = 1$ on a tensor product mesh and the original penalty term

(b) The flow lines for a lid driven cavity experiment on a mesh with local refinement at the lower right corner and the original penalty term

(c) The flow lines for a lid driven cavity experiment on a mesh with local refinement at the lower right corner and the modified penalty term

Figure 11: The flow lines for the lid driven cavity experiment with Re = 1 on a regular mesh and on a locally refined mesh. According to the other experiments, the solution on the regular mesh should be correct. Thus we can clearly see that neither the original nor the modified penalty term converge to the correct solution.

An interesting aspect of the Lid Driven Cavity flow is that the Dirichlet boundary conditions at the top lid are not continuous but have a jump from 0 to 1 at the top edges of the cavity. This is not a problem though, as the boundary conditions are imposed weakly over the penalty term. The effect thereof is that the boundary conditions do not have to be met exactly as can be seen in Figure 12 where the flow velocity along the driven lid is shown.
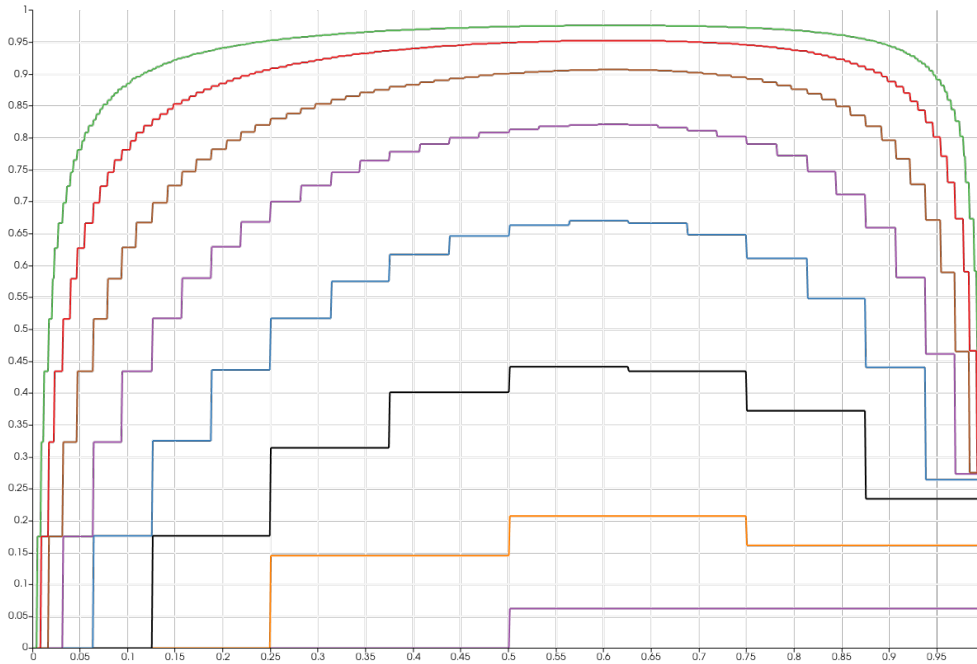
Figure 12: The flow velocity at the lid for different mesh resolutions for the original penalty term. We can clearly see the convergence towards the exact Diriclet boundary conditions with an increasing mesh resolution. This is because we impose them weakly in the tangential direction.

In Figure 13, the convergence of the numerical experiment is plotted in the L^2-norm. We can observe an algebraic convergence rate of approximately $\frac{1}{2}$.
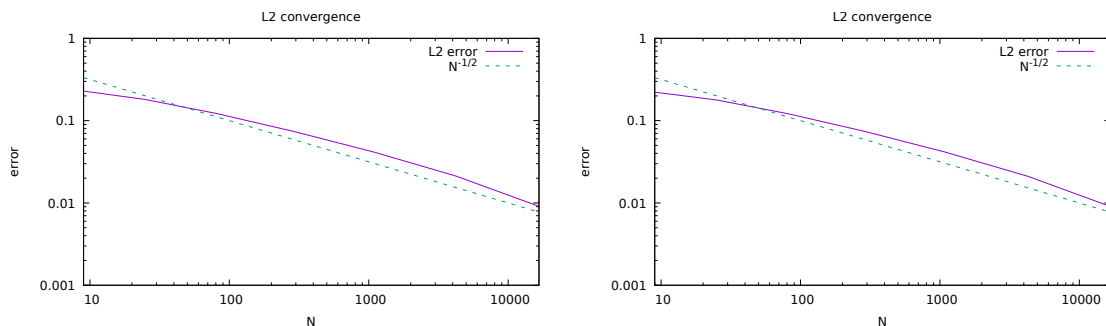
(a) The Convergence in the L^2 norm for the original penalty term

(b) The Convergence in the L^2 norm for the modified penalty term

Figure 13: Plots of the L^2-norm of the lid driven cavity experiment on a regular mesh. The algorithm seems to converge algebraically with rate $\frac{1}{2}$ for both the original and the modified penalty term.

We would expect the convergence rate to be diminished by discontinuous boundary conditions [8, Section 2.10.6]. Hoping for a better convergence rate, we take the weighted L^2 norm of the solution. This weighting is chosen such that the driven lid where the boundary conditions are discontinuous contributes very little to the overall norm

$$W_h(x,y) = \begin{cases} \frac{1}{2}\left(1 - \cos\left(\frac{\pi}{h}(1-y)\right)\right) & \text{for } y \geq 1 - h \\ 1 & \text{else} \end{cases} \tag{5.12}$$

The result of this weighted norm is plotted in Figure 14, but surprisingly we do not see an improvement in the convergence rate.

Finally, in Figure 15 the convergence in the DG-norm is plotted. As expected and proven in Proposition 1 we do not observe convergence.
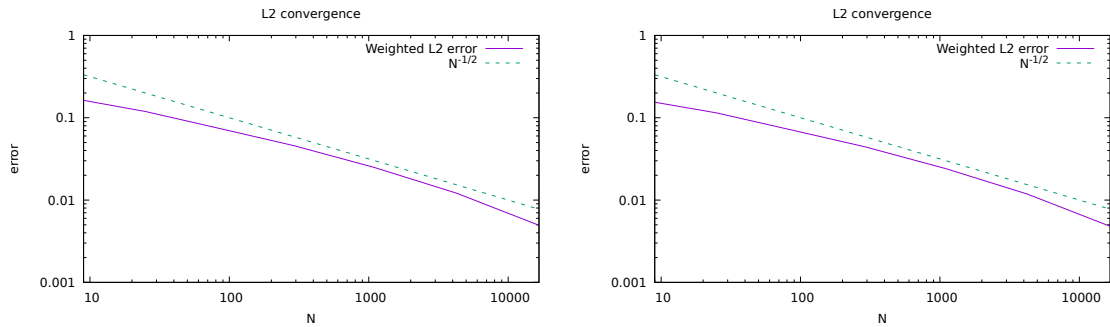
(a) The Convergence in the L^2 norm for the original penalty term weighted with W_h to not include the discontinuous boundary conditions

(b) The Convergence in the L^2 norm for the modified penalty term weighted with W_h to not include the discontinuous boundary conditions

Figure 14: Plots of the convergence in the weighted L^2-norm. Apart from a different multiplicative constant, the error behaves exactly the same as in the unweighted case. This leads us to the conclusion that discontinuous Dirichlet boundary condition do not influence the convergence rate.
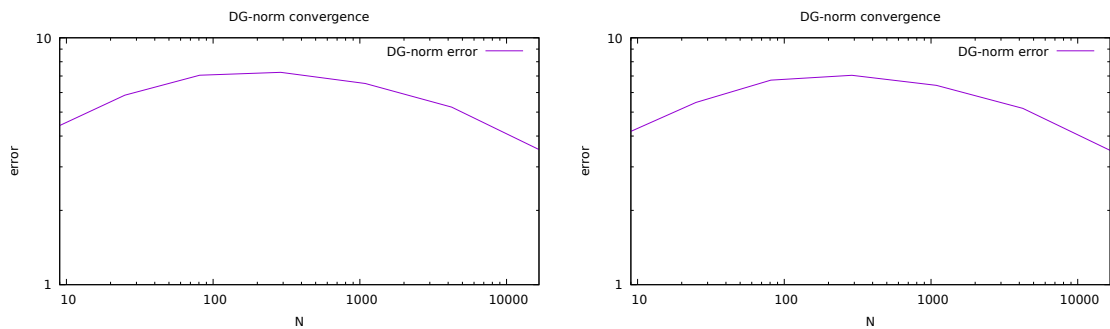
(a) The Convergence in the DG-norm for the original penalty term

(b) The Convergence in the DG-norm for the modified penalty term

Figure 15: Plots of the convergence in the DG norm. The decrease in the decrease in the error for the high mesh resolutions occurs because we use the solution on the finest mesh as the analytical solution.

21

5.4 Poiseuille

The Poiseuille velocity profile is the flow velocity attained by a fluid flowing between two infinitely extended plates [7, Chapter 6.1.2]. It is one of the few analytic solutions known for the full Navier-Stokes equations.
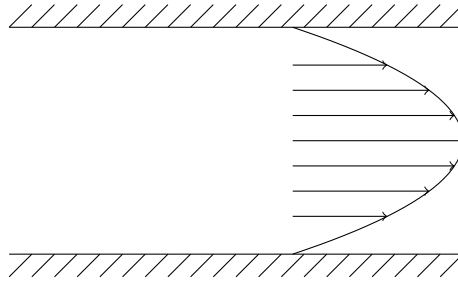
Figure 16: The Poiseuille velocity profile between two infinite plates

In order to analytically compute this velocity profile, we introduce the following assumptions:

$$\frac{\partial \boldsymbol{u}}{\partial t} = 0, \quad \frac{\partial \boldsymbol{u}}{\partial x} = 0, \quad \frac{\partial p}{\partial x} = \text{const}, \quad u_y = 0, \quad \boldsymbol{f} = 0 \tag{5.13}$$

Under these assumptions, the Navier-Stokes equations simplify considerably and we are left with

$$\mu \frac{\partial^2 u_x}{\partial y^2} = \frac{\partial p}{\partial x} \tag{5.14a}$$

$$0 = \frac{\partial p}{\partial y} \tag{5.14b}$$

Integrating the first equation twice gives

$$u_x(y) = \frac{1}{2\mu} \frac{\partial p}{\partial x} y^2 + C_1 y + C_2 \tag{5.15}$$

where we can choose C_1 and C_2 such that the velocity profile fulfills the no-slip condition at the plates.

For the numerical experiment, we choose the distance between the walls to be $\frac{1}{2}$ and the wall positions are respectively at $y = -\frac{1}{4}$ and $y = \frac{1}{4}$. Additionally, we scale the pressure gradient $\frac{\partial p}{\partial x}$ such that the total flow rate is equal to one and assume a dynamic viscosity 1.

The mesh was again a tensor product mesh like in Figure 10 but rectangular rather than square. The boundary conditions on the left and right side of the mesh were set to the previously computed Poiseuille velocity profile and the top and bottom wall were set to zero.
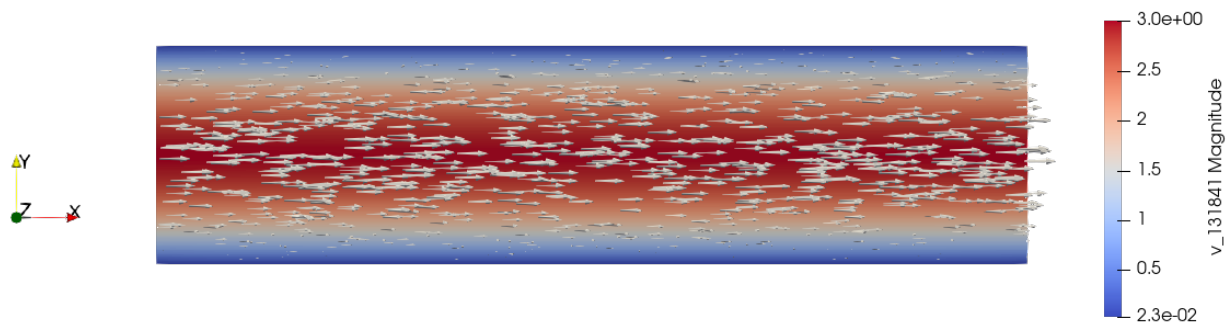
Figure 17: The solution of the PDE on the finest mesh

The simulated velocity profile shown in Figure 17 does not seem to change in the middle of the domain indicating that the numerical scheme derived is indeed correct. At least when no volumetric forces are present.
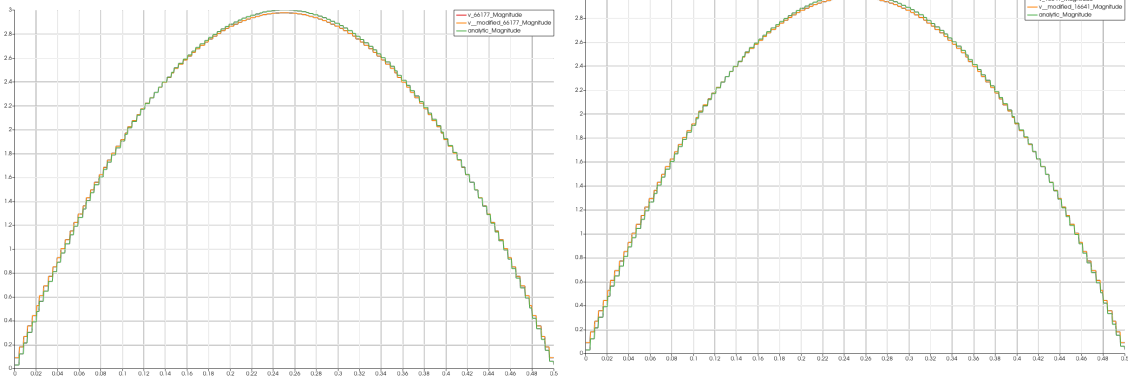
(a) The velocity profile in the middle of the domain compared to the analytical solution when solved on a regular mesh

(b) The velocity profile in the middle of the domain compared to the analytical solution when solved on a mesh with a 10 times smaller resolution in the x-direction

Figure 18: A comparison of the velocity profiles in the middle of the domain. For both meshes, the original and modified penalty terms seem to agree with the analytic solution. This suggests that not the mesh regularity measure directly but only its change over the computational domain is responsible for the distortion of solutions.

A cross-section of the numerical solution is compared to the analytical solution in Figure 18.. The clear agreement between the numerical and the analytic solution suggests its correctness. We can also observe that the shape of the triangles does not change the solution for either penalty term.

We can see in Figure 19 and Figure 20 that the convergence is again algebraic with rate $\frac{1}{2}$. The DG-norm plotted in Figure 21 and Figure 22 does again not converge as expected.
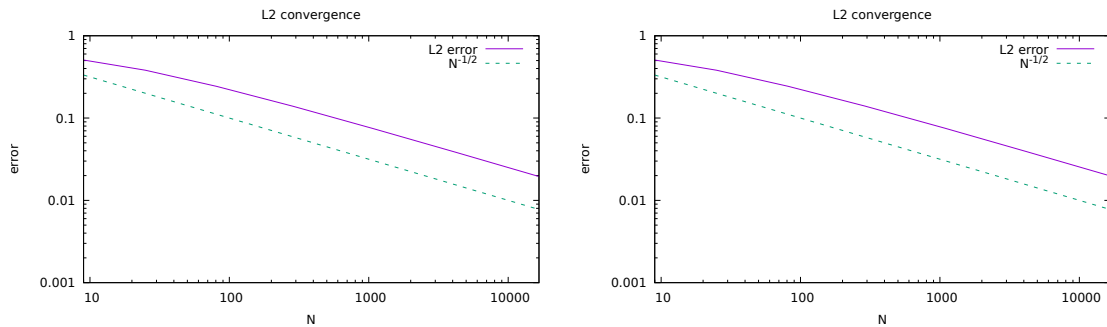
(a) The convergence of a pipe flow towards the Poiseuille velocity profile in L^2-norm for the original penalty term

(b) The convergence of a pipe flow towards the Poiseuille velocity profile in L^2-norm for the modified penalty term

Figure 19: Plots of the convergence of the Poiseuille velocity profile in L^2-norm using a mesh with a 10 times smaller resolution in the x-direction. Again, algebraic convergence with rate $\frac{1}{2}$ is observed indicating that a global change in the mesh regularity measure is not responsible for the distortion of solutions.
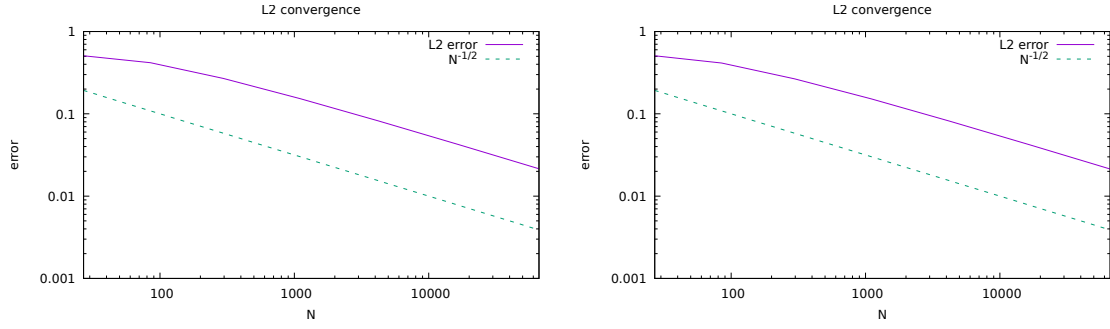
(a) The convergence of a pipe flow towards the Poiseuille velocity profile in L^2-norm for the original penalty term

(b) The convergence of a pipe flow towards the Poiseuille velocity profile in L^2-norm for the modified penalty term

Figure 20: Plots of the convergence of the Poiseuille velocity profile in L^2-norm using a regular mesh. The convergence is algebraic with rate $\frac{1}{2}$ for both the original and the modified penalty term.
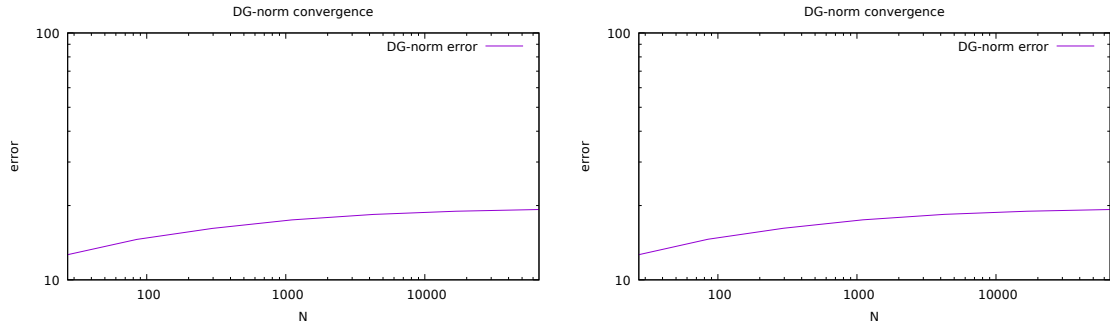
(a) The convergence of a pipe flow towards the Poiseuille velocity profile in the modified DG-norm for the original penalty term

(b) The convergence of a pipe flow towards the Poiseuille velocity profile in the modified DG-norm for the modified penalty term

Figure 21: Plots of the convergence towards the Poiseuille velocity profile in the DG-norm using a regular mesh. No convergence is observed as proven in Proposition 1.
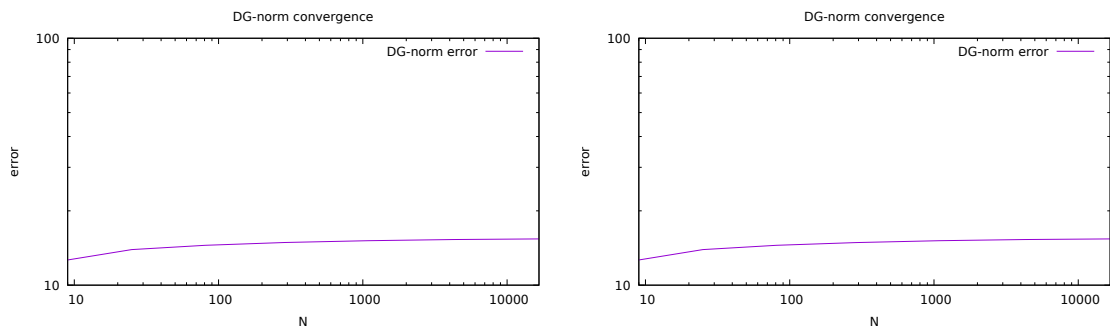
(a) The convergence of a pipe flow towards the Poiseuille velocity profile in the modified DG-norm for the original penalty term

(b) The convergence of a pipe flow towards the Poiseuille velocity profile in the modified DG-norm for the modified penalty term

Figure 22: Plots of the convergence towards the Poiseuille velocity profile in the DG-norm using a mesh with a 10 times smaller resolution in the x-direction. No convergence is observed as proven in Proposition 1.

24

5.5   Step

This experiment again simulates the flow between two infinite plates but this time the lower one has a step in it. The corresponding coarsest mesh is shown in Figure 23. The PDE was again solved on a mesh hierarchy generated by regular refinement of this base mesh.
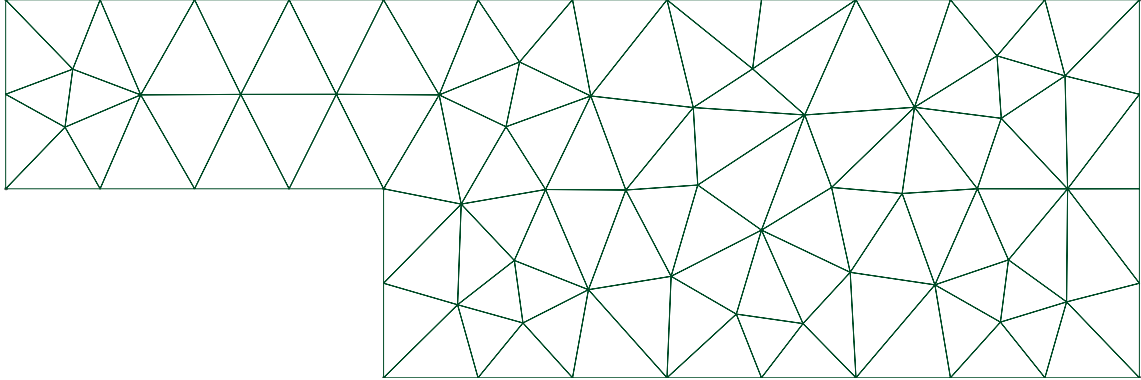
Figure 23: The coarsest triangulation of the domain generated by GMSH. For the experiments with higher resolution meshes, this mesh was taken and refined by the LehrFEM++ internal routines for regular refinement.

The boundary conditions were chosen such that the flow velocity is zero at the top and bottom plates and for the in and out flow a Poiseuille velocity profile was used with a total flow rate of 1. This is a suitable choice because sufficiently far away from the disturbance, it will almost be unnoticeable and thus the velocity profile can be well approximated by the analytical solution.

The result of solving the PDE on the finest mesh with 53761 DOFs is shown in Figure 24. An interesting observation is again the occurrence of a vortex in the bottom left corner of the step.
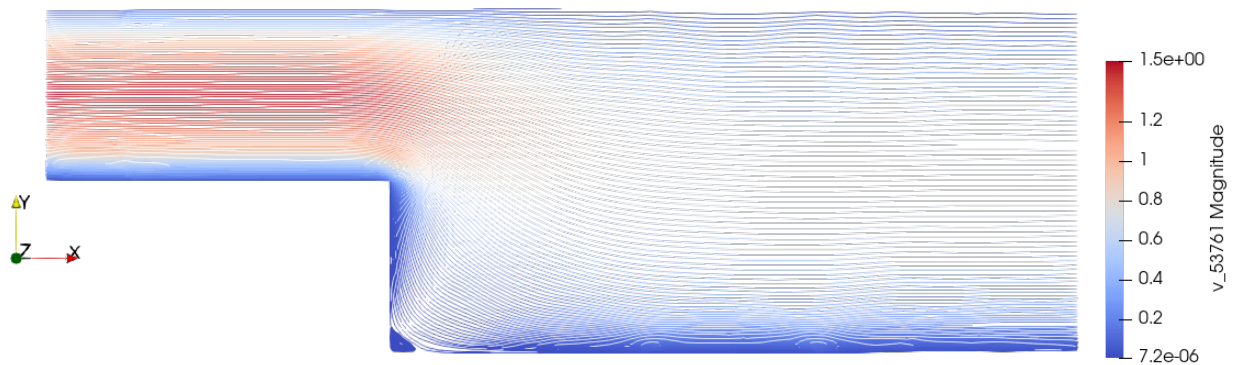
Figure 24: The solution of the PDE on the finest mesh

In spite of the domain not being convex and having an inward pointing corner, we still observe a convergence rate of $\frac{1}{2}$ in the L^2 norm as plotted in Figure 25. As expected, we again observe no convergence in the DG norm plotted in Figure 26.
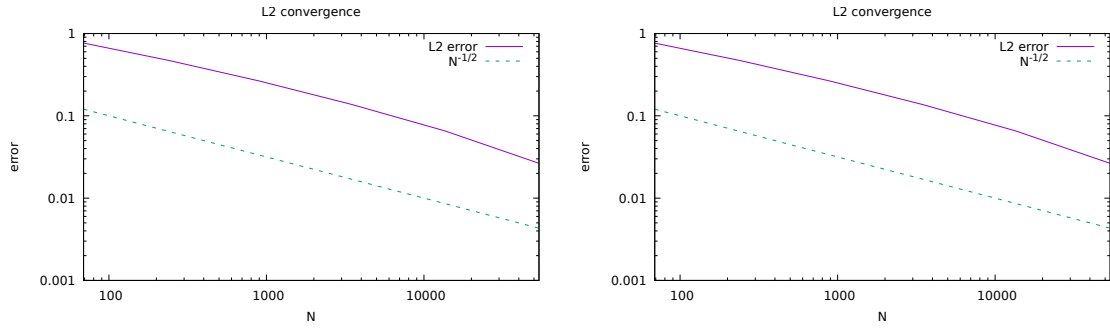
(a) The L^2 error for a pipe flow over a section of sudden change in diameter using the original penalty term

(b) The L^2 error for a pipe flow over a section of sudden change in diameter using the modified penalty term

Figure 25: The plots of the convergence in the L^2-norm for the flow over a sudden change in diameter. In spite of the domain having an inward pointing edge and thus not being convex, we can observe an algebraic convergence with rate $\frac{1}{2}$.
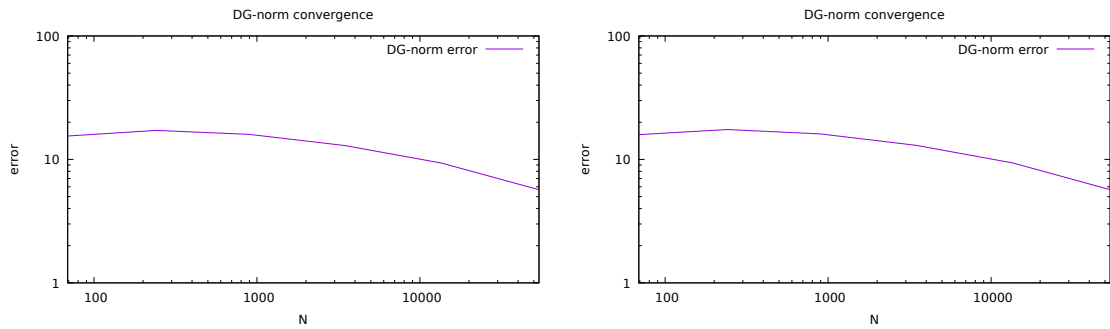
(a) The DG norm error for a pipe flow over a section of sudden change in diameter using the original penalty term

(b) The DG norm error for a pipe flow over a section of sudden change in diameter using the modified penalty term

Figure 26: The plots of the convergence in the DG-norm for the flow over a sudden change in diameter. We observe a decrease in the error for higher mesh resolutions. This has the same reason as explained in the lid driven cavity experiment in subsection 5.3. Because wo do not know the analytic solution to this problem, we rely on the solution on the finest mesh to be close enough to it. However, this makes the solutions on the higher resolution meshes seem to converge towards it.

5.6 Nested Cylinders

The nested cylinders is a demonstration of the topological constraints on the domain mentioned in subsection 4.1. We consider the flow of fluid between two rotating concentric cylinders as shown in Figure 27.
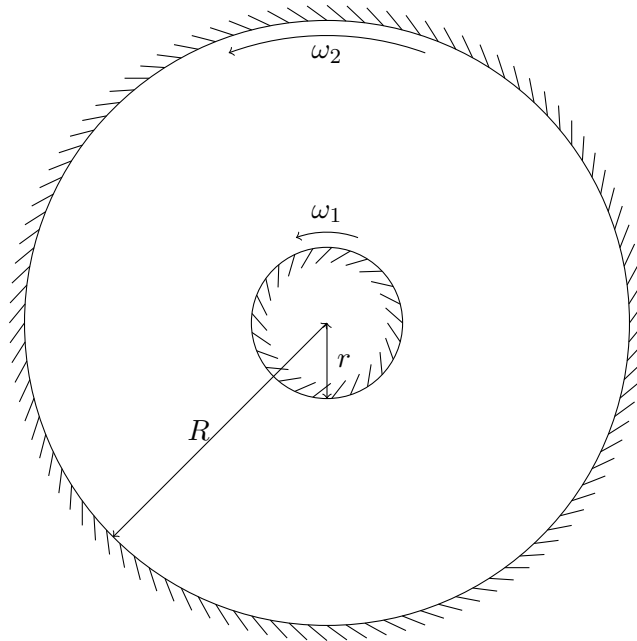
Figure 27: The experimental setup with fluid between the two concentric cylinders. The cylinders rotate with a constant angular velocity of ω_1 and ω_2 respectively.

This problem is actually analytically solvable by the plausible assumption that the radial velocity $u_r = 0$ and that we have no change in the angular direction $\frac{\partial \boldsymbol{u}}{\partial \theta} = 0$. In that case, the Navier-Stokes equations in polar coordinates simplify to

$$\rho \frac{u_\theta^2}{r} = \frac{\partial p}{\partial r} \tag{5.16a}$$

$$0 = \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial}{\partial r} (r u_\theta) \right) \tag{5.16b}$$

The second equation can easily be solved for u_θ and we get that [7, Eq 6.42]

$$u_\theta(r) = \frac{C_1}{2} r + \frac{C_2}{r} \tag{5.17}$$

For the numerical experiment, we choose the inner radius $r = \frac{1}{4}$ and the outer radius $R = 1$. Furthermore, the inner cylinder is fixed and the outer cylinder has an angular velocity of $\omega_2 = 1$. This gives the analytic solution

$$u_\theta(r) = \frac{16}{15} r - \frac{1}{15r} \tag{5.18}$$

The numerical solution however does not converge towards the predicted analytical solution as can be clearly seen in Figure 28 and Figure 29.

The reason for this is our choice of basis for the function space we seek our solutions in. By choosing the function space to be $\nabla \times \mathcal{S}_{p+1,0}^0$, we implicitly assume that all our functions are represented by some vector potential in $\mathcal{S}_{p+1,0}^0$. In Proposition 2 we proved that in this case, solutions to the problem do not necessarily exist on non simply connected domains.

From Figure 30 and Figure 31 we can clearly see that the numerical solution does indeed not converge to the analytical solution but the error term itself converges in the DG- as well as in the L^2 norm indicating that we have convergence towards the function shown in Figure 28a.

(a) Numerical solution
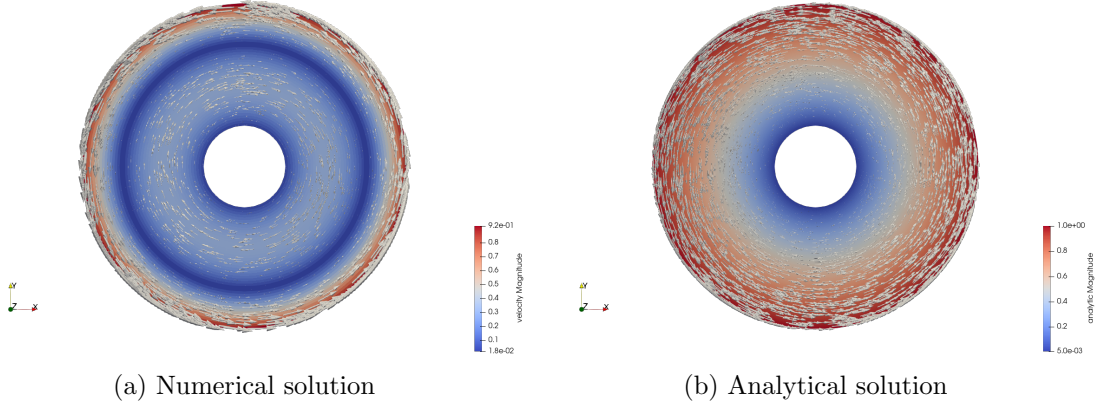
(b) Analytical solution

Figure 28: A comparison between the numerical and the analytical solution of the PDE for the described experimental setup. The numerical solution was obtained using the original penalty term. We see that for the numerical solution, the fluid flows in both directions around the inner cylinder. This is a consequence of Proposition 2
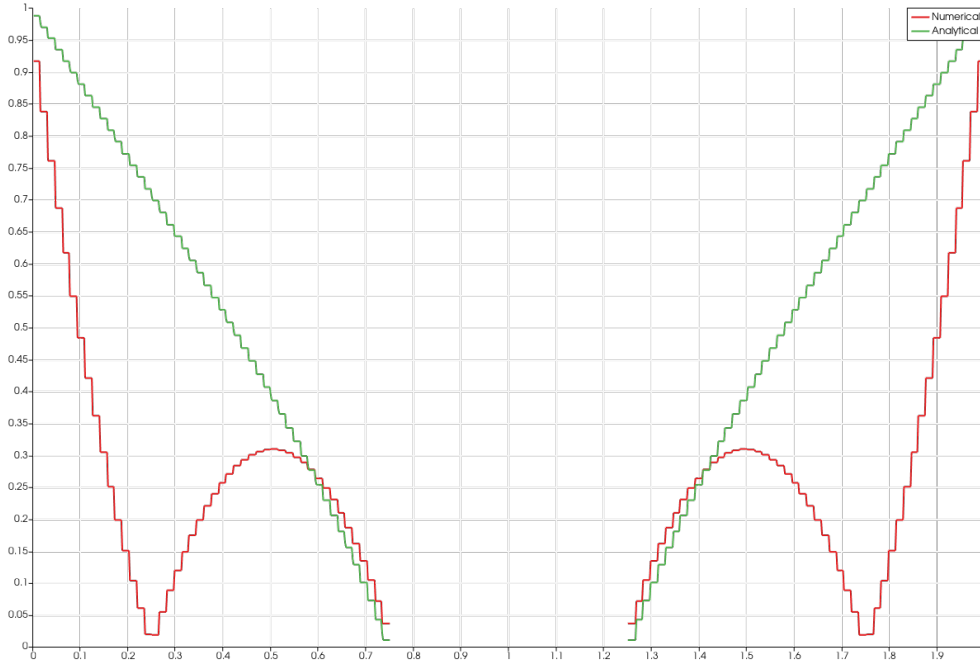

Figure 29: A cross section through the magnitude of the velocity profile of the analytical and numerical solutions on the finest mesh. The numerical solution was obtained using the original penalty term. Together with the flow direction in Figure 28a, we see that the total fluid flow over a cross section between the two cylinders is indeed zero as was shown in Proposition 2.

An approach to loosen the topological constraint on our domain is to use a vector potential with nonzero boundary conditions. Note however, that the value of the potential along a boundary must stay constant in order to comply with the constraint $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ on $\partial \Omega$. We thus replace the basis functions on each disconnected component $\partial \Omega_k$ of the domain boundary $\partial \Omega$ by a single basis function b_k^{∂} defined as

$$b_k^{\partial} = \sum_{i \in \partial \Omega_k} b_i \tag{5.19}$$

Evaluating some bilinear form $a(\boldsymbol{u}, \boldsymbol{v})$ or some linear form $l(\boldsymbol{v})$ involving this boundary basis function
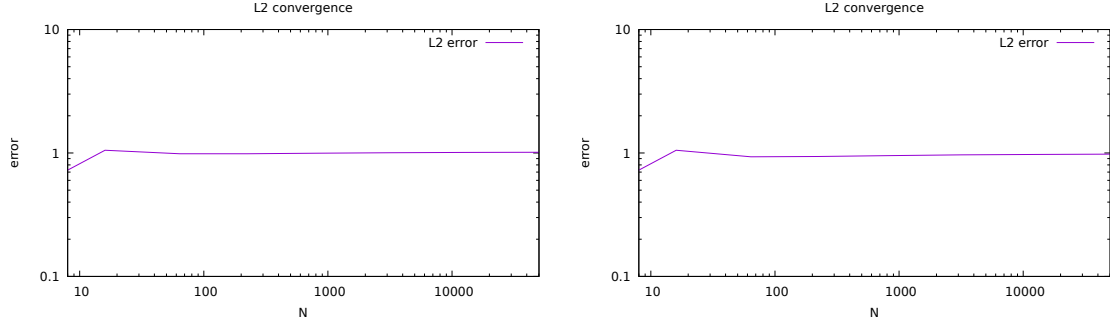
28

(a) The convergence in L^2 norm for the flow between two concentric rotating cylinders with zero potential at the boundary and the original penalty term

(b) The convergence in L^2 norm for the flow between two concentric rotating cylinders with zero potential at the boundary and the modified penalty term

Figure 30: The convergence in the L^2-norm for the nested cylinders experiment with zero potential at the boundary. As expected from Proposition 2, no convergence can be observed.
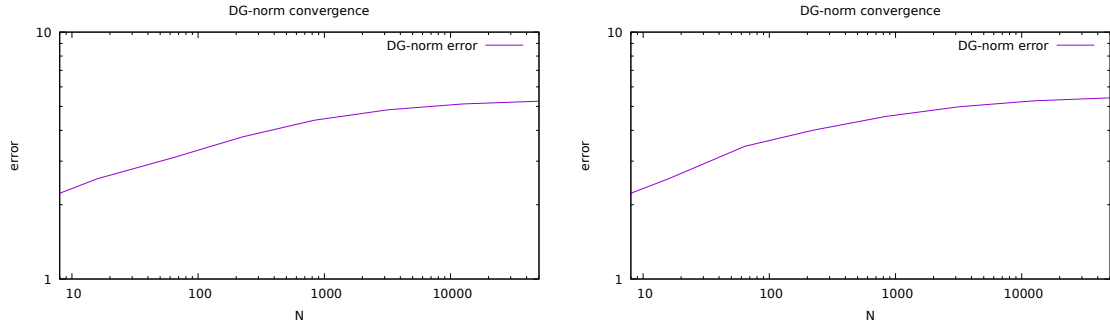
(a) The convergence in the DG norm for the flow between two concentric rotating cylinders with zero potential at the boundary and the original penalty term

(b) The convergence in the DG norm for the flow between two concentric rotating cylinders with zero potential at the boundary and the modified penalty term

Figure 31: The convergence in the DG-norm for the nested cylinders experiment with zero potential at the boundary. Again, no convergence can be observed as is proven in Proposition 1.

we get

$$a(b_i^\partial, b_j) = a\left(\sum_{i \in \partial\Omega_i} b_i, b_j\right) = \sum_{i \in \partial\Omega_i} a(b_i, b_j) \tag{5.20a}$$

$$l(b_i^\partial) = l\left(\sum_{i \in \partial\Omega_i} b_i\right) = \sum_{i \in \partial\Omega_i} l(b_i) \tag{5.20b}$$

We are thus able to compute this new system matrix from the one with arbitrary boundary conditions. LehrFEM++ stores its matrices in the COO-format for easy interoperability with the Eigen library [11]. This has the benefit that when two triplets (i, j, a_{ij}) have the same coordinates i and j, their values a_{ij} are added. To convert the system matrix for arbitrary boundary conditions to the ones given above, it therefore suffices to transform the triplets in the following way:

$$(i, j, a_{ij}) \mapsto \begin{cases} (i, j, a_{ij}) & \text{for } i, j \notin \partial\Omega_k \\ (M_k, j, a_{ij}) & \text{for } i \in \partial\Omega_k \\ (i, M_k, a_{ij}) & \text{for } j \in \partial\Omega_k \\ (M_k, M_k, a_{ij}) & \text{for } i, j \in \partial\Omega_k \end{cases} \tag{5.21}$$

29

where $M_k \in \mathbb{N}$ is the index of the DOF for the boundary basis function b_k^∂. Note that by doing this, we introduce one degree of freedom too much, as the vector potential is only unique up to some additive constant. We fix this by setting the vector potential to zero on some arbitrary boundary $\partial\Omega_k$.

This method is then applied to the current numerical experiment by hardcoding the position of the boundaries for this specific case. To apply it to some general mesh, we would need to identify the disconnected components of the boundary first. Additionally to the renumbering of rows and columns given above, all now unused degrees of freedom have to be removed from the matrix as it would be singular otherwise.

After implementing constant boundary conditions for the vector potential, we see in Figure 32 and Figure 33 that the method with the original penalty term does still not converge to the analytical solution given in Figure 28b. This is confirmed by the convergence plots in Figure 34 and Figure 37. However, the modified penalty term seems to converge initially, as we can observe an algebraic convergence with rate $\frac{1}{2}$ in the L^2-norm. The plot however suggests that the convergence rate might significantly slow down for higher mesh resolutions. This would imply that the modified penalty term also does not discretize the Laplacian. This is supported by the convergence plots in Figure 40 where uniform meshes were used and no slowing down of the convergence can be observed.

(a) Numerical solution with a constant potential at the boundaries using the original penalty term

(b) Numerical solution with a constant potential at the boundaries using the modified penalty term
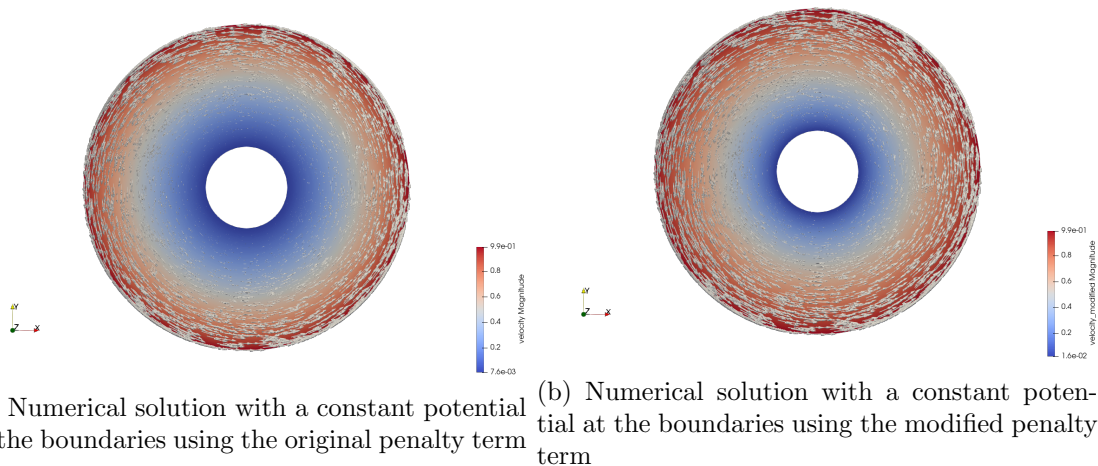
Figure 32: The flow velocity plotted for the nonzero boundary conditions. Although the total flow over a curve between the two cylinders is not zero anymore, we still observe slightly different solutions depending on whether we use the original or the modified penalty term. The solution using the modified penalty term however seems to agree better with he analytical solution in Figure 28b.

When solving the PDE on a mesh with a sudden jump in the resolution, it becomes evident that the modified penalty term must also be incorrect as can be seen from the flow velocities in Figure 36a and Figure 36b. In both cases, the jump in the resolution occurs at distance $r = 0.625$ from the origin. The first plot corresponds to a mesh with the inner part of the annulus more refined than the outer part. The second plot shows the solution when the outer part is more refined.

We can observe that the original penalty term is robust against sudden changes in the mesh resolution while the modified one is greatly influenced by it. This is not what we have expected as the previous experiments suggested that the distortion of the solutions using the original penalty term is due to a change in the mesh resolution.
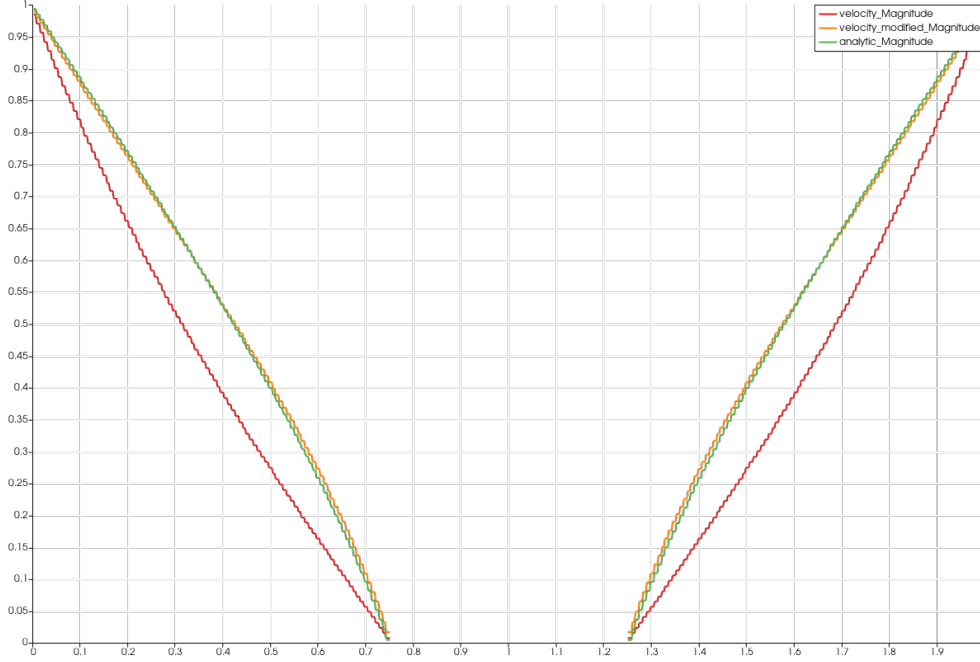
Figure 33: A cross section through the magnitude of the velocity profile of the analytical and numerical solutions on the finest mesh with a constant potential at the boundaries. Here we can clearly see that the modified penalty term (orange) agrees better with the analytical solution (green) than the original penalty term (red). This suggests that the change in the mesh regularity measure is responsible for the distortion of the solutions.
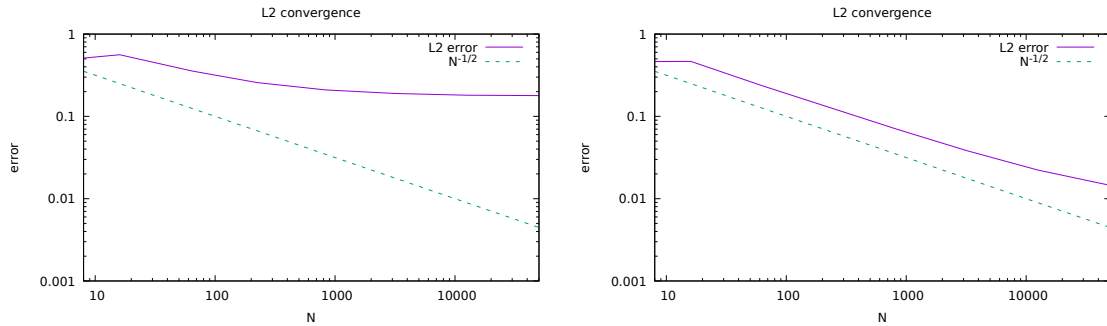
(a) The convergence in L^2 norm for the flow between two concentric rotating cylinders with constant potential at the boundary and the original penalty term

(b) The convergence in L^2 norm for the flow between two concentric rotating cylinders with constant potential at the boundary and the modified penalty term

Figure 34: The convergence in the L^2-norm for the nested cylinders experiment with nonzero potential at the boundary. We can clearly see that the original penalty term does not converge towards the analytical solution. The convergence of the modified penalty term also seems to slow down for higher mesh resolutions indicating that it also does not discretize the Laplacian on irregular meshes.
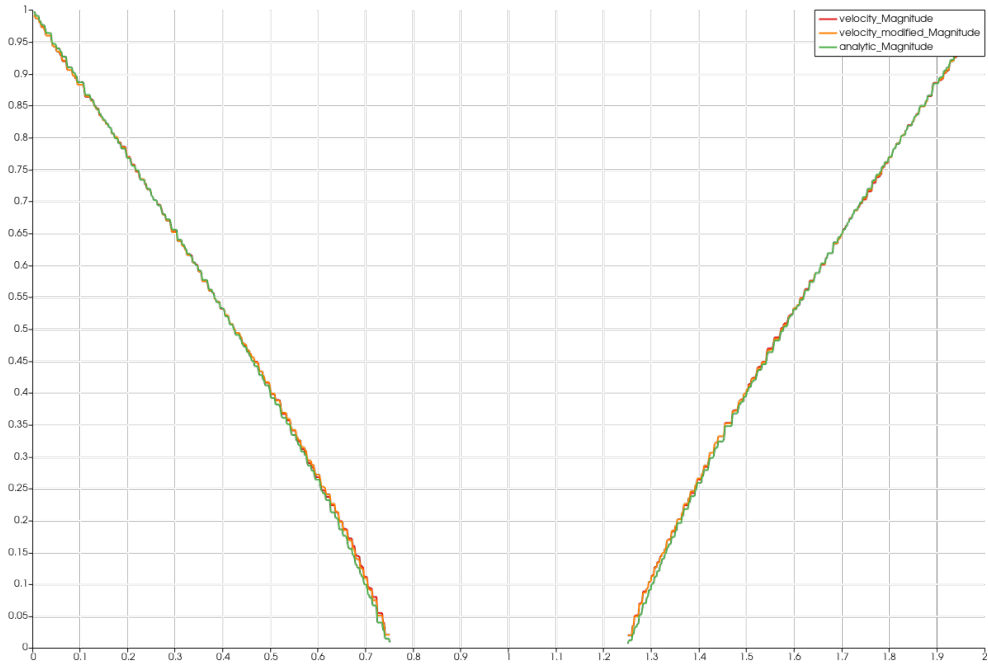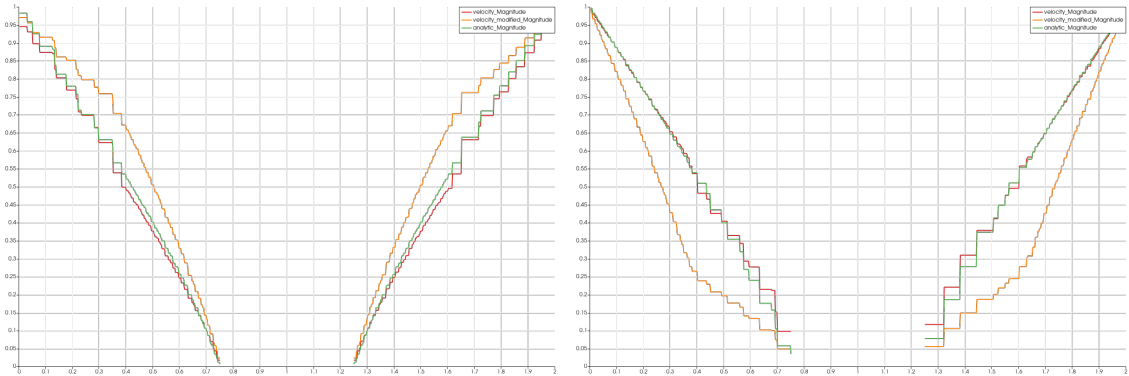
31

Figure 35: A cross section through the magnitude of the velocity profile of the analytical and numerical solutions on a uniform mesh with a constant potential at the boundaries. This plot again confirms that both the original and the modified penalty term give the correct solutions on uniform meshes.

(a) A cross section through the magnitude of the velocity profile of the analytical and numerical solutions on a irregular mesh with a higher resolution towards the inner cylinder and a constant potential at the boundaries

(b) A cross section through the magnitude of the velocity profile of the analytical and numerical solutions on a irregular mesh with a higher resolution towards the outer cylinder and a constant potential at the boundaries

Figure 36: The cross section through the magnitude of the velocity profile on meshes with a sudden jump in resolution. The jump occurs at $r = 0.625$. We can see that the original penalty is robust against this sudden change while the modified penalty term is distorted. This is not what we expected, as in the previous experiments it was established that the change in the mesh resolution is responsible for the distortion of the solutions using the original penalty term.
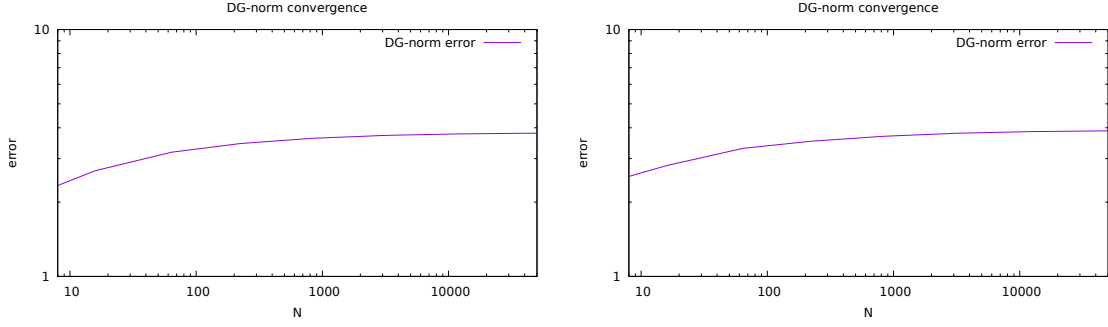
(a) The convergence in the DG norm for the flow between two concentric rotating cylinders with constant potential at the boundary and the original penalty term

(b) The convergence in the DG norm for the flow between two concentric rotating cylinders with constant potential at the boundary and the modified penalty term
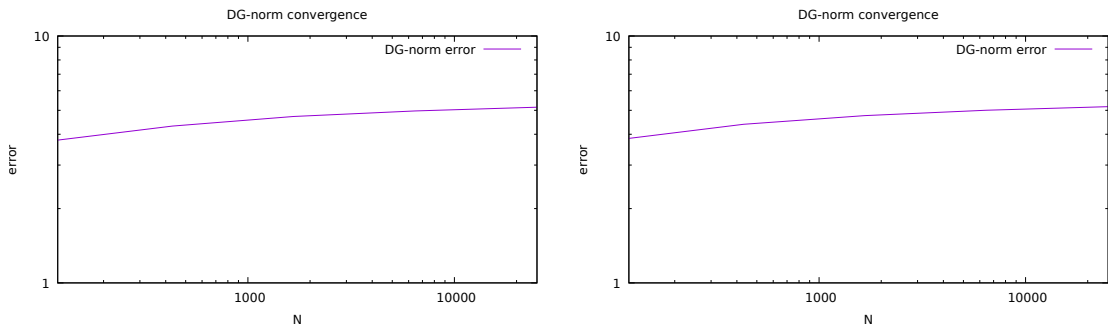
Figure 37: The convergence in the DG-norm for the nested cylinders experiment with nonzero potential at the boundary. No convergence can be observed as proven in Proposition 1.
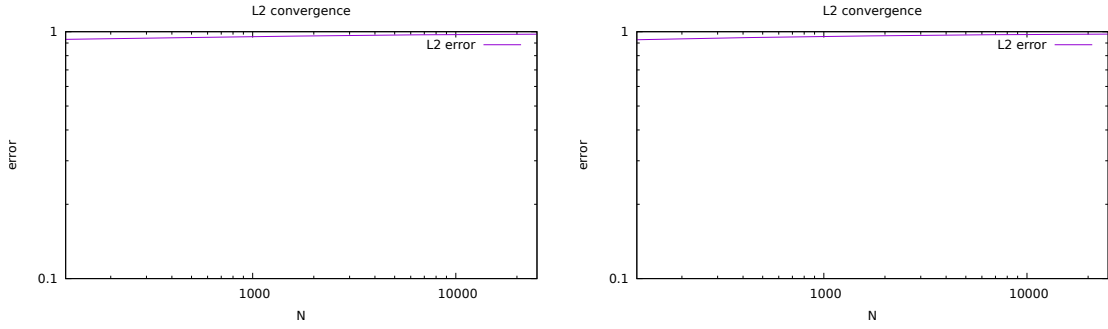
(a) The convergence in the L^2-norm with a uniform mesh, zero potential at the boundary and the original penalty term

(b) The convergence in the L^2-norm with a uniform mesh, zero potential at the boundary and the modified penalty term
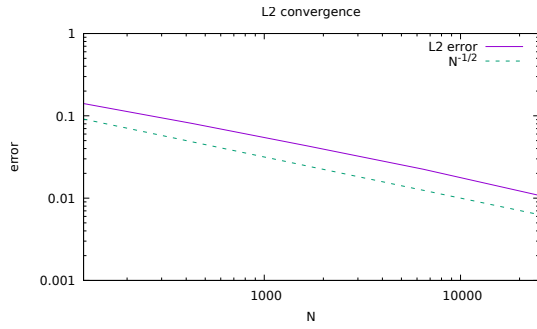
Figure 38: The convergence in the L^2-norm of the nested cylinders experiment on a uniform mesh and zero potential at the boundaries. Due to Proposition 2, no convergence can be observed.
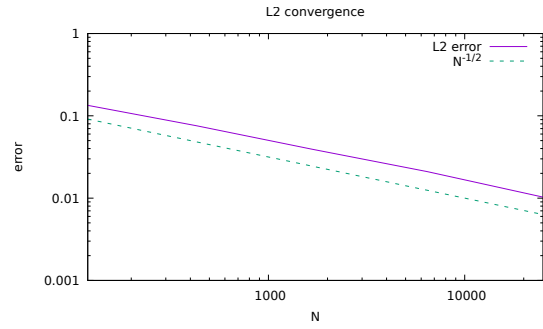

(a) The convergence in the DG-norm with a uniform mesh, zero potential at the boundary and the original penalty term

(b) The convergence in the DG-norm with a uniform mesh, zero potential at the boundary and the modified penalty term

Figure 39: The convergence in the DG-norm of the nested cylinders experiment on a uniform mesh and zero potential at the boundaries. As previously established in Proposition 1, no convergence in the DG-norm can be observed.
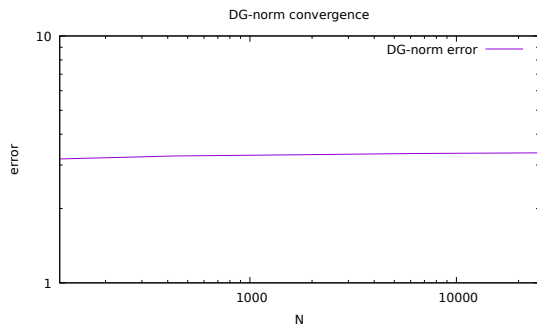
(a) The convergence in the L^2-norm with a uniform mesh, nonzero potential at the boundary and the original penalty term
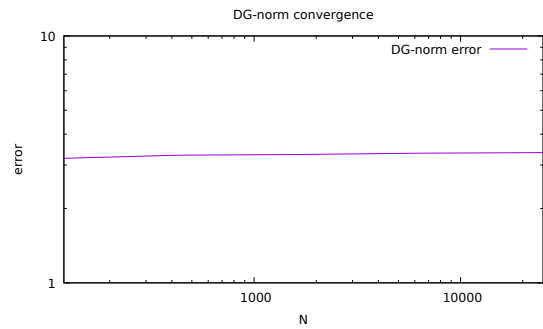
(b) The convergence in the L^2-norm with a uniform mesh, nonzero potential at the boundary and the modified penalty term

Figure 40: The convergence in the L^2-norm of the nested cylinders experiment on a uniform mesh and nonzero potential at the boundaries. For both the original and the modified penalty term, an algebraic convergence with rate $\frac{1}{2}$ can be observed.

(a) The convergence in the DG-norm with a uniform mesh, nonzero potential at the boundary and the original penalty term

(b) The convergence in the DG-norm with a uniform mesh, nonzero potential at the boundary and the modified penalty term

Figure 41: The convergence in the DG-norm of the nested cylinders experiment on a uniform mesh and nonzero potential at the boundaries. Again no convergence can be observed due to Proposition 1.

6 Conclusions

We have successfully derived the Interior Penalty Discontinuous Galerkin discretization of the Stokes Problem and implemented it for the finite element space $V_h = \nabla \times \mathcal{S}_{1,0}^0$ on triangular meshes. The normal component of the Dirichlet boundary conditions were enforced using the offset function technique while the tangential component was enforced weakly via the penalty term.

We showed that the zeroth order approximation suffers from some major problems including a dependence on the mesh. We were however not able to track down the cause of this mesh dependency. The experiment conducted in subsection 5.4 suggests that the solution to the discretized problem is invariant under a global change in the mesh regularity measure. This hints at the difference of the mesh regularity measure between two triangles being responsible for the distortion of solutions. In the nested cylinders experiment in subsection 5.6, we observed that a sudden change in the mesh resolution does not noticeably change the solution using the original penalty term but drastically distorts the solution using the modified penalty term. This challenges the conclusion made previously. A general trend observed over all experiments is that the implementations seem to agree with the analytical solution when solved on uniform meshes and no volumetric forces are present. In that case the convergence rate in L^2 is algebraic with rate $\frac{1}{2}$. This convergence rate is invariant even on non convex domains and with discontinuous Dirichlet boundary conditions.

Some restrictions of using the lowest order finite element space were stated in subsection 5.1 and later demonstrated and explained in subsection 5.2 and subsection 5.6. Other numerical experiments were investigated to test the correctness of the method. In the lid driven cavity experiment in subsection 5.3 we demonstrated that the convergence rate is invariant even when the Dirichlet boundary conditions are discontinuous.

A big improvement over the current implementation would be to use the second order Raviart-Thomas finite element space instead of $\nabla \times \mathcal{S}_{1,0}^0$. The former is not piecewise constant and thus does not have the problems observed in section 5. This comes at the cost of having a significantly more complicated implementation. Furthermore, because RT_p is not necessarily representable by a vector potential in $\nabla \times \mathcal{S}_{p+1,0}^0$, solutions also exist for non simply connected domains without the need for identifying different parts of the domain boundary and modifying the system matrix.

References

[1] Douglas N. Arnold et al. "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems". In: *SIAM J. Numer. Anal.* 39.5 (May 2001), pp. 1749–1779. ISSN: 0036-1429. DOI: 10.1137/S0036142901384162. URL: http://dx.doi.org/10.1137/S0036142901384162.

[2] Sougata Biswas and Jiten C. Kalita. "Moffatt vortices in the lid-driven cavity flow". In: *Journal of Physics Conference Series*. Vol. 759. Journal of Physics Conference Series. Oct. 2016, p. 012081. DOI: 10.1088/1742-6596/759/1/012081. arXiv: 1601.05186 [physics.comp-ph].

[3] Long Chen. *Finite Element Methods For Maxwell Equations*. URL: https://www.math.uci.edu/~chenlong/226/FEMMaxwell.pdf.

[4] Bernardo Cockburn, Guido Kanschat, and Dominik Schötzau. "A Locally Conservative LDG Method for the Incompressible Navier-Stokes Equations". In: *Mathematics of Computation* 74.251 (2005), pp. 1067–1095. ISSN: 00255718, 10886842. URL: http://www.jstor.org/stable/4100171.

[5] Bernardo Cockburn, Guido Kanschat, and Dominik Schötzau. "A Note on Discontinuous Galerkin Divergence-free Solutions of the Navier–Stokes Equations". In: *Journal of Scientific Computing* 31.1 (May 2007), pp. 61–73. ISSN: 1573-7691. DOI: 10.1007/s10915-006-9107-7. URL: https://doi.org/10.1007/s10915-006-9107-7.

[6] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities". In: *International Journal for Numerical Methods in Engineering* 79.11 (2009), pp. 1309–1331. DOI: 10.1002/nme.2579. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.2579. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579.

[7] Nuri Aksel H.Joseph Spurk. *Fluid Mechanics*. Springer, 2008. ISBN: 978-3-540-73536-6. DOI: https://doi.org/10.1007/978-3-540-73537-3. URL: https://www.springer.com/de/book/9783540735366.

[8] Ralf Hiptmair. *Numerical Methods for Partial Differential Equations*. URL: https://www.sam.math.ethz.ch/~grsam/NUMPDE/NUMPDE18.pdf.

[9] Robert C. Kirby et al. "Common and unusual finite elements". In: *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Ed. by Anders Logg, Kent-Andre Mardal, and Garth Wells. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 95–119. ISBN: 978-3-642-23099-8. DOI: 10.1007/978-3-642-23099-8_3. URL: https://doi.org/10.1007/978-3-642-23099-8_3.

[10] *LehrFEM++*. URL: https://github.com/craffael/lehrfempp.

[11] *Sparse matrix manipulations*. URL: https://eigen.tuxfamily.org/dox/group__TutorialSparse.html.

[12] "The Space H(div;Ω)". In: *An Introduction to Sobolev Spaces and Interpolation Spaces*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 99–101. ISBN: 978-3-540-71483-5. DOI: 10.1007/978-3-540-71483-5_20. URL: https://doi.org/10.1007/978-3-540-71483-5_20.

[13] C. Varsakelis and Y. Marichal. "Numerical approximation of elliptic interface problems via isoparametric finite element methods". In: *Computers & Mathematics with Applications* 68.12, Part A (2014), pp. 1945–1962. ISSN: 0898-1221. DOI: https://doi.org/10.1016/j.camwa.2014.10.001. URL: http://www.sciencedirect.com/science/article/pii/S0898122114004738.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Divergence Conforming Discontinuous Galerkin FEM for the Stokes Problem

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	First name(s):
Rohner	Tobias

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	Signature(s)
Heiden, 1. Sept. 2019	D. Roh

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.