Department of Civil, Geo and Environmental Engineering
Chair for Computation in Engineering
Prof. Dr. rer. nat. Ernst Rank

Multi-level *hp*-FEM and the Finite Cell Method for the Navier-Stokes equations using a Variational Multiscale Formulation

Philipp Kopp

Master's thesis

for the Master of Science program Computational Mechanics

Author:	Philipp Kopp
Supervisors:	Prof. Dr. rer. nat. Ernst Rank
	Prof. Ph.D. Victor Manuel Calo
Advisors:	Dr.-Ing. Nils Zander
	Dr.-Ing. Stefan Kollmannsberger
Date of issue:	20. October 2016
Date of submission:	20. April 2017

Involved Organisations

Chair for Computation in Engineering
Department of Civil, Geo and Environmental Engineering
Technische Universität München
Arcisstraße 21
D-80333 München

Declaration

With this statement I declare, that I have independently completed this Master's thesis. The thoughts taken directly or indirectly from external sources are properly marked as such. This thesis was not previously submitted to another academic institution and has also not yet been published.

München, April 20, 2017

Philipp Kopp

Philipp Kopp
e-Mail: philipp.kopp@tum.de

Contents

Chapter 1

Introduction

In mechanical and structural engineering, the typical product development cycle consists of iterating design and analysis until both, shape and properties match well enough the requirements. Historically, the analysis of mechanical properties was done by experiments, but with increasing hardware performance nowadays they are mostly replaced by numerical computations. Making this process efficient requires developing methods that robustly deliver accurate results and can deal with the geometry from design software packages.

Recently, attention has been drawn towards the transition from design to analysis. Traditional methods such as the Finite Element Method require a mesh to perform computations. Converting the geometry from Computer Aided Design (CAD) tools to such a mesh has two major drawbacks. First, manual adjustment is still required in many situations despite the existence of advanced automatic tools. Second, this step is often connected with a loss of information, as the original geometry is approximated by lower order polynomials. New technologies such as the isogeometric analysis (IGA), see e.g. Cottrell et al. [2009], or the Finite Cell Method (FCM), introduced in Parvizian et al. [2007], try to tackle this problem. Isogeometric analysis seeks to use the same geometric description from CAD systems for the analysis. While this was shown to be very successful in the case of thin-walled structures, problems arise for volumetric geometries, as they are usually stored in a boundary representation (BREP) format. The Finite Cell Method, in contrast, requires only the information whether a given point is inside or outside the solid model and is, therefore, able to handle arbitrary input shapes. The use of high order polynomials as basis functions provides the approximation power to obtain highly accurate solutions.

Much research has been devoted over the last years into further developing the Finite Cell methodology. Some major contributions towards making FCM a general tool for computational mechanics are the extensions to e.g. large deformation [Schillinger et al., 2012], structural dynamics [Elhaddad et al., 2015], contact mechanics [Bog et al., 2015] or structural optimization [Parvizian et al., 2011]. The critical aspect in the FCM is the accurate integration which is particularly challenging, as a discontinuity in the integrand has to be captured (see section 2.3). Various methods have been proposed to address this issue. In Joulaian et al. [2016], they use the moment fitting technique to compute an integration rule that captures the discontinuity. Another direction is taken by Kudela et al. [2016], introducing the "smart tree" method, which attempts to map the cells of a conventional space tree partitioning to approximate the geometry to a specified accuracy. Both methods deliver excellent results compared to the conventional space tree partitioning.

Closely related to the success of the Finite Cell Method is the introduction of the multi-level *hp*-method by Zander et al. [2015] (see also section 2.2). This multi-level description circumvents the need to constrain hanging nodes as in alternative *hp*-methods, such as in Demkowicz et al. [1989]. Because the Finite Cell method can use structured (e.g. Cartesian) meshes, the multi-level *hp*-scheme adds the ability to refine the discretization in a very simple manner. In Zander et al. [2016], an extension to three-dimensional problems was presented.

To date, structural mechanics was the application of choice for the combination of Finite Cell Method and multi-level *hp*-adaptivity. The aim of this thesis is to extend applications to include fluid mechanics. To this end, we extend the Finite Cell methodology to encompass other equations. Interpreting the contribution from the weak form on the fictitious domain purely as a stabilization term allows us to use simpler equations. Although Cai [2013], Xu et al. [2015] and Kamensky et al. [2015] used the FCM in the context of fluid-flow modelling, only moderate polynomial degrees were used. Our goal is to extend this framework to higher polynomial degrees and to combine FCM with the aforementioned multi-level *hp*-scheme.

Solving the incompressible Navier-Stokes equations with the Finite Element Method has three main difficulties. First, the non-symmetric operator is not well-suited for standard Bubnov-Galerkin methods, leading to non-physical oscillations for under-resolved discretizations. Second, the incompressibility constraint introduces a pressure field as a Lagrange multiplier. Thus, the problem has a saddle point structure and is not positive definite. When no stabilization is used, care has to be taken in selecting discretization pairs for velocity and pressure. Last but not least, the solution for high Reynolds numbers is characterized by a turbulent behavior. Failing to resolve the details leads to a wrong large-scale solution. To address these problems, a residual-based Variational Multiscale (VMS) method is used as both, stabilization scheme and turbulence model (see section 4.1).

The outline of this thesis is as follows. Chapter 2 introduces the general theory and notation related to the discretization methods used throughout this work. To obtain some insight into the performance of high-order Finite Elements on advection dominated problems, chapter 3 provides studies on the one- and two-dimensional advection-diffusion equation. These include stabilized formulations where we address the definition of the stabilization parameter τ in particular. We introduce a *p*-adapted definition that delivers promising results on a simple one-dimensional example. Chapter 4 continues with introducing the residual-based Variational Multiscale formulation for the Finite Cell method. After a general introduction to the theory, the definitions of space- and time discretizations follow. To demonstrate the potential of the presented methods, an application to common benchmark problems such as the lid driven cavity and the flow around a cylinder follow in chapter 5. A detailed comparison of drag and lift coefficients to reference solutions allows us to assess the quality of the method. Finally, a summary of the results and an outlook of our future work conclude this thesis in chapter 6.

Chapter 2

The multi-level *hp*-adaptive Finite Cell Method

This chapter provides a general overview of the discretization methods used throughout the thesis. This includes an introduction to the Finite Cell method and the multi-level *hp*-scheme as well as a discussion of related difficulties.

To introduce the general Finite Element framework, a domain composed of an interior Ω and a boundary Γ is considered. The boundary is partitioned into a Dirichlet part Γ_D and a Neumann part Γ_N with $\Gamma_N \cup \Gamma$ and $\Gamma_N \cap \Gamma = \emptyset$. On this domain, a linear equation of the type

$$\mathcal{L}u = f \qquad \text{on } \Omega \tag{2.1}$$

$$u = g \qquad \text{on } \Gamma_D \tag{2.2}$$

$$\mathcal{B}u = h \qquad \text{on } \Gamma_N \tag{2.3}$$

is defined, where $\mathcal{L}$ and $\mathcal{B}$ denote two linear differential operators acting on the domain interior and the Neumann boundary, respectively. Without loss of generality, $\mathcal{L}$ is assumed to contain only up to second order derivatives. $\mathcal{B}$ contains the boundary terms arising later in the weak form from integration by parts.

To approximate eq. (2.1) using the Finite Element method, we obtain a weak form by requiring the equation to hold under the L^2 inner product with any element w from a test function space $\mathcal{W}$:

$$\langle w, \mathcal{L}u \rangle = \langle w, f \rangle \qquad \forall w \in \mathcal{W}, \tag{2.4}$$

where $\langle \cdot, \cdot \rangle$ denotes the L^2 inner product on Ω. Integrating terms in $\mathcal{L}u$ by parts yields the weak form

$$B(w, u) = F(w) \qquad \forall w \in \mathcal{W} \tag{2.5}$$

with the bilinear form B and the linear functional F. Now, the continuity requirements on u are weaker as derivatives were transferred to w. Thus, the function spaces in eq. (2.5) are the following:

$$\mathcal{W} = \{\, v(\mathbf{x}) \in H^1 \mid v(\mathbf{x}) = 0 \text{ on } \Gamma_D \,\} \tag{2.6}$$

$$\mathcal{S} = \{\, v(\mathbf{x}) \in H^1 \mid v(\mathbf{x}) = g(\mathbf{x}) \text{ on } \Gamma_D \,\}, \tag{2.7}$$

where H^1 denotes the Sobolev space of first order. The terms on the right hand side are now:

$$F(w) = \langle w, f \rangle + \langle w, \mathcal{B}u \rangle_\Gamma$$

where

$$\langle w, \mathcal{B}u \rangle_\Gamma = \langle w, \mathcal{B}u \rangle_{\Gamma_N} + \langle w, \mathcal{B}u \rangle_{\Gamma_D} = \langle w, h \rangle_{\Gamma_N}, \tag{2.8}$$

due to the definition of the test space $\mathcal{W}$ in (2.6) and the Neumann boundary condition in (2.3).

The space $\mathcal{S}$ incorporates the Dirichlet boundary condition from eq. (2.2). Using a function $\bar{g} \in \mathcal{S}$ defined on Ω that takes the value of g at Γ_D, we can split any $v \in \mathcal{S}$ into

$$v = u + \bar{g} \qquad v \in \mathcal{V}, \tag{2.9}$$

where $\mathcal{V}$ contains only functions that vanish on Γ_D. Substituting eq. (2.9) into eq. (2.5) yields

$$\begin{aligned} B(w, u + \bar{g}) &= F(w) && \forall w \in \mathcal{W} \\ \Leftrightarrow B(w, u) &= F(w) - B(w, \bar{g}) && \forall w \in \mathcal{W}, \end{aligned}$$

which is often referred to as the strong way of imposing boundary conditions. However, sometimes it is preferable or even necessary to impose Dirichlet data in a weak manner. Instead of restricting the function spaces at the boundary, the weak form is modified in a way that the solution fulfills the boundary condition in a weak sense. This is particularly important in the Finite Cell method (see sections 2.3 and 2.4) where boundaries are not conforming to the Finite Element mesh.

The solution to the strong form (eq. 2.1) is also a weak solution. This is often referred to as the consistency property. To show that the exact solution is, in fact, the only solution to the weak problem, it is sufficient to prove continuity

$$\exists \lambda \text{ s.t. } |B(w, v)| \leq \lambda \|w\| \, \|u\| \quad \forall w, v \in \mathcal{W}$$

and coercivity

$$\exists \mu \text{ s.t. } B(v, v) \geq \mu \|v\|^2 \quad \forall v \in \mathcal{W},$$

of the bilinear form. Here, $\|\cdot\|$ denotes the norm associated with $\mathcal{W}$ (e.g. $\|\cdot\|_{\mathcal{H}^1}$). The uniqueness follows then from the Lax-Milgram lemma [Quarteroni and Valli, 1994].

In the next step, we discretize the weak form by choosing finite dimensional subspaces $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{W}^h \subset \mathcal{W}$, which allows functions to be described in terms of discrete bases:

$$u \approx u^h = N_i^u(\mathbf{x})\, \hat{u}_i \tag{2.10}$$

$$w \approx w^h = N_i^w(\mathbf{x})\, \hat{w}_i, \tag{2.11}$$

where N_i denotes the basis function with the index i. The individual choice of $\mathcal{V}^h$ and $\mathcal{W}^h$ depends on many factors. Apart from practical arguments such as implementational effort, test and trial spaces are selected according to the following considerations: The trial or interpolation space is responsible for approximability, that is, it needs to be capable of representing the exact solution as good as possible. The test or weighting space, however, controls the stability of the solution, determining how close the approximation is to the optimal one (in the discrete space). From stability and approximability follows convergence. More details can be found in e.g. Arnold et al. [1984]. The fundamental error estimate for a discretization method is then stated as

$$\|u - u^h\| \leq \frac{\lambda}{\mu} \inf_{v \in \mathcal{W}^h} \|u - v\|, \tag{2.12}$$

which is often denoted as Céa's lemma. In other words, the error in the discrete solution is bounded from above by a constant multiplying the optimal approximation error in the discrete space.

The conventional (or Bubnov-Galerkin) choice of selecting $\mathcal{V}^h = \mathcal{W}^h$ yields optimal results for symmetric problems with well behaved coefficients. In that case, the bilinear form defines an inner product $\langle \cdot, \cdot \rangle_B : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ and the solution represents a best-fit in the (semi-) norm induced by this inner product:

$$\|v\|_B = \sqrt{\langle v, v \rangle_B} = \sqrt{B(v, v)} \tag{2.13}$$

In that case, we have [Hughes, 2000]

$$\|u - u^h\|_B \leq \inf_{v \in \mathcal{W}^h} \|u - v\|_B. \tag{2.14}$$

This property is called the "best approximation property" and is one of the main reasons for the success of the classical Finite Element Method for symmetric problems, such as linear elasticity. Appendix A provides a short derivation of the best approximation property and interprets the results in a Variational Multiscale setting (following section 3.2).

Now, eq. (2.5) can be discretized using the standard basis vectors for $\hat{w}_i$ and separating the degrees of freedom $\hat{u}_i$:

$$B(N_i, N_j)\, \hat{u}_j = F\,(N_i) \qquad i = 1 \ldots n.$$

Employing a suitable quadrature rule, B and F are approximated numerically, and a matrix equation of the form

$$\mathbf{K}\hat{\mathbf{u}} = \mathbf{f} \tag{2.15}$$

is obtained. After solving eq. (2.15) with an appropriate linear algebra tool, the solution can be recovered using eq. (2.10) - (2.11).

2.1 Hierarchical Finite Elements

Polynomial spaces are particularly effective at approximating smooth solutions [Szabó and Babuška, 1991]. Various possibilities for defining a polynomial basis exist. Conventional Finite Element implementations commonly use Lagrange shape functions. From an engineering point of view this has several advantages. First, we can interpret each unknown as the field value at the node and second, these bases have the partition of unity property. A different approach named the p-version was introduced in Babuska et al. [1981] and uses integrated

Legendre polynomials, defined as

$$\begin{aligned} N_1(\xi) &= \frac{1}{2}(1-\xi) \\ N_2(\xi) &= \frac{1}{2}(1+\xi) \\ N_i(\xi) &= \phi_{i-1} \qquad i = 3 \dots p+1, \end{aligned}$$

where ϕ is defined in terms of the Legendre polynomials L_i:

$$\phi_j = \frac{1}{\sqrt{4j-2}}\big(L_j(\xi) - L_{j-2}(\xi)\big).$$

This choice yields the desirable orthogonality property of the shape function derivatives rendering stiffness matrices diagonal. Unfortunately, the sparsity is not preserved in multiple dimensions, where function spaces are usually constructed by a tensor product. However, stiffness matrices are better conditioned compared to using Lagrange basis functions. Also, the hierarchical nature of the integrated Legendre basis allows to easily remove basis functions from the tensor product space while preserving optimal convergence rates. The remaining space is called the trunk space. More information can be found in e.g. Szabó and Babuška [1991].

Furthermore, shape functions in the tensor product space can be associated to topological components. This is possible, as linear modes remain in the 1D bases at all polynomial degrees and higher order 1D basis functions have no support at the nodes. In that way, nodal, edge, face and volume modes can be obtained. Figure 2.1 illustrates this concept.

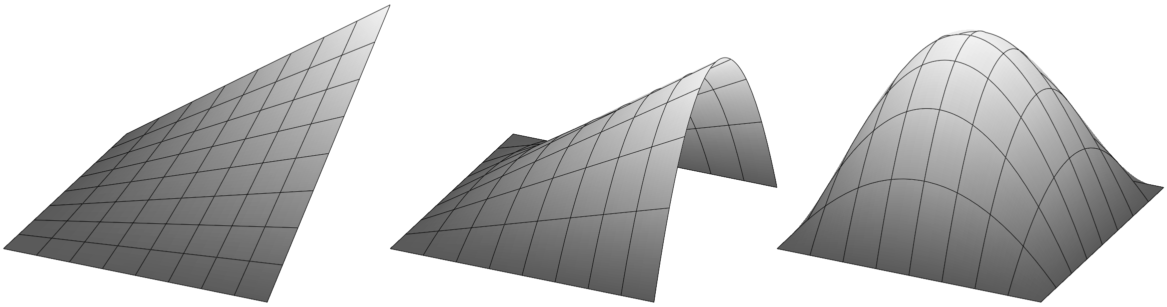


Figure 2.1: Association of basis functions to topological components.

2.2 Multi-level *hp*-FEM

In fluid- and structural mechanics solutions commonly possess strong gradients and singularities (e.g., at re-entrant corners). In this case, the exponential convergence rate of p-FEM deteriorates to an algebraic convergence [Gui and Babuška, 1986a]. However, an exponential convergence can be recovered by combining h- and p-refinement. This hp-version refines towards the discontinuity and thereby localizes its effect while using the full polynomial approximation power on the rest of the domain [Gui and Babuška, 1986b].

While elevating the polynomial degree is relatively simple, performing h-refinement for high-order elements is not straightforward. The most common choice is a refinement by replacement where smaller elements substitute coarse elements. Although there are other more flexible approaches, in this work only a space tree refinement type is considered (*i.e.* elements are always bisected in each dimension when refined). After the replacement step, the mesh is non-conforming and therefore constraints have to be added. Constructing these can be cumbersome as dependencies can get very complex and opaque. This can result, e.g., in special cases where circular dependencies block the refinement process leading to deadlocks if not handled.

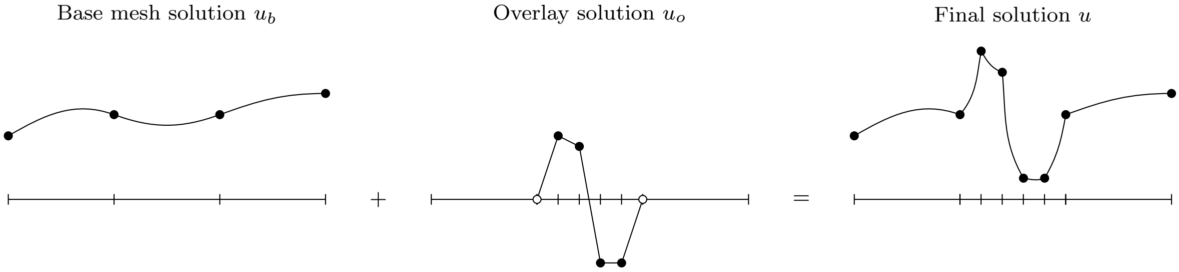


Figure 2.2: Refinement by superposition [Rank, 1992; Zander, 2016].

To greatly simplify that process, the multi-level *hp*-method was introduced by Zander et al. [2015]. Instead of replacing elements, the refined space is constructed by a superposition of the original mesh with overlay meshes. This is an extension of the *hp-d* refinement scheme [Rank, 1992], where only linear overlay elements are used (see fig. 2.2).

To construct a valid Finite Element space with a multi-level *hp*-basis, two simple rules are sufficient. First, to obtain compatibility, the method imposes homogeneous Dirichlet boundary conditions on the superposed patches. Using an integrated Legendre basis, this translates to deactivating topological components on the boundary. These are identified by comparing the refinement levels of adjacent elements. That means, in three dimensions nodes, edges and faces that are connected to elements with a different refinement level are deactivated. Second, to ensure linear independence of the basis, all topological components that have active subcomponents are deactivated. For example, the interior degrees of freedom of a refined hex (that has 8 sub-hexes) are inactive. Figure 2.3 illustrates the concept in one and two dimensions.

Topological relations simplify the imposition of these rules. These relations include adjacency lists containing information about connected elements as well as information about subcomponents. As an example, a refined hex has eight children after its refinement. All faces that were divided in the process have four sub-face, all edges have two sub-edges and each node has one sub-node. Newly created components have no children. After this refinement step, the lists of adjacent elements are updated for all topological components (with respect to the new, refined setting). With this, the application of multi-level *hp* rules, as described above, are straightforward. A detailed description of the multi-level *hp* theory and implementation can by found in Zander [2016].

There are two possibilities to ensure linear independence in the linear modes. In addition to deactivating nodes on the base level, it is also possible to deactivate them on the leaf elements. Both variants yield the same space, but in the first case basis functions have a smaller support which is favorable in the solution of the linear system, see, e.g., Pardo et al.

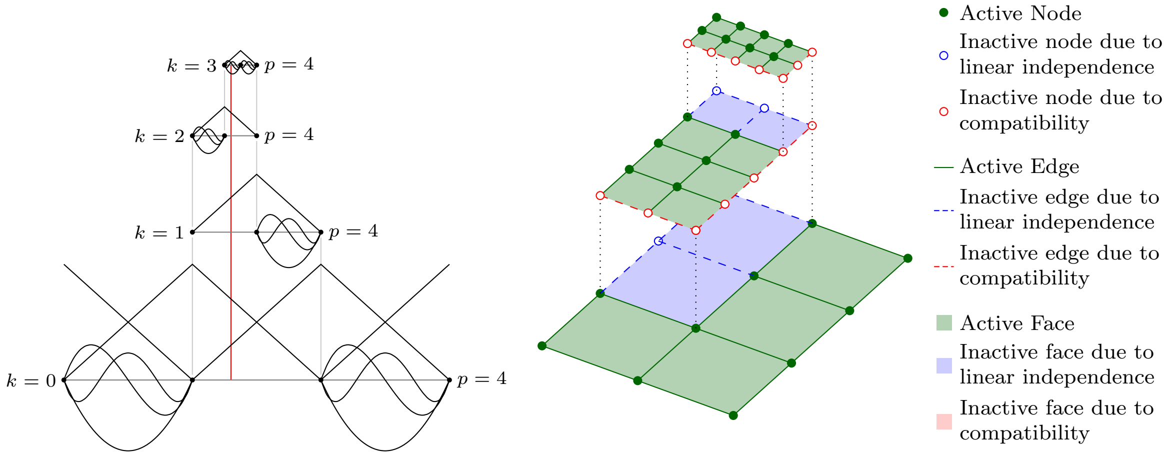

$k = 3$
$p = 4$
$k = 2$
$p = 4$
$k = 1$
$p = 4$
$k = 0$
$p = 4$
● Active Node
○ Inactive node due to linear independence
○ Inactive node due to compatibility
— Active Edge
- - Inactive edge due to linear independence
- - Inactive edge due to compatibility
🟩 Active Face
🟦 Inactive face due to linear independence
🟥 Inactive face due to compatibility

Figure 2.3: The multi-level *hp* concept: One- and two-dimensional cases [Zander, 2016].

[2012].

2.3 The Finite Cell Method

Although the p-version provides superior accuracy compared to the h-version for a fixed mesh size, the problem of mesh generation becomes even more severe. That is, the convergence strongly depends on how good the mesh approximates the boundary. While h-meshes automatically converge to the exact description by choosing finer segments, high-order elements need to be constructed using the blending function method such that the approximation is of the same order as the polynomial degree. The blending method seeks to control the error originating from an insufficient boundary approximation such that it does not dominate the discretization error.

This task is challenging for mesh generators, thus, it is desirable to circumvent the meshing procedure. The Finite Cell Method avoids complex mesh generation, as the physical domain is embedded into a larger, fictitious domain. The shape of the fictitious domain can be chosen arbitrarily (e.g. as the bounding box of the geometry) such that meshing becomes trivial. To recover the effect of the geometry, we modify the differential equation as it is now solved on a larger domain. To this end, an indicator function $\alpha(\mathbf{x})$ is defined as

$$\alpha(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_{phys} \\ \epsilon & \text{if } \mathbf{x} \in \Omega_{fict} \end{cases}, \tag{2.16}$$

where ϵ is a parameter that is chosen as small as possible. The setting is sketched in fig. 2.4.

At this stage, two points of view exist to develop the Finite Cell Method. In the physical interpretation, the material parameters in the weak form (eq. 2.5) are multiplied by α. This way, the solutions of the modified problem coincide with the solution of the original problem inside Ω_{phys}. As a result, the influence of material outside the domain of interest is eliminated. Although ϵ can be arbitrarily small, in practice the range is limited by numerical stability. Alternatively, we can view this process as a simple multiplication of the weak form by α. While the two interpretations are equivalent for linear elasticity, the second view allows for

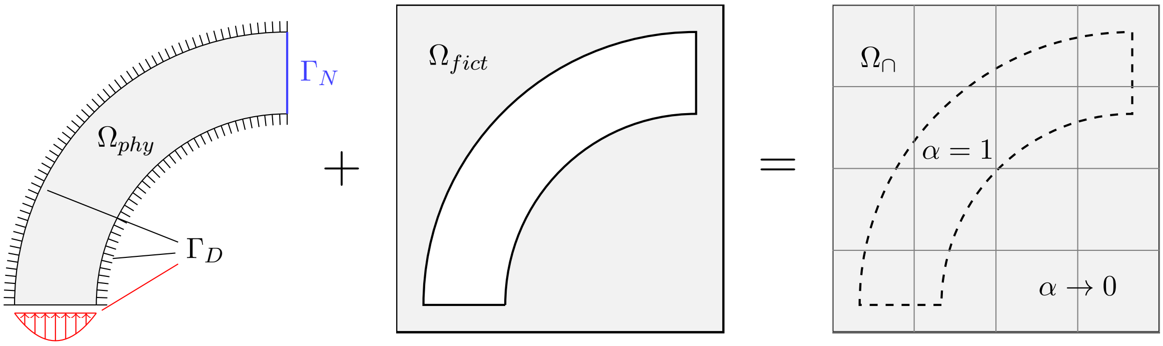


Figure 2.4: The Finite Cell Method: General concept.

more flexibility concerning the application to other equations.

Although an analysis of the convergence was performed in Dauge et al. [2015], it has not yet been explained how a Finite Cell formulation can be developed in general for a given partial differential equation. Specifically, derivations that include a multiplication by $\alpha(\boldsymbol{x})$ are problematic, as the question arises how this operation relates to the integration by parts. In fact, it is crucial to multiply the final version of the weak form, as otherwise a derivative would be acting on a discontinuous function. In Dauge et al. [2015], two different bilinear forms are defined on Ω_{phys} and Ω_{fict}, respectively, where the one on Ω_{fict} is multiplied by ϵ. A right hand side is only considered in Ω_{phys}. As a result, the dependency on $\boldsymbol{x}$ is not present and the integration by parts is not a problem. Our attempt to extend these ideas for formulating Finite Cell Methods in general is the following. In the construction of the weak form, the L^2 inner product is taken only on Ω_{phys} but the function spaces are extended to $\Omega_\cap$. Now, the uniqueness of the solution is lost, as no equation is specified on Ω_{fict} rendering the problem ill-posed. This can be resolved by defining an additional bilinear form on Ω_{fict}. To ensure convergence, the magnitude of the additional terms needs to be small enough, such that the modeling error does not dominate the overall error. In the traditional Finite Cell Method the original bilinear form is used and multiplied by ϵ. Note, that other choices are possible and could be advantageous in certain settings. In chapters 4 and 5, the weak form of the Navier-Stokes equation is extended to $\Omega_\cap$ by defining a diffusion problem on Ω_{fict}. A different approach to regain well-posedness is to consider only basis functions with a support intersecting Ω_{phys}. This way, the bilinear form on Ω_{fict} would arise as a stabilization technique for badly cut cells (where the intersection is small).

The first main difficulty in the Finite Cell Method from a practical point of view is to integrate the modified bilinear form accurately. Standard quadrature schemes are usually constructed by considering a polynomial approximation of the function. They perform therefore poorly at handling the present discontinuities. The simplest and most flexible approach is to refine the integration mesh towards the boundary using a space tree. This allows capturing the discontinuity by distributing integration points in its close vicinity. The drawback is that many integration points are needed to achieve the desired convergence, and therefore the integration time is considerably increased. In this work two integration methods are considered: A conventional refinement based on a space tree and the blended partitioning approach introduced in Kudela et al. [2015].

2.4 Non-conforming boundary conditions

The purpose of the Finite Cell Method is to decouple the computation domain from the geometry of the physical problem. As a result, boundaries are not conforming to the Finite Element mesh. While this does not affect the (consistent) integration of Neumann boundary conditions, now Dirichlet boundary conditions have to be imposed weakly. To this end, the weak form is extended such that a modified equation of the type

$$\tilde{B}(w,u) = \tilde{F}(w) \qquad \forall w \in \mathcal{W} \tag{2.17}$$

is obtained. Now, the test and trial spaces are not required to satisfy the boundary conditions (*cf.* eq. 2.6 - 2.7). Two methods are adressed in this work: The penalty method and Nitsche's method.

The penalty method extends the weak form as follows:

$$B(w,u) + \beta \langle w, u-g \rangle_{\Gamma_D} = F(w) \qquad \forall w \in \mathcal{W}, \tag{2.18}$$

such that

$$\begin{aligned} \tilde{B}(w,u) &= B(w,u) + \beta \langle w, u \rangle_{\Gamma_D} \\ \tilde{F}(w) &= F(w) + \beta \langle w, g \rangle_{\Gamma_D}. \end{aligned}$$

The penalty parameter β is chosen as large as possible. While being very simple and convenient to implement, several drawbacks limit the applicability of the penalty method. First, the method lacks consistency as the weighting space $\mathcal{W}$ is not restricted at the boundary. Therefore, the boundary integral on the Dirichlet part in eq. (2.8) does not vanish. Moreover, the range of floating point numbers is limited, such that a stable computation of the solution is only possible when β lies within a certain range, making its selection a challenging task.

An improvement of the penalty method can be obtained by adding extra terms. In Nitsche's method, the following formulation is used [Nitsche, 1971]:

$$B(w,u) - \langle w, \mathcal{B}u \rangle_{\Gamma_D} - \langle \mathcal{B}^* w, u-g \rangle_{\Gamma_D} + \beta \langle w, u-g \rangle_{\Gamma_D} = F(w) \qquad \forall w \in \mathcal{W},$$

or in terms of eq. (2.17):

$$\begin{aligned} \tilde{B}(w,u) &= B(w,u) - \langle w, \mathcal{B}u \rangle_{\Gamma_D} - \langle \mathcal{B}^* w, u \rangle_{\Gamma_D} + \beta \langle w, u \rangle_{\Gamma_D} \\ \tilde{F}(w) &= F(w) - \langle \mathcal{B}^* w, g \rangle_{\Gamma_D} + \beta \langle w, g \rangle_{\Gamma_D}. \end{aligned}$$

With this extension, the weak form is symmetric, consistent, and the penalty term is not required to be as large. Although β has a longer range where it has practically not influence on the results, the choice of a parameter remains.

Chapter 3

Multi-level *hp*-FEM for advection-diffusion problems

A first step towards extending the methodology introduced in chapter 2 to the field of fluid mechanics is to study a scalar advection-diffusion equation. This allows to investigate the behavior in a setting where the operators are not symmetric, and therefore the Galerkin orthogonality does not exist. Along with the theory in this chapter, some results are presented for illustration. These were taken from Zander [2016] and are based on an advection-diffusion solver that was implemented in the context of the present thesis.

The model problem of scalar advection-diffusion reads

$$\begin{aligned} \mathbf{a} \cdot \nabla u - \nabla \cdot (k \nabla u) &= f \qquad \text{on } \Omega && (3.1)\\ u &= g \qquad \text{on } \Gamma_D && (3.2)\\ k \nabla u \cdot \mathbf{n} &= h \qquad \text{on } \Gamma_N . && (3.3) \end{aligned}$$

It describes the evolution of a quantity u being transported by a velocity field $\mathbf{a}$ while also being subject to a diffusion process with a diffusivity k. In general, k is a second order tensor that can be simplified to a scalar in the isotropic case. Following the standard procedure presented in chapter 2, the weak form becomes

$$\langle w, \mathbf{a} \cdot \nabla u \rangle + \langle \nabla w, k \nabla u \rangle = \langle w, f \rangle + \langle w, h \rangle_{\Gamma_N} \qquad \forall w \in \mathcal{W} \tag{3.4}$$

or in the form of eq. (2.5):

$$B^{cd}(w, u) = F^{cd}(w) \qquad \forall w \in \mathcal{W}. \tag{3.5}$$

The first order ("advective") term renders the bilinear form arising from eq. (3.4) non-symmetric in u and w. As a consequence, eq. (3.5) does not define an inner product. Orthogonality and the best approximation properties, as in eq. (2.14) can therefore not be derived.

3.1 Deficiencies of Bubnov-Galerkin discretizations

The standard Finite Element solution exhibits spurious oscillations in case the advection is dominant. A first indicator that was already given in the introduction of this chapter is the loss of optimality compared to symmetric problems. This section gives more details and points of view on this phenomenon.

To characterize the quality of a Bubnov-Galerkin approximation of advection-diffusion problems, an element Péclet number is defined as

$$Pe^h = \frac{ah}{2k}.$$

It relates the mesh size h, the advection velocity, and the diffusivity in a one-dimensional setting (see eq. 3.1). For values of $Pe^h > 1$, the solution shows "instabilities" at boundary layers. The term instability can be understood in that way that the constant in eq. (2.12) gets arbitrarily large when $Pe^h \to \infty$. However, this does not give any insight or explanation to the problem.

In the context of the Variational Multiscale (VMS) method (that will be addressed in more detail in section 3.2), the oscillations are interpreted as an insufficient resolution of the small scale solution characteristics. The coupling term in the scale separation vanishes for symmetric bilinear forms in a Bubnov-Galerkin discretization when the corresponding B-norm is chosen. In other words, the small scales remain unresolved in both cases, but the large scales are influenced only in the non-symmetric case. This is demonstrated in appendix A in more detail.

Another perspective on the problem is to interpret oscillations as a result of non-matching boundary conditions. For $Pe^h \to \infty$ the diffusion term vanishes, leaving a pure transport equation. In this case, boundary conditions can be imposed only at the inflow (or only at the outflow) boundary. Therefore, keeping original boundary conditions that are not valid for pure transport phenomena renders the problem ill-posed. While this occurs only in the limit case of $k \to 0$, it is important that the discretization might be too coarse to "see" the diffusive nature of the problem. This transition is defined by $Pe^h = 1$. Interestingly, the eigenvalues of element matrices (of linear elements) become complex for $Pe^h > 1$, confirming this interpretation.

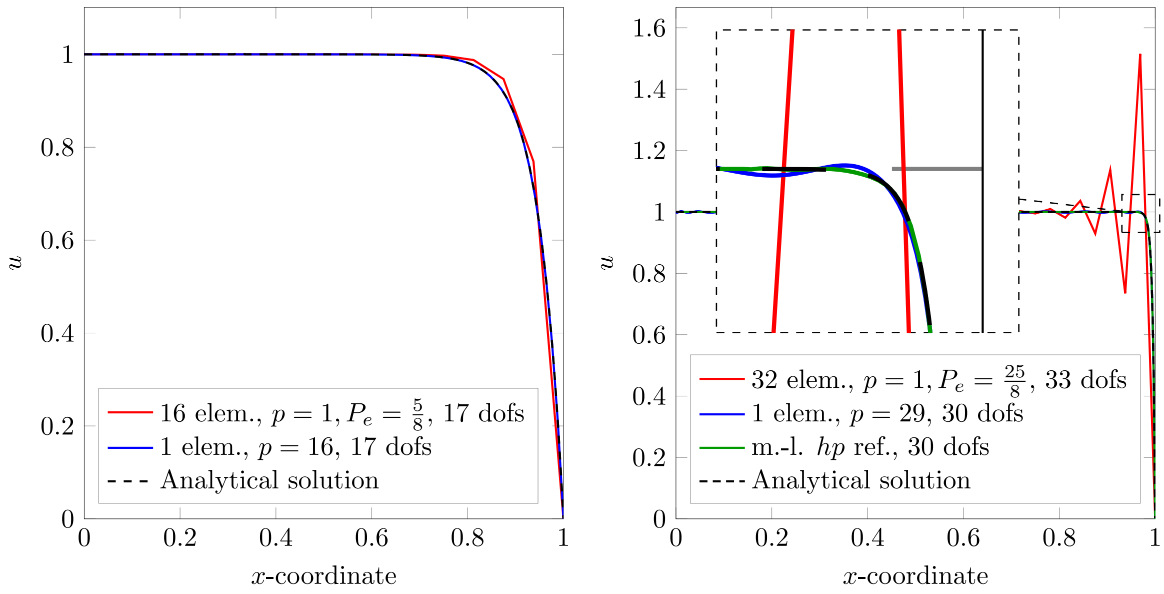
An analysis for high-order Finite Elements can be found in e.g. Schwab and Suri [1996] and Cai et al. [2014].

3.1.1 Example 1: Advection-diffusion in 1D

To obtain a first impression on the performance of numerical schemes applied to advection-diffusion equations a 1D benchmark with $u(0) = 1$ and $u(1) = 0$ is studied. The physical parameters a and k are chosen to be constant on Ω. The exact solution is given by

$$u = \frac{e^{\frac{a}{k}x} - e^{\frac{a}{k}}}{1 - e^{\frac{a}{k}}}.$$

As $u \in C^{\infty}$, a good performance of the p-version is to be expected.



(a) $a = 1$, $k = 5 \cdot 10^{-2}$

(b) $a = 1$, $k = 5 \cdot 10^{-3}$

Figure 3.1: Finite Element solutions to one-dimensional advection-diffusion problem [Zander, 2016].

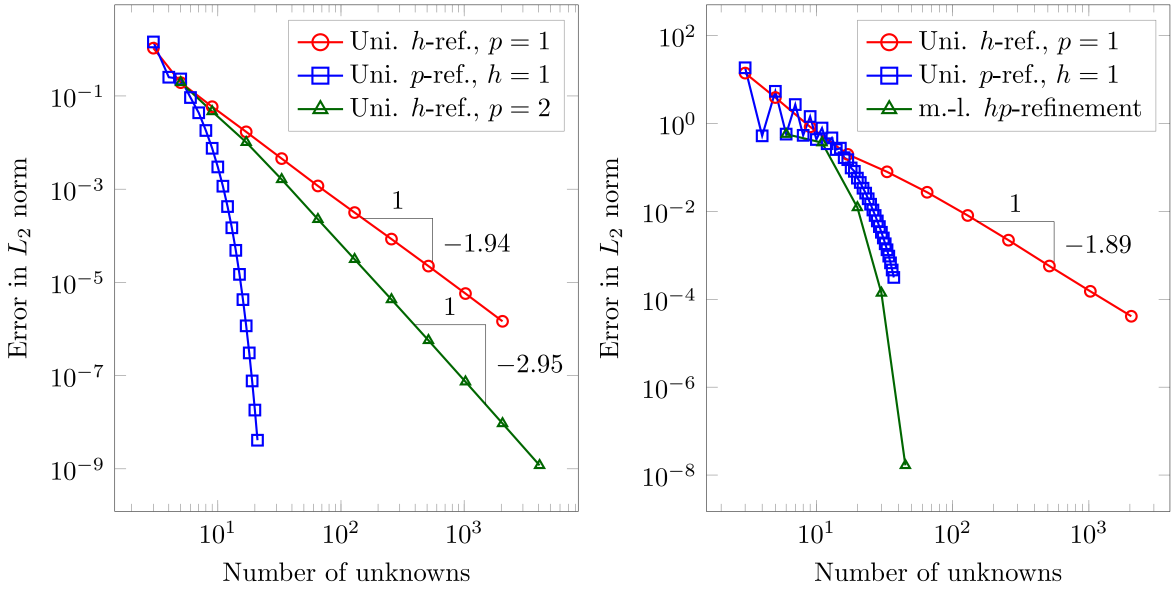
Figure 3.1 compares the solution to this benchmark for different Finite Element methods and Péclet numbers. While in fig. 3.1a an h- and a p-version are compared on a very low element Péclet number ($Pe^h < 1$), in fig. 3.1b the diffusion is reduced such that the problem becomes unstable (for the given h- and p-methods). Additionally, the performance of an hp-discretization with a graded polynomial degree of

$$p_{\text{left}} = \left\lfloor \frac{s}{2} \right\rfloor + 1 \quad \text{and} \quad p_{\text{right}} = 2s + 1,$$

for an element with refinement level s (*cf.* Zander [2016]) is shown.

It can be seen that the p-version provides a better approximation to the exact solution as it is sufficiently smooth. This is also expressed in the corresponding convergence of the L^2 error, depicted in fig. 3.2. However, also the stability of the displayed methods differs drastically. While in fig 3.1b the solution for $p = 1$ and 32 element exhibits strong oscillations around the boundary layer, the solution on one element with $p = 29$ remains relatively stable (although using fewer degrees of freedom). Stability also influences the convergence by extending the pre-asymptotic range corresponding to the constant in Céa's lemma (see eq. 2.12). Furthermore, the p-method shows an odd-even pattern in the L^2 error. This coincides with the results presented in Cai [2013] where a different stabilizing effect for odd and even polynomial degrees was found.

Moreover, an intelligent combination of h- and p-refinement is superior in this example. As the p-version requires a high polynomial degree to reach the regime of exponential convergence, the method is very limited in practice. The hp-version, however, requires only a moderate degree. From this follows, that a robust hp-scheme is indispensable when both, boundary layers (and internal layers) as well as smooth solution characteristics on other parts of the domain shall be accurately resolved by a Finite Element method.



(a) $a = 1$, $k = 5 \cdot 10^{-2}$

(b) $a = 1$, $k = 5 \cdot 10^{-3}$

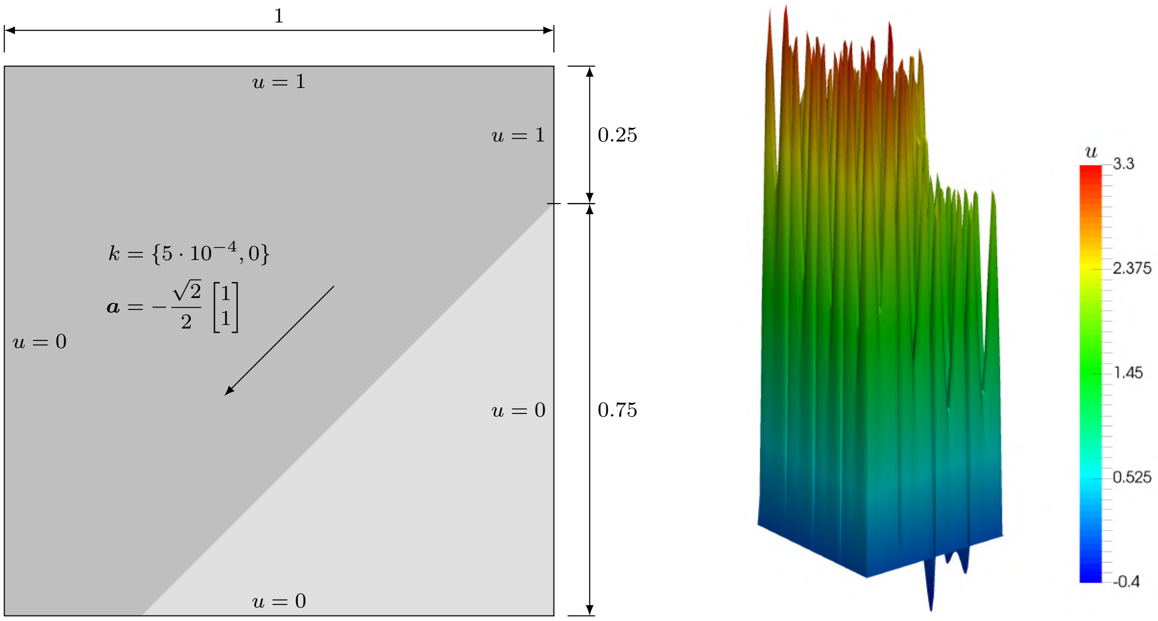
Figure 3.2: Convergence of Finite Element solutions to the one-dimensional advection-diffusion problem [Zander, 2016].

3.1.2 Example 2: Advection-diffusion skew to the mesh

Now, the two-dimensional example of an advection skew to the mesh is considered, following Brooks and Hughes [1982]. Figure 3.3a shows the setup of the problem, featuring a discontinuous inflow Dirichlet boundary condition. Similar to the previous example, boundary layers form on the left and the bottom edge. In addition, an internal layer extends from the discontinuity in the boundary data into the domain. Figure 3.3b shows that an insufficient discretization leads to a completely oscillatory solution. Again, the multi-level *hp*-scheme is used to refine the discretization. The adaptivity is driven by the gradient of the solution, *i.e.* an element is refined if the maximum value of the solution gradient in the element exceeds a given threshold. In the same way, an element is coarsened if the maximum gradient does not exceed this threshold value. This process is iterated until a maximum number of refinement levels is reached, or the newly computed solution does not cause a change in the discretization. Figure 3.4 displays the result for each of a maximum of 6 refinement steps. It can be seen that after several initial steps of uniform refinement, the sharp layers are automatically identified and in the following steps elements on the rest of the domain are coarsened again. This also illustrates the interpretation of the poor behavior of the Bubnov-Galerkin method as an insufficient resolution of boundary layers.

3.2 The Variational Multiscale method

The first numerical methods that successfully stabilize advection dominated problems were developed in a Finite Difference context. It was found that changing the second order accurate central difference stencil to a first order accurate upwind scheme makes solving problems up to



(a) Problem setup

(b) Numerical approximation using 4×4 elements of order $p = 4$, $\kappa = 5 \cdot 10^{-4}$, no stabilization

Figure 3.3: Advection-diffusion skew to the mesh benchmark following [Brooks and Hughes, 1982; Zander, 2016].

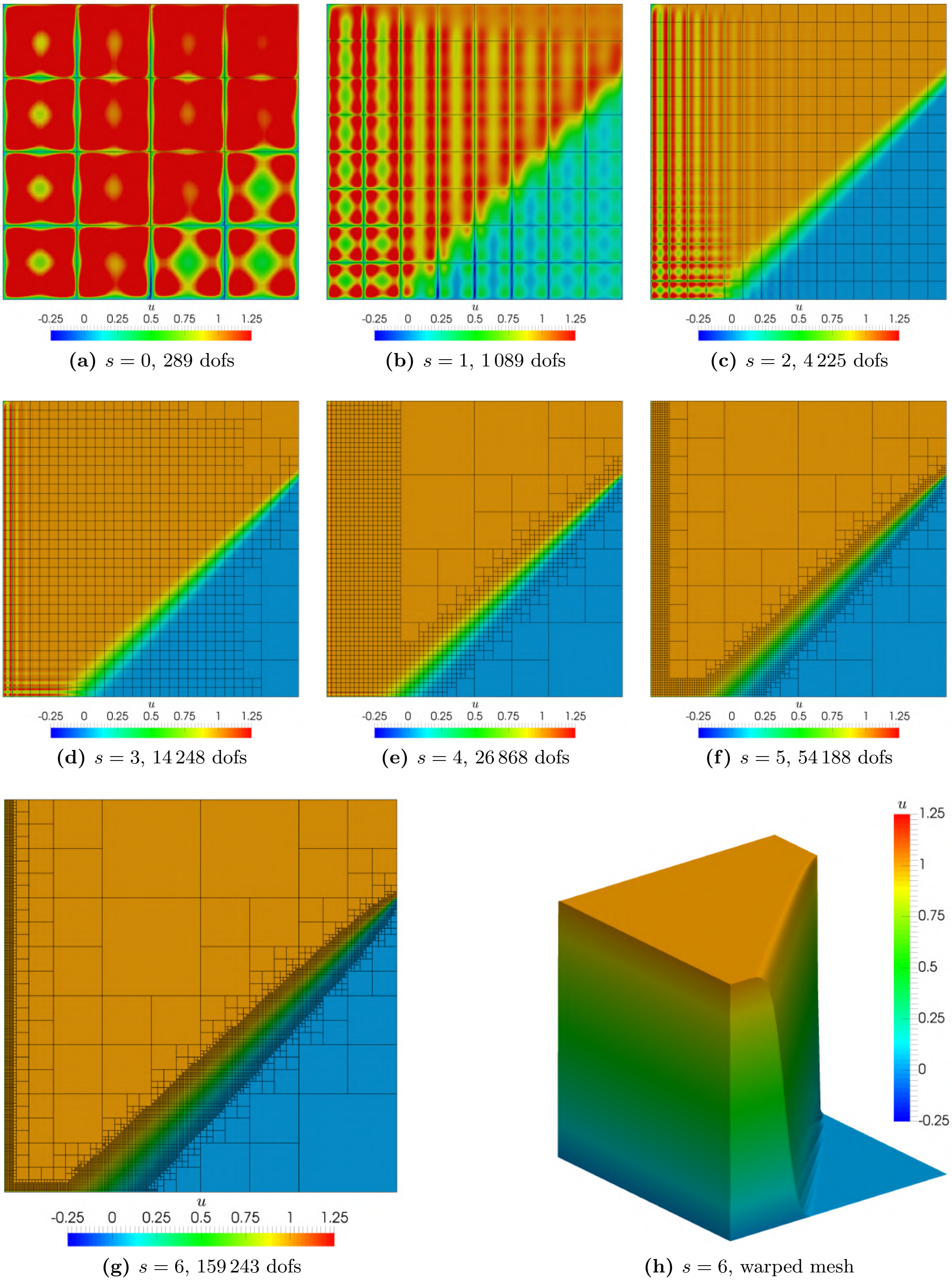
$Pe^h \to \infty$ possible [Zienkiewicz et al., 2005]. Attempts to interpret this approach as weighting the upstream contribution more than the downstream contribution resulted in the Streamline-Upwind Petrov-Galerkin (SUPG) method. However, until the introduction of the Variational Multiscale (VMS) method in Hughes [1995] and Hughes et al. [1998], a framework to explain the problem and to develop methods was missing. While first being designed in the context of stabilizing non-symmetric operators, it was discovered later that the same methodology applied to the Navier-Stokes equations also adds an effect known from turbulence models. As such, it accounts for the energy loss by small solution characteristics that are not captured by the discretization. Despite being closely related, there is a distinction between the VMS method as stabilization scheme and as turbulence model. The latter is commonly referred to as the residual-based VMS method.

Within the VMS, solution and weighting spaces are decomposed into a resolved and an unresolved part. Following the notation used in chapter 2, the split reads

$$\mathcal{S} = \overline{\mathcal{S}} \oplus \mathcal{S}'$$

$$\mathcal{W} = \overline{\mathcal{W}} \oplus \mathcal{W}'.$$

Figure 3.5 illustrates this concept schematically. The resolved scales are to be thought of as the part that can be captured by the Finite Element discretization. The small scale contribution is the missing part such that in a direct sum the original space is recovered. Therefore, the space of small scales is necessarily infinite dimensional. To make this split precise, projectors need to be introduced that uniquely define the transformation from the



(a) $s = 0$, 289 dofs

(b) $s = 1$, 1 089 dofs

(c) $s = 2$, 4 225 dofs

(d) $s = 3$, 14 248 dofs

(e) $s = 4$, 26 868 dofs

(f) $s = 5$, 54 188 dofs

(g) $s = 6$, 159 243 dofs

(h) $s = 6$, warped mesh

Figure 3.4: Numerical solution to the skew-mesh benchmark using multi-level *hp*-refinement [Zander, 2016]

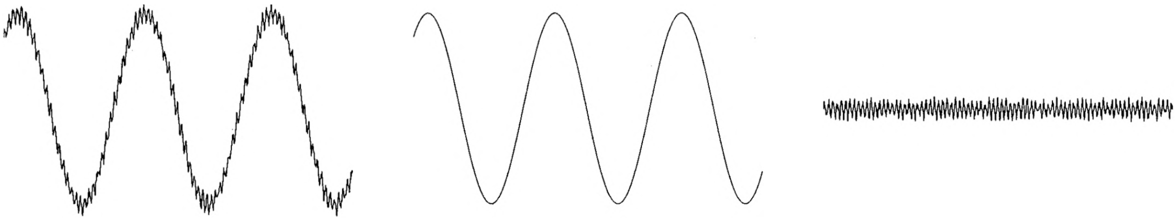

(a) Solution u (b) Coarse scale component $\overline{u}$ (c) Fine scale component u'

Figure 3.5: Illustration of coarse and fine scale contributions [Hughes et al., 2000].

complete space to the discrete space.

With this, the decompositions $u = \overline{u} + u'$ and $w = \overline{w} + w'$ can be inserted into the weak form (eq. 2.4), giving

$$\langle \overline{w} + w', \mathcal{L}(\overline{u} + u') \rangle = \langle \overline{w} + w', f \rangle \qquad \forall \overline{w} \in \overline{\mathcal{W}} \text{ and } \forall w' \in \mathcal{W}'. \tag{3.6}$$

As $\overline{\mathcal{W}}$ and $\mathcal{W}'$ are linear independent, eq. (3.6) can be split into a coarse and a fine scale problem:

$$\begin{aligned} \langle \overline{w}, \mathcal{L}(\overline{u} + u') \rangle &= \langle \overline{w}, f \rangle \qquad && \forall \overline{w} \in \overline{\mathcal{W}} \\ \langle w', \mathcal{L}(\overline{u} + u') \rangle &= \langle w', f \rangle \qquad && \forall w' \in \mathcal{W}'. \end{aligned}$$

The inner product on the left-hand side can then be separated in its second argument as follows

$$\begin{aligned} \langle \overline{w}, \mathcal{L}\overline{u}) + \langle \overline{w}, \mathcal{L}u' \rangle &= \langle \overline{w}, f \rangle \qquad && \forall \overline{w} \in \overline{\mathcal{W}} \\ \langle w', \mathcal{L}\overline{u} \rangle + \langle w', \mathcal{L}u' \rangle &= \langle w', f \rangle \qquad && \forall w' \in \mathcal{W}'. \end{aligned}$$

Note, that the last step requires the linearity of $\mathcal{L}$.

At this point, it would be desirable to solve the coarse scale equation directly. However, the second term includes the small scale solution, rendering the problem not closed. Solving the infinite dimensional small scale problem, in turn, includes the resolved scales and would result in solving the original problem. The conventional Bubnov-Galerkin method would just not consider the $\langle \overline{w}, \mathcal{L}u' \rangle$ term and solve the remaining equation. While this is reasonable in the symmetric case, the intention of the Variational Multiscale Method is to find a unified approach to stabilize the problem in the general case.

The main idea is to model the small scale problem and substitute the result into the coarse scale equation. To this end, the derivatives in $\langle \overline{w}, \mathcal{L}u' \rangle$ are transferred to the weighting functions. The new operator that is now acting on $\overline{w}$ is the adjoint of $\mathcal{L}$, denoted as $\mathcal{L}^*$. This ensures that no derivatives act on u'. Furthermore, the terms in the small scale equation are rearranged as follows:

$$\begin{aligned} \langle \overline{w}, \mathcal{L}\overline{u} \rangle + \langle \mathcal{L}^*\overline{w}, u' \rangle &= \langle \overline{w}, f \rangle \qquad && \forall \overline{w} \in \overline{\mathcal{W}} \\ \langle w', \mathcal{L}u' \rangle &= -\langle w', \mathcal{L}\overline{u} - f \rangle \qquad && \forall w' \in \mathcal{W}'. \end{aligned}$$

This last step makes evident that the small scales are driven by the residual of the resolved scales. The small scale problem is formally solved by

$$u' = -\mathcal{G}'\big(\mathcal{L}\overline{u} - f\big), \tag{3.7}$$

where $\mathcal{G}'$ denotes the small scale Green's operator that is related to the inverse of $\mathcal{L}$ but takes the defined projectors and boundary conditions into account. The mathematical details can be found e.g. in Hughes and Sangalli [2007]. With this, the coarse problem can be closed:

$$\langle \overline{w}, \mathcal{L}\overline{u}\rangle - \langle \mathcal{L}^*\overline{w}, \mathcal{G}'\big(\mathcal{L}\overline{u} - f\big)\rangle = \langle \overline{w}, f\rangle \qquad \forall \overline{w} \in \overline{\mathcal{W}}.$$

The remaining task is to find a good approximation to $\mathcal{G}'$. While extremely simple, u' is commonly approximated as

$$u' \approx \tau\big(\mathcal{L}\overline{u} - f\big), \tag{3.8}$$

where τ denotes a stabilization parameter. With this, the stabilized weak form reads:

$$\langle \overline{w}, \mathcal{L}\overline{u}\rangle - \langle \mathcal{L}^*\overline{w}, \tau\big(\mathcal{L}\overline{u} - f\big)\rangle = \langle \overline{w}, f\rangle \qquad \forall \overline{w} \in \overline{\mathcal{W}}. \tag{3.9}$$

Section 3.3 will address the selection of τ in more detail. The method provides a starting point for the development of more advanced schemes and is therefore not limited to this simplification.

Furthermore, eq. (3.9) coincides with the general structure of stabilization techniques [Donéa and Huerta, 2003]. That is, stabilization techniques can be grouped under the following form:

$$\langle \overline{w}, \mathcal{L}\overline{u}\rangle + \langle \boldsymbol{\mathcal{S}}\overline{w}, \tau\big(\mathcal{L}\overline{u} - f\big)\rangle_{\Omega^e} = \langle \overline{w}, f\rangle \qquad \forall \overline{w} \in \overline{\mathcal{W}},$$

where $\boldsymbol{\mathcal{S}}$ denotes a stabilization operator. It is defined for different methods as

$$\begin{aligned} \text{VMS}: &\qquad \boldsymbol{\mathcal{S}} = -\mathcal{L}^* \\ \text{SUPG}: &\qquad \boldsymbol{\mathcal{S}} = \mathcal{L}_{adv} \\ \text{GLS}: &\qquad \boldsymbol{\mathcal{S}} = \mathcal{L}, \end{aligned}$$

where $\mathcal{L}_{adv}$ denotes the advective part of $\mathcal{L}$. A detailed summary of the Variational Multiscale method can be found in Hughes et al. [2004].

3.2.1 Example 3: Advection-diffusion skew to the mesh

To show the performance of the multi-level *hp*-method combined with a stabilization scheme, the setting of section 3.1.2 is considered with zero diffusivity. In this case, no discrete Bubnov-Galerkin method would deliver meaningful results.

To stabilize the Finite Element solution, an SUPG-scheme is used. The weak form (*cf.* eq. 3.5) is therefore extended by adding the residual weighted with $\mathcal{L}_{adv}w = \mathbf{a}\cdot\nabla w$:

$$B^{cd}(u,w) + \sum_e \big\langle \mathbf{a}\cdot\nabla w, \tau\big[\mathbf{a}\cdot\nabla u - f\big]\big\rangle_{\Omega^e} = F^{cd}(u,w) \qquad \forall w \in \mathcal{W}. \tag{3.10}$$

The stabilization parameter τ is selected as

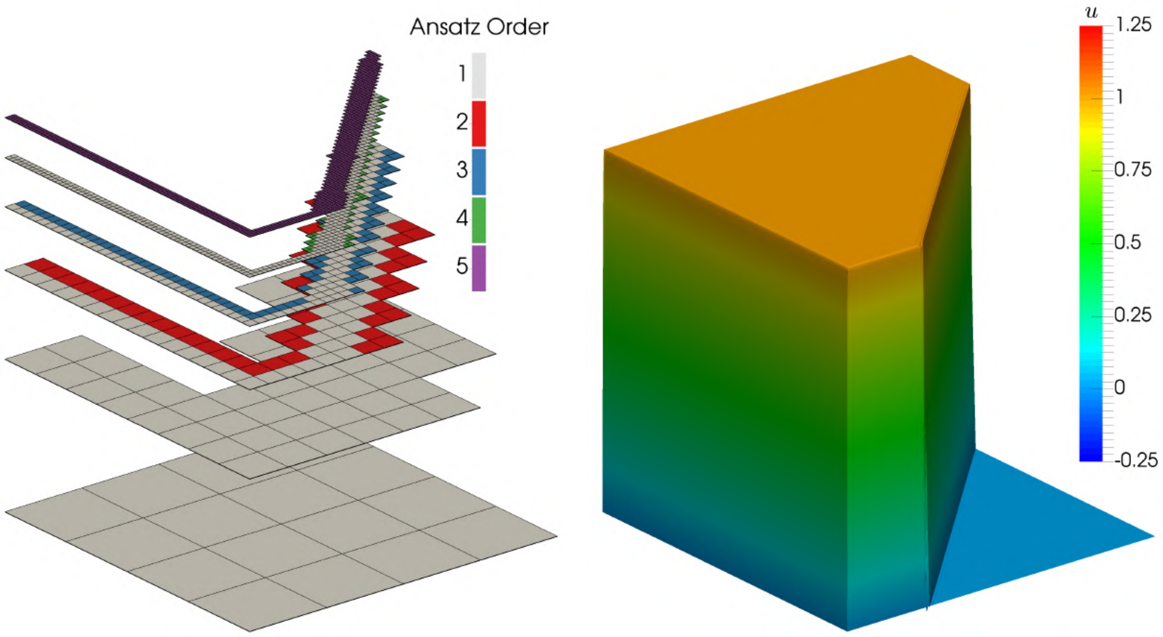
$$\tau = \frac{\tau^h}{\sqrt{p}}, \tag{3.11}$$

where

$$\tau^h = \frac{h}{\sqrt{2}\,\|\mathbf{a}\|}. \tag{3.12}$$

Equation (3.12) extends a definition of τ for one-dimensional advection-diffusion equations to multiple dimensions, that is known to deliver nodally exact values for $p = 1$ [Donéa and Huerta, 2003]. This value is then adapted to higher polynomial degrees in eq. (3.11). The choice to divide by $\sqrt{p}$ is founded on empirical (visual) considerations. A discussion about selecting τ follows in section 3.3.

Figure 3.6 shows the result of a graded multi-level *hp*-discretization applied to the SUPG-stabilized problem. Clearly, the discontinuous boundary and internal layers are captured by the gradient driven refinement (as in section 3.1.2).



(a) Refined discretization, 41 489 dofs

(b) Warped solution on refined discretization

Figure 3.6: Stabilized Finite Element solution to the skew-mesh benchmark using five levels of multi-level *hp*-refinement [Zander, 2016].

3.3 Choice of the stabilization parameter

The main difficulty in the development of stabilization schemes for advection-dominated problem has been the selection of the stabilization parameter τ. One question that has to be addressed in advance is the definition of the quality of an approximation. In the Finite

Element framework, the error is measured using a norm that maps functions from the solution space to a real number. This way, the quality of different approximations can be compared by looking at the norm of the error $\|e\| = \|u - u^h\|$. Now, stabilization techniques such as the Variational Multiscale Method can be tuned to deliver good results with respect to the chosen norm. It might be natural to optimize the difference of solution and approximation, and therefore choose the L^2 norm. However, L^2-optimal approximations of potentially very steep or even discontinuous solutions arising from advection-dominant problems are characterized by Gibbs' phenomenon. That is, especially in a p-FEM context, unphysical oscillations around the discontinuity limit the applicability of such a method in practice. To suppress this phenomenon, the weak form is commonly extended by a shock-capturing term that adds diffusion depending on the steepness of the solution. This renders the problem nonlinear, increasing its complexity significantly.

The standard procedure is, therefore, to seek optimality in common norms such as H^1 or L^2 and apply a shock capturing scheme if oscillations are too severe. For a one-dimensional advection-diffusion equation, the definition

$$\tau_h = \frac{h}{2a}\left(\coth Pe - \frac{1}{Pe}\right) \tag{3.13}$$

was found to give nodally exact values for linear elements and represents an optimal solution in the H^1 semi-norm. This corresponds to the behavior of Bubnov-Galerkin discretizations applied to symmetric problems. Although it does not transfer to multiple dimensions, eq. (3.13) is regularly used.

In the context of this thesis, it is crucial to find a definition for τ that also includes the polynomial degree p. Equation (3.13) leads to an over-diffusive behavior when p is increased, as it does not account for the additional approximation power. Moreover, convergence cannot be guaranteed, as the stabilization term needs to possess the same magnitude as the weak residual. For low-order elements, the optimal scaling includes a factor h in the definition of τ. This can be explained looking at the advective limit ($k \to 0$). Then, $\tau = h/2a$ rescales the weighting functions $L_{adv}(N_i) = a \cdot dN_i/dx$ to be precisely of magnitude one.

To adapt this for p-refinement, various possibilities exist. Some definitions have been studied in this work and are discussed in the following. Figure 3.7 provides a comparison of L^2 errors on a benchmark similar to example 3.1.1 with $a = 0.5$ and $k = 0.001$. The boundary conditions were set to $u(0) = 0$ and $u(1) = 1$ and emphasize the influence of the boundary layer. Moreover, an optimal τ was found by iteratively testing the error for a range of values (from 0 to τ_h) and refining around the minimum until the increment $|\tau_p^{n+1} - \tau_p^n|$ reaches machine precision.

The first choice is to select

$$\tau_p = \frac{\tau_h}{p}, \tag{3.14}$$

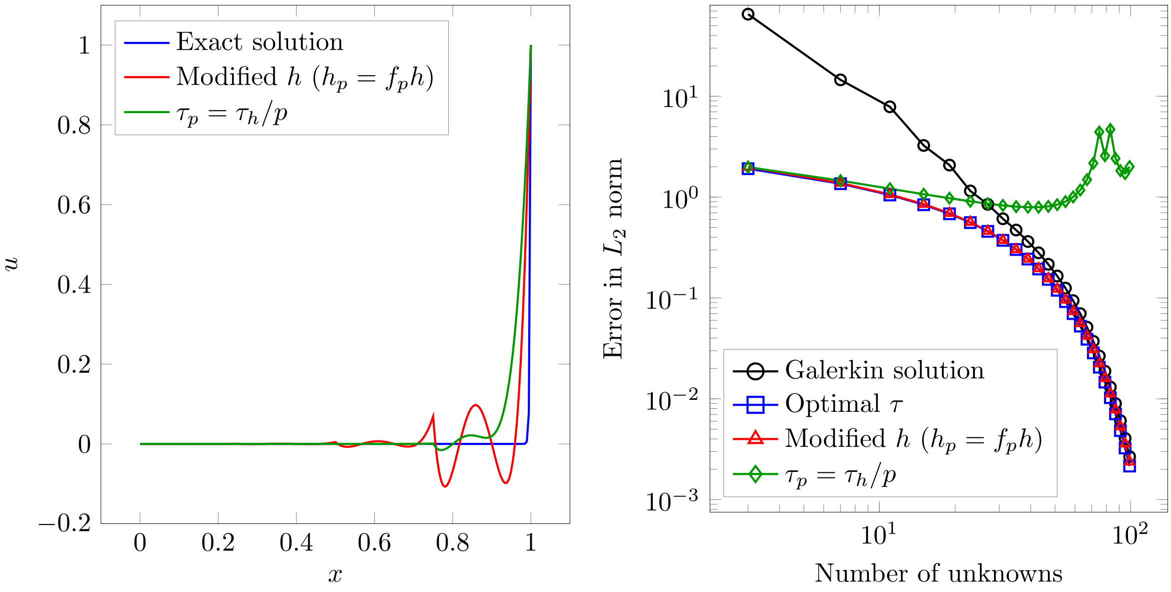
as suggested e.g. in Donéa and Huerta [2003]. Although an improvement can be achieved for low polynomial degrees, it is apparent that the scaling is not correct for $p \to \infty$. A different approach is to define a p-adapted element length h_p that accounts for the approximation quality of the high-order polynomials. This results in new definitions for Pe and τ:

$$Pe_p = Pe(h_p) \tag{3.15}$$
$$\tau_p = \tau_h(h_p, Pe_p) \tag{3.16}$$

A common choice in combination with Lagrange basis functions is to consider the distance between interpolation points, *i.e.* using $h_p = h/p$ for an equidistant spacing. The results obtained with this definition are very similar to the previous formula (eq. 3.14). A better scaling was obtained with

$$\begin{aligned} h_p &= f_p h \\ f_p &= \frac{3}{p^2+2p}. \end{aligned} \tag{3.17}$$

Definition (3.17) is based on fitting a curve to the (numerically computed) optimal τ_p. It confirms the results shown in fig. 3.1b that for high polynomial degrees p-refinement is more efficient than h-refinement. Although this definition performs well in the presented one-dimensional benchmark, it has not yet been investigated, how generically applicable it is. Moreover, the question of optimality arises, as the oscillations are clearly more significant compared to using eq. (3.14).



(a) Solution for $p = 4$

(b) L^2-Convergence

Figure 3.7: Comparison of different definitions for τ_p.

Besides eq. (3.13), many other possibilities exist to define τ. Promising concerning high-order polynomials are in particular the approaches introduced by Franca et al. [1992], where τ is calculated based on a modified Péclet number, and Tezduyar and Osawa [2000], where the element matrices are used to define the stabilization parameter.

The extension to include a reaction term is particularly important in transient advection-diffusion problems as it appears as a consequence of the time discretization. The significance of this reaction term is then characterized by the time step Δt. In this setting, the stabilization term can be adapted using formulations as introduced in e.g. Codina [2000] by analyzing the discrete system:

$$\tau = \left(\sigma + \frac{2a}{h} + \frac{4k}{h^2}\right)^{-1},$$

where σ denotes the reaction constant. A fourth-order accurate definition can be found in Shakib et al. [1991]:

$$\tau = \left(\sigma^2 + \left(\frac{2a}{h}\right)^2 + 9\left(\frac{4k}{h^2}\right)^2\right)^{-\frac{1}{2}}. \tag{3.18}$$

Both attempt to blend the advective, diffusive and reactive limits of τ. The multi-dimensional version of eq. (3.18) reads

$$\tau = \left(\sigma^2 + \boldsymbol{a}^T \boldsymbol{G} \boldsymbol{a} + 9k^2 \boldsymbol{G} : \boldsymbol{G}\right)^{-\frac{1}{2}}, \tag{3.19}$$

with the second order metric tensor $\boldsymbol{G}$ defined as

$$G_{ij} = \sum_k \frac{\partial \xi_k}{\partial x_i} \frac{\partial \xi_k}{\partial x_j}. \tag{3.20}$$

It is derived by transforming the advection-diffusion equation into the element coordinate system. This definition is commonly used in the Variational Multiscale setting, see, e.g. Bazilevs et al. [2010]. Equation (3.19) can be used to stabilize a transient advection-diffusion equation:

$$\tau = \left(\frac{4}{\Delta t^2} + \boldsymbol{a}^T \boldsymbol{G} \boldsymbol{a} + 9k^2 \boldsymbol{G} : \boldsymbol{G}\right)^{-\frac{1}{2}}.$$

To adapt this formula to high-order discretizations, the derivative of the mapping from global to local in eq. (3.20) is identified as a measure for h^{-1} and is therefore multiplied with f_p^{-1}, according to eq. (3.15) - (3.17):

$$G_{ij}^p = \frac{G_{ij}}{f_p^2}. \tag{3.21}$$

Last, but not least it is not clear how good a scalar can approximate $\mathcal{G}'$ (*cf.* eq. 3.8). It might be necessary to choose a more sophisticated approximation to exploit the full potential of high-order polynomials.

Chapter 4

Residual-based Variational Multiscale turbulence modeling using FCM

In this chapter, the Variational Multiscale method is applied to the Navier-Stokes equations, following section 3.2. The weak form is then discretized using the multi-level *hp*-adaptive Finite Cell method.

The starting point is the strong form of the incompressible Navier-Stokes equations:

$$\begin{aligned}
\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \nu \Delta \boldsymbol{u} + \boldsymbol{f} \qquad & \text{on } \Omega \\
\nabla \cdot \boldsymbol{u} = 0 \qquad & \text{on } \Omega \\
\boldsymbol{u} = \boldsymbol{g} \qquad & \text{on } \Gamma_D \\
\nu \nabla \boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{0} \qquad & \text{on } \Gamma_N \\
\boldsymbol{u} = \boldsymbol{u}_0 \qquad & \text{at } t = 0 \qquad .
\end{aligned}$$

After defining test and trial functions as

$$\begin{aligned}
\boldsymbol{U} = \{\boldsymbol{u}, p\} \qquad & \boldsymbol{u} \in \boldsymbol{\mathcal{S}}, \quad p \in \mathcal{P} \\
\boldsymbol{W} = \{\boldsymbol{w}, q\} \qquad & \boldsymbol{w} \in \boldsymbol{\mathcal{W}}, q \in \mathcal{Q},
\end{aligned}$$

the weak form becomes

$$B(\boldsymbol{W}, \boldsymbol{U}) = L(\boldsymbol{W}) \qquad \forall \boldsymbol{W} \in \{\boldsymbol{\mathcal{W}} \times \mathcal{Q}\}. \tag{4.1}$$

The nonlinear form B is composed of a linear form B_1 and a trilinear form B_2 that represents the nonlinear part:

$$B(\boldsymbol{W}, \boldsymbol{U}) = B_1(\boldsymbol{W}, \boldsymbol{U}) + B_2(\boldsymbol{W}, \boldsymbol{U}, \boldsymbol{U}),$$

where

$$B_1(\boldsymbol{W},\boldsymbol{U}) = \left\langle \boldsymbol{w}, \frac{\partial \boldsymbol{u}}{\partial t} \right\rangle + \left\langle \nabla \cdot \boldsymbol{w}, p \right\rangle - \left\langle q, \nabla \cdot \boldsymbol{u} \right\rangle + \left\langle \nabla^S \boldsymbol{w}, \nu \nabla^S \boldsymbol{u} \right\rangle$$
$$B_2(\boldsymbol{W},\boldsymbol{V},\boldsymbol{U}) = -\left\langle \nabla \boldsymbol{w}, \boldsymbol{v} \otimes \boldsymbol{u} \right\rangle.$$

The new argument in B_2 is defined as $\boldsymbol{V} = \{\boldsymbol{v}, 0\}$, $\boldsymbol{v} \in \boldsymbol{\mathcal{W}}$. This step separates the nonlinear form B into a bilinear and a trilinear form and is used in the application of the Variational Multiscale method in section 4.1. Note also, that

$$\boldsymbol{\mathcal{S}} = \{\, \mathcal{S}_1, \, \dots \, , \mathcal{S}_d \,\}$$
$$\boldsymbol{\mathcal{W}} = \{\, \mathcal{W}_1, \, \dots \, , \mathcal{W}_d \,\},$$

where d denotes the number of space dimensions. As only homogeneous Neumann boundary conditions are considered here, the right hand side $L(\boldsymbol{W})$ contains only the volume forces and is therefore defined as

$$L(\boldsymbol{W}) = \langle \boldsymbol{W}, \boldsymbol{f} \rangle.$$

Now, the problem has a saddle-point nature, *i.e.* it can not be obtained from the minimization of an (energy-) functional. The consequence is, that in contrast to previous equations the weak form does not satisfy coercivity. Therefore, the Lax-Milgram lemma (*cf.* chapter 2) does not apply, and the uniqueness of the solution cannot be guaranteed in general. Discretizing eq. (4.1) with a standard Finite Element method also yields a zero block-matrix as no interaction between q and p is present. To prove invertibility in this setting, the weaker, nevertheless sufficient, Babuška-Brezzi or inf-sup condition is used which reads [Donéa and Huerta, 2003]:

$$\inf_{q \in \mathcal{Q}} \sup_{w \in \mathcal{V}} \frac{\langle q, \nabla \cdot \boldsymbol{w} \rangle}{\|\boldsymbol{w}\|_{\mathcal{W}} \|q\|} > 0,$$

or equivalently:

$$\exists \lambda \text{ s.t.} \qquad \sup_{w \in \mathcal{V}} \frac{\langle q, \nabla \cdot \boldsymbol{w} \rangle}{\|\boldsymbol{w}\|_{\mathcal{W}}} \geq \lambda \|q\| \qquad \forall q \in \mathcal{Q}. \tag{4.2}$$

Equation (4.2) formulates the condition in the form of an eigenvalue problem. It is, therefore, apparent that the inf-sup condition requires $\langle q, \nabla \cdot \boldsymbol{w} \rangle$ to have non-zero eigenvalues which guarantees that problem (4.1) has a unique solution. While this property can be verified in the continuous setting, it does not generally transfer to the discrete spaces. As a consequence, inf-sup stability needs to be proven for each velocity-pressure discretization pair. Otherwise, spurious pressure modes arise, rendering the solution unphysical. A detailed analysis of Galerkin methods and inf-sup conditions can be found in Durán [2005].

Instead of choosing inf-sup stable spaces, the problem can be stabilized. To this end, formulations like the Galerkin Least-Squares (GLS) or the Variational Multiscale method add a small amount of diffusion to the problem to suppress the spurious modes. In practice, this is very convenient, as arbitrary combinations of pressure and velocity discretizations can then be used.

4.1 The residual-based Variational Multiscale method

Following the formulation and notation used by Bazilevs et al. [2007a], velocity and pressure interpolation spaces, as well as momentum and continuity weighting spaces, are split into a resolved and an unresolved part:

$$\begin{aligned} \boldsymbol{\mathcal{S}} &= \overline{\boldsymbol{\mathcal{S}}} \oplus \boldsymbol{\mathcal{S}}' \qquad & \mathcal{P} &= \overline{\mathcal{P}} \oplus \mathcal{P}' \\ \boldsymbol{\mathcal{W}} &= \overline{\boldsymbol{\mathcal{W}}} \oplus \boldsymbol{\mathcal{W}}' \qquad & \mathcal{Q} &= \overline{\mathcal{Q}} \oplus \mathcal{Q}'. \end{aligned}$$

Then, the test and weighting function pairs become

$$\begin{aligned} \boldsymbol{U} &= \overline{\boldsymbol{U}} + \boldsymbol{U}' &&= \{\overline{\boldsymbol{u}}, \overline{p}\} + \{\boldsymbol{u}', p'\} \\ \boldsymbol{W} &= \overline{\boldsymbol{W}} + \boldsymbol{W}' &&= \{\overline{\boldsymbol{w}}, \overline{q}\} + \{\boldsymbol{w}', q'\}. \end{aligned}$$

In the following, the projector is chosen to separate the two scales with respect to the H^1 semi-norm. Inserting the split into eq. (4.1) yields

$$B(\overline{\boldsymbol{W}}, \overline{\boldsymbol{U}} + \boldsymbol{U}') = \boldsymbol{L}(\overline{\boldsymbol{W}}) \qquad \forall \overline{\boldsymbol{W}} \in \overline{\boldsymbol{\mathcal{W}}} \tag{4.3a}$$

$$B(\boldsymbol{W}', \overline{\boldsymbol{U}} + \boldsymbol{U}') = \boldsymbol{L}(\boldsymbol{W}') \qquad \forall \boldsymbol{W}' \in \boldsymbol{\mathcal{W}}', \tag{4.3b}$$

where the goal is to close eq. (4.3a) using eq. (4.3b). Now, the fine scale weak form (eq. 4.3b) is split as follows

$$\begin{aligned} B_1(\boldsymbol{W}', \overline{\boldsymbol{U}}) + B_1(\boldsymbol{W}', \boldsymbol{U}') + B_2(\boldsymbol{W}', \overline{\boldsymbol{U}}, \overline{\boldsymbol{U}}) + B_2(\boldsymbol{W}', \overline{\boldsymbol{U}}, \boldsymbol{U}') \\ + B_2(\boldsymbol{W}', \boldsymbol{U}', \overline{\boldsymbol{U}}) + B_2(\boldsymbol{W}', \boldsymbol{U}', \boldsymbol{U}'). = \boldsymbol{L}(\boldsymbol{W}'). \end{aligned} \tag{4.4}$$

Defining the strong residual of the Navier-Stokes equations as $\boldsymbol{r}(\boldsymbol{U})$ allows to compress the terms $B_1(\boldsymbol{W}', \overline{\boldsymbol{U}})$, $B_2(\boldsymbol{W}', \overline{\boldsymbol{U}}, \overline{\boldsymbol{U}})$ and $\boldsymbol{L}(\boldsymbol{W}')$ into $\langle \boldsymbol{W}', \boldsymbol{r}(\overline{\boldsymbol{U}}) \rangle$. Rearranging the remaining three terms yields then:

$$B(\boldsymbol{W}', \boldsymbol{U}') = -\langle \boldsymbol{W}', \boldsymbol{r}(\overline{\boldsymbol{U}}) \rangle - B_2(\boldsymbol{W}', \overline{\boldsymbol{U}}, \boldsymbol{U}') - B_2(\boldsymbol{W}', \boldsymbol{U}', \overline{\boldsymbol{U}}), \tag{4.5}$$

which represents the small scale equation that is used to determine (approximate) the small scales. The formal inverse operator, that by definition solves eq. (4.5), depends now not only on the coarse scale residual but also on the solution:

$$\boldsymbol{U}' = -\boldsymbol{\mathcal{F}}'(\overline{\boldsymbol{U}}, \boldsymbol{r}(\overline{\boldsymbol{U}}))$$

This differs from the linear case where the fine scale Green's operator is only applied to the residual (*cf.* eq. 3.7).

In the same fashion as in the linear case, $\boldsymbol{\mathcal{F}}'$ is now approximated by a matrix of scalars:

$$\boldsymbol{U}' \approx -\boldsymbol{\tau}\, \boldsymbol{r}(\overline{\boldsymbol{U}}). \tag{4.6}$$

While this is similar to eq. (3.8), the solution and the residual are vector quantities, therefore

$\boldsymbol{\tau}$ is also a matrix. It is defined as follows [Bazilevs et al., 2007a]:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_M \boldsymbol{I}^{d\times d} & \mathbf{0} \\ \mathbf{0} & \tau_C \end{bmatrix},$$

where τ_M denotes the stabilization parameter of the momentum residual and τ_C the stabilization parameter for the continuity residual. The former is chosen according to eq. (3.19), that was obtained from the analysis of advection-diffusion systems. The definition for τ_C reads

$$\tau_C = (\boldsymbol{g} \cdot \tau_M \boldsymbol{g})^{-1},$$

with

$$g_i = \sum_{j=1}^{d} \frac{\partial \xi_j}{\partial x_i}.$$

Following section 3.3, an adaption to higher polynomial degrees is achieved by defining $g_i^p = g_i / f_p$, similar to eq. (3.21).

Now, eq. (4.3a) can be closed by inserting approximation (4.6). As this step introduces a modeling error, the exact coarse scale solution $\overline{\boldsymbol{U}}$ is replaced by the approximated solution $\boldsymbol{U}^h$. Inserting eq. (4.6) into eq. (4.3a) yields the Variational Multiscale weak form

$$B^{MS}(\boldsymbol{W}^h, \boldsymbol{U}^h) = \boldsymbol{L}(\boldsymbol{W}^h) \qquad \forall \boldsymbol{W}^h \in \{\overline{\mathcal{W}} \times \overline{\mathcal{Q}}\}, \tag{4.7}$$

with

$$\begin{aligned} B^{MS}(\boldsymbol{W}^h, \boldsymbol{U}^h) = & B(\boldsymbol{W}^h, \boldsymbol{U}^h) - B_1(\boldsymbol{W}^h, \boldsymbol{\tau}\, \boldsymbol{r}[\boldsymbol{U}^h]) \\ & - B_2(\boldsymbol{W}^h, \boldsymbol{U}^h, \boldsymbol{\tau}\, \boldsymbol{r}[\boldsymbol{U}^h]) - B_2(\boldsymbol{W}^h, \boldsymbol{\tau}\, \boldsymbol{r}[\boldsymbol{U}^h], \boldsymbol{U}^h) \\ & + B_2(\boldsymbol{W}^h, \boldsymbol{\tau}\, \boldsymbol{r}[\boldsymbol{U}^h], \boldsymbol{\tau}\, \boldsymbol{r}[\boldsymbol{U}^h]). \end{aligned}$$

To conclude the Variational Multiscale formulation, three assumptions are made at this point (*cf.* Bazilevs et al. [2007a]):

$$\begin{aligned} &(1) \qquad \left\langle \boldsymbol{w}^h, \frac{\partial \boldsymbol{u}'}{\partial t} \right\rangle = 0 \\ &(2) \qquad \boldsymbol{u}' = 0 \qquad \text{on } \Gamma \\ &(3) \qquad \langle \nabla^S \boldsymbol{w}^h, \nabla^S \boldsymbol{u}' \rangle = 0. \end{aligned}$$

Although assumption (1) has been shown to have a significant influence on the solution [Codina et al., 2007] it can be justified by the simplification that follows from it. Moreover, as a consequence of choosing an H^1-projector, coarse and small scales are orthogonal, *i.e.*, $\langle \nabla^S \overline{\boldsymbol{w}}, \nabla^S \boldsymbol{u}' \rangle = 0$ holds (*cf.* appendix A). Of course, the approximation $\boldsymbol{u}^h$ is not an exact H^1-projection, however, assumption (3) seems reasonable in this context. Now, applying assumptions (1) - (3), B_1 and B_2 can be expanded:

$$\begin{aligned} B^{MS}(\boldsymbol{W}^h, \boldsymbol{U}^h) = \; & B(\boldsymbol{W}^h, \boldsymbol{U}^h) + \langle \nabla q^h, \boldsymbol{\tau}_M \boldsymbol{r}_M \rangle + \langle \nabla \cdot \boldsymbol{w}^h, \tau_C r_C \rangle \\ & + \langle \nabla \boldsymbol{w}^h, \boldsymbol{u}^h \otimes \boldsymbol{\tau}_M \boldsymbol{r}_M + \boldsymbol{\tau}_M \boldsymbol{r}_M \otimes \boldsymbol{u}^h \rangle \\ & - \big\langle \nabla \boldsymbol{w}^h, \boldsymbol{\tau}_M \boldsymbol{r}_M \otimes \boldsymbol{\tau}_M \boldsymbol{r}_M \big\rangle . \end{aligned}$$

The explicit dependencies of $\boldsymbol{r}_M$ and r_C on $\boldsymbol{u}^h$ and p^h have been omitted for readability. The first two terms originate from B_1 and the remaining terms from B_2. Using the equalities

$$\begin{aligned} \langle \nabla \boldsymbol{w}^h, \boldsymbol{u}^h \otimes \boldsymbol{\tau}^M \boldsymbol{r}_M \rangle &= \langle \boldsymbol{u}^h \cdot \nabla \boldsymbol{w}^h, \boldsymbol{\tau}_M \boldsymbol{r}_M \rangle \\ \langle \nabla \boldsymbol{w}^h, \boldsymbol{\tau}^M \boldsymbol{r}_M \otimes \boldsymbol{u}^h \rangle &= \langle \boldsymbol{u}^h \cdot (\nabla \boldsymbol{w}^h)^T, \boldsymbol{\tau}^M \boldsymbol{r}_M \rangle \end{aligned}$$

allows to further rearrange the additional terms and arrive at:

$$B^{MS}(\boldsymbol{W}^h, \boldsymbol{U}^h) = B(\boldsymbol{W}^h, \boldsymbol{U}^h) + \langle \boldsymbol{u}^h \cdot \nabla \boldsymbol{w}^h + \nabla q^h, \boldsymbol{\tau}_M \boldsymbol{r}_M \rangle + \langle \nabla \cdot \boldsymbol{w}^h, \tau_C r_C \rangle \tag{4.8}$$

$$+ \big\langle \boldsymbol{u}^h \cdot (\nabla \boldsymbol{w}^h)^T, \boldsymbol{\tau}_M \boldsymbol{r}_M \big\rangle - \big\langle \nabla \boldsymbol{w}^h, \boldsymbol{\tau}_M \boldsymbol{r}_M \otimes \boldsymbol{\tau}_M \boldsymbol{r}_M \big\rangle . \tag{4.9}$$

It is interesting to notice that $B_1(\boldsymbol{W}^h, \boldsymbol{\tau}\ \boldsymbol{r}[\boldsymbol{U}^h])$ introduces a coupling between q^h and p^h. Consequently, the zero block-matrix in the original saddle point problem is filled with a diffusion term that is scaled with τ_C. Therefore, the Variational Multiscale method stabilizes the weak form for velocity-pressure discretization pairs that do not satisfy the inf-sup condition.

4.2 Weak imposition of boundary conditions

The fact that boundaries are not conforming to the mesh is crucial in the Finite Cell context. As a consequence, a weak imposition needs to be considered (*cf.* section 2.4). Moreover, it was shown by Bazilevs and Hughes [2007] that this has several benefits compared to applying boundary conditions in a strong way. First, the instabilities of the discrete operator that occur at boundary layers when convection is dominant can be stabilized by relaxing the constraint appropriately. That is, in the transition between pure diffusion and pure convection the outflow Dirichlet condition is weakened more and more. In the limit of zero diffusion, no outflow condition is imposed as the system becomes hyperbolic and information can only travel downstream. Second, in addition to the stability argument, it has been shown in Bazilevs et al. [2007b] that often inaccurate tangential velocity components near turbulent boundary layers can be greatly improved. To this end, a wall-function model is incorporated to account for the lack of approximability in the vicinity of no-slip boundaries. Comparisons to experiments, direct numerical simulations (DNS) and other turbulence models can be found in Bazilevs et al. [2010].

To enforce Dirichlet boundary conditions weakly using Nitsche's method (*cf.* section 2.4), the formulation presented in Bazilevs et al. [2010] extends the Variational Multiscale weak form as follows:

$$
\begin{aligned}
B^{MSW}(\boldsymbol{W}^h, \boldsymbol{U}^h) = B^{MS}(\boldsymbol{W}^h, \boldsymbol{U}^h) \\
&+ \langle \boldsymbol{w}^h, (\boldsymbol{u}^h \cdot \boldsymbol{n})\boldsymbol{u}^h + p^h\boldsymbol{n} - 2\nu\nabla^s\boldsymbol{u}^h \cdot \boldsymbol{n} \rangle_{\Gamma_D} \\
&+ \langle -2\nu\nabla^s\boldsymbol{w}^h \cdot \boldsymbol{n} - q^h\boldsymbol{n}, \boldsymbol{u}^h - \boldsymbol{g} \rangle_{\Gamma_{D,out}} \\
&+ \langle -2\nu\nabla^s\boldsymbol{w}^h \cdot \boldsymbol{n} - q^h\boldsymbol{n} - (\boldsymbol{u}^h \cdot \boldsymbol{n})\boldsymbol{w}^h, \boldsymbol{u}^h - \boldsymbol{g} \rangle_{\Gamma_{D,in}} \\
&+ \langle \boldsymbol{w}^h \beta_W, \boldsymbol{u}^h - \boldsymbol{g} \rangle_{\Gamma_D} \\
&+ \left\langle \boldsymbol{w}^h \cdot \boldsymbol{n}\left(\beta - \beta_W\right), \boldsymbol{u}^h \cdot \boldsymbol{n} - \boldsymbol{g} \right\rangle_{\Gamma_D},
\end{aligned}
\tag{4.10}
$$

where

$$\beta = \frac{C_B^I}{h_b} \tag{4.11}$$

and

$$\beta_W = \frac{u^{*2}}{\|\boldsymbol{u}^h_{tan}\|}. \tag{4.12}$$

The first (boundary-) term in eq. (4.10) recovers consistency of the method by canceling the Dirichlet part of the boundary integral as described in section 2.4. The two following terms add adjoint-consistency (*cf.* section 2.4). To impose only inflow boundary conditions in the advection dominant case the outflow boundary integral contains only the diffusive contribution and therefore vanishes for $\nu \to 0$. The last two terms penalize the deviation from the prescribed values and ensure the stability of the method. Again the term scales with k and vanishes in the advective limit. In that case, the inflow boundary condition is only enforced by the contribution of the adjoint consistency term. Furthermore, the tangential (no-slip) part of the penalty term is treated slightly different than the wall-normal (no-penetration) part. Instead of using the the original parameter β the tangential contribution is enforced by a wall-penalty β_W instead. Equation (4.12) defines β_W in terms of a wall friction velocity enabling the use of a wall model, see e.g. Bazilevs et al. [2007b].

Although it is desirable to incorporate eq. (4.10) - (4.12) the formulation implemented in the context of this thesis uses a penalty method due to its simplicity (see section 2.4). That is, only the penalty term is added to the weak form:

$$B^{MSW}(\boldsymbol{W}^h, \boldsymbol{U}^h) = B^{MS}(\boldsymbol{W}^h, \boldsymbol{U}^h) + \beta\langle \boldsymbol{w}^h, \boldsymbol{u^h} - \boldsymbol{g} \rangle_{\Gamma_D}, \tag{4.13}$$

where β is chosen to be large enough to enforce the boundary conditions without leading to numerical instabilities (*cf.* eq. 2.18). However, the next step in the developing the Finite Cell method beyond this thesis and exploiting its full potential for turbulence modeling is the extension of the boundary conditions as presented.

4.3 Spatial discretization and linearization

To discretize the weak form, $\boldsymbol{\mathcal{W}}^h$ and $\mathcal{P}^h$ need to be defined. In the context of this work a hierarchical polynomial basis (*cf.* section 2.1) is used. A possibly different polynomial degree is chosen for each field component and space dimension, defining an ansatz order template of the form

$$\boldsymbol{\mathcal{P}} = \begin{bmatrix} p^v_{11}, & \cdots & , & p^v_{1d}, & p^p_1 \\ \vdots & & & \vdots & \vdots \\ p^v_{d1}, & \cdots & , & p^v_{dd}, & p^p_d \end{bmatrix} \tag{4.14}$$

for each element. The numbering is as follows: p^v_{ij} denotes the polynomial degree for velocity component j in space dimension i; p^p_i denotes the degree to discretize the pressure field in space component i. Associated with (4.14) are the basis function vectors $\boldsymbol{N}_j$ and $\boldsymbol{N}_p$ that span the basis for velocity, acceleration and pressure spaces. Note, that accelerations are discretized with the same basis functions as the velocities. In the selection of $\boldsymbol{\mathcal{P}}$ care has to be taken that neighboring elements remain conforming. Using the introduced discretization the primary variables can be expressed as

$$\begin{aligned} \boldsymbol{u}^h(\boldsymbol{x},t) &= \sum_{j=1}^{d} \sum_{A_j=1}^{n_j} N^j_{A_j}(\boldsymbol{x})\, V_{jA_j}(t)\, \boldsymbol{e}_j \\ \dot{\boldsymbol{u}}^h(\boldsymbol{x},t) &= \sum_{j=1}^{d} \sum_{A_j=1}^{n_j} N^j_{A_j}(\boldsymbol{x})\, \dot{V}_{jA_j}(t)\, \boldsymbol{e}_j \\ p^h(\boldsymbol{x},t) &= \sum_{A_p=1}^{n_p} N^p_{A_p}(\boldsymbol{x})\, P_{A_p}(t), \end{aligned}$$

where d and n_j denote the dimensionality of the problem and the number of unknowns for solution component j (*i.e.* the dimensionality of $\mathcal{W}_j$). Furthermore, V_{jA_j}, $\dot{V}_{jA_j}$ and P_{A_p} represent the degrees of freedom for velocities, accelerations and pressure and $\boldsymbol{e}_j$ denotes the jth standard Cartesian basis vector. As a different discretization can be chosen for each solution field component, the shape functions and corresponding indices are different. Despite this, the subscripts in A_j and A_p are dropped to enhance readability in the following derivations. In N^j_A, the range of index A is, therefore, considered with respect to the field component j.

Now, the discrete momentum and continuity residual vectors can be defined as

$$\boldsymbol{R}^M = (R^M_{iA}) = (\, R^M_{11},\, \ldots\, , R^M_{1n_1},\, \ldots\, , R^M_{d1},\, \ldots\, , R^M_{dn_d}\,)^T \tag{4.15a}$$

$$\boldsymbol{R}^C = (R^C_A) \;= (\, R^C_1,\, \ldots\, , R^C_{n_c}\,)^T, \tag{4.15b}$$

where n_i denotes the number of equations for the ith space dimension. The order in which the residual vectors are organized is arbitrary; therefore eq. (4.15) represents just an example. R^M_{iA} and R^C_A are defined as follows

$$\begin{aligned} R^M_{iA} &= B^{MSW}(\{N^i_A\boldsymbol{e}_i, 0\}, \{\boldsymbol{u}^h, p^h\}) - L(\{N^i_A\boldsymbol{e}_i, 0\}) \\ R^C_A &= B^{MSW}(\{\boldsymbol{0}, N^p_A\}, \{\boldsymbol{u}^h, p^h\}) - L(\{\boldsymbol{0}, N^p_A\}). \end{aligned}$$

Note, that the boundary terms arising from eq. (4.13) are also included in the residual.

The nonlinear system arising from requiring the residual to vanish is solved using a Newton-Raphson procedure. The resulting iterations can be performed either in terms of the velocity-pressure degrees of freedom or the acceleration-velocity degrees of freedom. A relation between velocities and accelerations needs to be established, allowing to apply the chain rule. To remain general, $\tilde{\boldsymbol{V}}$ is used instead of $\boldsymbol{V}$ or $\dot{\boldsymbol{V}}$. After defining $\boldsymbol{\phi} = \{\tilde{\boldsymbol{V}}, \boldsymbol{P}\}$, the first-order Taylor expansion of the residual can be computed:

$$\boldsymbol{R}(\boldsymbol{\phi} + \Delta\boldsymbol{\phi}) \approx \boldsymbol{R}(\boldsymbol{\phi}) + \frac{\partial \boldsymbol{R}(\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} \Delta\boldsymbol{\phi}.$$

As the goal is to have $\boldsymbol{R}(\boldsymbol{\phi} + \Delta\boldsymbol{\phi}) = \boldsymbol{0}$, the iteration scheme can be constructed as follows:

$$\frac{\partial \boldsymbol{R}(\boldsymbol{\phi}^l)}{\partial \boldsymbol{\phi}^l} \Delta\boldsymbol{\phi}^{l+1} = -\boldsymbol{R}(\boldsymbol{\phi}^l).$$

The missing piece is now the tangent matrix $\boldsymbol{T} = \boldsymbol{R}(\boldsymbol{\phi}^l)/\partial\boldsymbol{\phi}^l$. It is composed of a velocity part, an acceleration part and a pressure part:

$$\boldsymbol{T} = C_v \boldsymbol{T}_v + C_a \boldsymbol{T}_a + C_p \boldsymbol{T}_p. \tag{4.16}$$

The constants are determined by the time integration scheme and depend on the choice of $\tilde{\boldsymbol{V}}$ and the relation between $\boldsymbol{V}$ and $\dot{\boldsymbol{V}}$. Furthermore, $\boldsymbol{T}$ can be written in a block form as:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{G} \\ \boldsymbol{D} & \boldsymbol{L} \end{bmatrix} = \begin{bmatrix} [\boldsymbol{K}_{ij}] & [\boldsymbol{G}_i] \\ [\boldsymbol{D}_j] & \boldsymbol{L} \end{bmatrix}. \tag{4.17}$$

The block matrices in eq. (4.17) are defined as follows:

$$\begin{aligned}
\boldsymbol{K} &= C_v \left[\frac{\partial \boldsymbol{R}_i^M}{\partial \boldsymbol{V}_j}\right] + C_a \left[\frac{\partial \boldsymbol{R}_i^M}{\partial \dot{\boldsymbol{V}}_j}\right] \\
\boldsymbol{G} &= C_p \left[\frac{\partial \boldsymbol{R}_i^M}{\partial \boldsymbol{P}}\right] \\
\boldsymbol{D} &= C_v \left[\frac{\partial \boldsymbol{R}^C}{\partial \boldsymbol{V}_j}\right] + C_a \left[\frac{\partial \boldsymbol{R}^C}{\partial \dot{\boldsymbol{V}}_j}\right] \\
\boldsymbol{L} &= C_p \frac{\partial \boldsymbol{R}^C}{\partial \boldsymbol{P}},
\end{aligned}$$

with

$$\begin{aligned}
\frac{\partial \boldsymbol{R}_i^M}{\partial \boldsymbol{V}_j} &= \left[\frac{\partial R_{iA}^M}{\partial V_{jB}}\right] & \frac{\partial \boldsymbol{R}_i^M}{\partial \dot{\boldsymbol{V}}_j} &= \left[\frac{\partial R_{iA}^M}{\partial \dot{V}_{jB}}\right] & \frac{\partial \boldsymbol{R}_i^M}{\partial \boldsymbol{P}} &= \left[\frac{\partial R_{iA}^M}{\partial P_B}\right] \\
\frac{\partial \boldsymbol{R}^C}{\partial \boldsymbol{V}_j} &= \left[\frac{\partial R_A^C}{\partial V_{jB}}\right] & \frac{\partial \boldsymbol{R}^C}{\partial \dot{\boldsymbol{V}}_j} &= \left[\frac{\partial R_A^C}{\partial \dot{V}_{jB}}\right] & \frac{\partial \boldsymbol{R}^C}{\partial \boldsymbol{P}} &= \left[\frac{\partial R_A^C}{\partial P_B}\right].
\end{aligned}$$

While i and j define the space dimension and velocity component, the new indices A_i and B_j represent the corresponding equation number and degree of freedom. The resulting matrices are given as

$$\begin{aligned}
\frac{\partial R^M_{iA}}{\partial V_{jB}} =& \langle N^i_A, \boldsymbol{u}^h \cdot \nabla N^j_B\rangle\delta_{ij} + \langle \nabla N^i_A \nu, \nabla N^j_B\rangle\delta_{ij} + \langle \nabla N^i_A \cdot \boldsymbol{e}\nu, \nabla N^j_B \cdot \boldsymbol{e}_i\rangle \\
&+ \langle \boldsymbol{u}^h \cdot \nabla N^i_A \tau_M, \boldsymbol{u}^h \cdot \nabla N^j_B\rangle\delta_{ij} + \langle \nabla N^i_A \cdot \boldsymbol{e}_i \tau_C, \nabla N^j_B \cdot \boldsymbol{e}_j\rangle_{\Omega'} \\
&+ \beta\langle N^i_A, N^j_B\rangle_{\Gamma_D} \\
\frac{\partial R^M_{iA}}{\partial \dot{V}_{jB}} =& \langle N^i_A, N^j_B\rangle\delta_{ij} + \langle \boldsymbol{u}^h \cdot \nabla N^i_A \tau_M, N^j_B\rangle\delta_{ij} \\
\frac{\partial R^M_{iA}}{\partial P_B} =& -\langle \nabla N^i_A \cdot \boldsymbol{e}_i, N^p_B\rangle + \langle \boldsymbol{u^h} \cdot \nabla N^i_A \boldsymbol{e}_i \tau_M, \nabla N^p_B\rangle_{\Omega'} \qquad (4.18) \\
\frac{\partial R^C_{A}}{\partial V_{jB}} =& \langle N^i_A, \nabla N^p_B \cdot \boldsymbol{e}_j\rangle + \langle \nabla N^i_A \tau_M, \boldsymbol{u}^h \cdot \nabla N^p_B \boldsymbol{e}_j\rangle \\
\frac{\partial R^C_{A}}{\partial \dot{V}_{jB}} =& \langle \nabla N^p_A \tau_M, N^p_B \boldsymbol{e}_j\rangle \\
\frac{\partial R^C_{A}}{\partial P_B} =& \langle \nabla N^p_A \tau_M, \nabla N^p_B\rangle_{\Omega'}.
\end{aligned}$$

Note, that the last term in the definition of $\partial R^M_{iA}/\partial V_{jB}$ represents the linearization of the penalty term introduced in eq. (4.13).

Following section 2.3, the last step is to apply the Finite Cell methodology by multiplying B^{MS} and L with $\alpha(\boldsymbol{x})$ (*cf.* eq. 2.16):

$$B^{FCM} = \alpha(\boldsymbol{x})B^{MS} \qquad (4.19a)$$

$$L^{FCM} = \alpha(\boldsymbol{x})L. \qquad (4.19b)$$

This way, the geometry of the physical domain can be recovered. An extension of an existing code towards the Finite Cell Method requires, therefore, an adaption of the integration scheme and a multiplication by $\alpha(\boldsymbol{x})$. Note, that the linearization of eq. (4.19) simply adds a factor $\alpha(\boldsymbol{x})$ to the matrices defined in eq. (4.18). This can be implemented e.g. by adjusting the weight of integration points. From a physical point of view, the Finite Cell Method reduces the density and the viscosity of the fluid in Ω_{fict} to a very small value. Also, incompressibility is not enforced in Ω_{fict}, and the pressure gradient is only acting on Ω_{phys}.

As suggested in section 2.3, the stabilization of the weak form on the fictitious domain is not required to be a scaled version of the original weak form. In this work, the following form has been used to maintain uniqueness in Ω_{fict}:

$$\langle \nabla \boldsymbol{w}^h, \epsilon\nu\nabla\dot{\boldsymbol{u}}^h\rangle_{\Omega_{fict}} + \langle \nabla q^h, \epsilon\nu\nabla p^h\rangle_{\Omega_{fict}} = 0 \qquad (4.20)$$

This particular choice originates from the fact that the nonlinear iterations within the time integration scheme (presented in section 4.4) are performed in terms of $\dot{\boldsymbol{V}}$. In numerical tests eq. (4.20) has been shown to be more stable and deliver a better conditioned system compared to the full stabilized VMS weak form. Moreover the computational time is decreased significantly as the evaluation requires less effort compared to eq. (4.8). However, the performance of eq. (4.20) has to be evaluated in future investigations and serves only as a first experiment. The original approach in the Finite Cell Method that defines the weak form of the problem also on the fictitious domain reduces the influence of the residual on Ω_{fict} independent of the physical parameters. Using a modified weak form instead, adds another layer of difficulty in choosing ϵ as the influence of the terms on Ω_{fict} is not automatically reduced by a constant factor. Therefore, a generic definition of ϵ potentially includes problem dependent quantities.

4.4 Generalized-α time integration

As suggested in Bazilevs et al. [2007a], the semi-discrete Navier-Stokes weak form is advanced in time using a generalized-α scheme. Extended to first order systems by Jansen et al. [2000], the generalized-α scheme allows to damp high frequencies while remaining second-order accurate.

As mentioned in the previous section, the nonlinear system can be solved either in terms of velocities or accelerations. In this work, an acceleration-based formulation is used. With this, the method becomes: Given $\boldsymbol{V}^n$ and $\dot{\boldsymbol{V}}^n$ at time step n, find the intermediate quantities $\boldsymbol{V}^{n+\alpha_f}$ and $\dot{\boldsymbol{V}}^{n+\alpha_m}$ as well as new solution $\boldsymbol{V}^{n+1}$, $\dot{\boldsymbol{V}}^{n+1}$ and $\boldsymbol{P}^{n+1}$, such that

$$\boldsymbol{R}^M(\boldsymbol{V}^{n+\alpha_f}, \dot{\boldsymbol{V}}^{n+\alpha_m}, \boldsymbol{P}^{n+1}) = \boldsymbol{0} \tag{4.21a}$$

$$\boldsymbol{R}^C(\boldsymbol{V}^{n+\alpha_f}, \dot{\boldsymbol{V}}^{n+\alpha_m}, \boldsymbol{P}^{n+1}) = \boldsymbol{0}, \tag{4.21b}$$

where

$$\boldsymbol{V}^{n+1} = \boldsymbol{V}^n + \Delta t \dot{\boldsymbol{V}}^n + \gamma \Delta t (\dot{\boldsymbol{V}}^{n+1} - \dot{\boldsymbol{V}}^n) \tag{4.22}$$

$$\boldsymbol{V}^{n+\alpha_f} = \boldsymbol{V}^n + \alpha_f (\boldsymbol{V}^{n+1} - \boldsymbol{V}^n) \tag{4.23}$$

$$\dot{\boldsymbol{V}}^{n+\alpha_m} = \dot{\boldsymbol{V}}^n + \alpha_m (\dot{\boldsymbol{V}}^{n+1} - \dot{\boldsymbol{V}}^n). \tag{4.24}$$

The three parameters α_m, α_f and γ determine the method. While $\alpha_m \leq \alpha_f \leq 1/2$ is required for stability, second-order accuracy in time is achieved if $\gamma = 1/2 + \alpha_m - \alpha_f$ [Jansen et al., 2000]. The family of generalized-α methods is obtained by defining the parameters as follows:

$$\alpha_m = \frac{1}{2}\left(\frac{3-\rho_\infty}{1+\rho_\infty}\right)$$

$$\alpha_f = \frac{1}{1+\rho_\infty},$$

with a spectral radius ρ_∞ controlling the numerical damping of high frequencies (in time). Throughout this thesis, ρ_∞ was selected as $1/2$. Numerical studies on the influence of ρ on the solution can be found in [Jansen et al., 2000]. Besides that, a fully implicit backward Euler scheme is obtained by choosing $\alpha_m = \alpha_f = \gamma = 1$.

As the residual is evaluated at α_m and α_f, but the solution is searched at time step $n+1$, the linearization of the residual requires the application of the chain rule:

$$\frac{\partial \boldsymbol{R}^M(\boldsymbol{V}^{n+\alpha_f}, \dot{\boldsymbol{V}}^{n+\alpha_m}, \boldsymbol{P}^{n+1})}{\partial \dot{\boldsymbol{V}}^{n+1}} = \frac{\partial \boldsymbol{R}^M}{\partial \boldsymbol{V}^{n+\alpha_f}} \frac{\partial \boldsymbol{V}^{n+\alpha_f}}{\partial \dot{\boldsymbol{V}}^{n+1}} + \frac{\partial \boldsymbol{R}^M}{\partial \dot{\boldsymbol{V}}^{n+\alpha_m}} \frac{\partial \dot{\boldsymbol{V}}^{n+\alpha_m}}{\partial \dot{\boldsymbol{V}}^{n+1}} \tag{4.25}$$

The partial derivatives in eq. (4.25) represent the dependencies of the residual parameters on the solution variable $\dot{\boldsymbol{V}}^{n+1}$. The first partial derivative follows from (4.24):

$$\frac{\partial \dot{\boldsymbol{V}}^{n+\alpha_m}}{\partial \dot{\boldsymbol{V}}^{n+1}} = \frac{\partial}{\partial \dot{\boldsymbol{V}}^{n+1}} \left[\dot{\boldsymbol{V}}^n + \alpha_m (\dot{\boldsymbol{V}}^{n+1} - \dot{\boldsymbol{V}}^n) \right] = \alpha_m$$

and the second partial derivative from inserting eq. (4.23) into eq. (4.22):

$$\begin{aligned} \frac{\partial \boldsymbol{V}^{n+\alpha_f}}{\partial \dot{\boldsymbol{V}}^{n+1}} &= \frac{\partial}{\partial \dot{\boldsymbol{V}}^{n+1}} \left[\boldsymbol{V}^n + \alpha_f (\boldsymbol{V}^{n+1} - \boldsymbol{V}^n) \right] \\ &= \frac{\partial}{\partial \dot{\boldsymbol{V}}^{n+1}} \left[\boldsymbol{V}^n + \alpha_f \left(\left[\boldsymbol{V}^n + \Delta t \dot{\boldsymbol{V}}^n + \gamma \Delta t (\dot{\boldsymbol{V}}^{n+1} - \dot{\boldsymbol{V}}^n) \right] - \boldsymbol{V}^n \right) \right] \\ &= \alpha_f \gamma \Delta t \end{aligned}$$

This defines the constants required in the computation of the tangent matrix (*cf.* eq. 4.16):

$$C_a = \alpha_m \qquad C_v = \alpha_f \gamma \Delta t \qquad C_p = 1.$$

Note, that the pressure is evaluated at $n+1$, therefore no chain rule is required (giving $C_p = 1$).

The time integration described in the following is summarized in algorithm 1. Each time step begins with evaluating a predictor to initialize the nonlinear iterations that attempt to solve eq. (4.21). Throughout this work, the following predictor is used:

$$\boldsymbol{V}^{n+1,0} = \boldsymbol{V}^n \tag{4.26a}$$

$$\dot{\boldsymbol{V}}^{n+1,0} = \frac{\gamma - 1}{\gamma} \dot{\boldsymbol{V}}^n \tag{4.26b}$$

$$\boldsymbol{P}^{n+1,0} = \boldsymbol{P}^n, \tag{4.26c}$$

Several other definitions are discussed in Jansen et al. [2000]. Now follows a loop of nonlinear iterations which begin with evaluating quantities at the intermediate step, defined by α_f and α_m:

$$\boldsymbol{V}^{n+\alpha_f,l} = \boldsymbol{V}^n + \alpha_f (\boldsymbol{V}^{n+1,l-1} - \boldsymbol{V}^n) \tag{4.27a}$$

$$\dot{\boldsymbol{V}}^{n+\alpha_m,l} = \dot{\boldsymbol{V}}^n + \alpha_m (\dot{\boldsymbol{V}}^{n+1,l-1} - \dot{\boldsymbol{V}}^n) \tag{4.27b}$$

$$\boldsymbol{P}^{n+1,l} = \boldsymbol{P}^{n+1,l-1}, \tag{4.27c}$$

where l is the index of the nonlinear iteration $l = 1, \ldots, l_{max}$. In the second step, the tangent matrix and the residual are assembled using $\boldsymbol{V}^{n+\alpha_f,l}$, $\dot{\boldsymbol{V}}^{n+\alpha_m,l}$ and $\boldsymbol{P}^{n+1,l}$. The linear system

$$\begin{bmatrix} \boldsymbol{K}^l & \boldsymbol{G}^l \\ \boldsymbol{D}^l & \boldsymbol{L}^l \end{bmatrix} \begin{pmatrix} \Delta \dot{\boldsymbol{V}}^{n+1,l} \\ \Delta \boldsymbol{P}^{n+1,l} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{R}^M_l \\ -\boldsymbol{R}^C_l \end{pmatrix} \tag{4.28}$$

is then solved. In the third and last step of the nonlinear iterations, the solution is updated

Algorithm 1 Time integration

```
function performTimeIntegration($\boldsymbol{V}^0$, $\dot{\boldsymbol{V}}^0$, $\boldsymbol{P}^0$, $\Delta t$, $n$)
    // Initialize variables
    $\boldsymbol{V}^n = \boldsymbol{V}^0$
    $\dot{\boldsymbol{V}}^n = \dot{\boldsymbol{V}}^0$
    $\boldsymbol{P}^n = \boldsymbol{P}^0$

    // Start time loop
    for $n = 0$ to $n_{max}$ do
        // Predict solution at next time step (cf. eq. (4.26))
        $\boldsymbol{V}^{n+1} = \boldsymbol{V}^n$
        $\dot{\boldsymbol{V}}^{n+1} = (\gamma - 1)/\gamma\, \dot{\boldsymbol{V}}^n$
        $\boldsymbol{P}^{n+1} = \boldsymbol{P}^n$

        // Start Newton-Raphson iterations
        for $l = 0$ to $l_{max}$ do
            // Evaluate quantities at intermediate step (cf. eq. (4.27))
            $\boldsymbol{V}^{n+\alpha_f} = \boldsymbol{V}^n + \alpha_f(\boldsymbol{V}^{n+1} - \boldsymbol{V}^n)$
            $\dot{\boldsymbol{V}}^{n+\alpha_m} = \dot{\boldsymbol{V}}^n + \alpha_m(\dot{\boldsymbol{V}}^{n+1} - \dot{\boldsymbol{V}}^n)$

            // Assemble linear system
            $\boldsymbol{T}$ = integrateTangentMatrix($\boldsymbol{V}^{n+\alpha_f}$, $\dot{\boldsymbol{V}}^{n+\alpha_m}$, $\boldsymbol{P}^{n+1}$, $\Delta t$)
            $\boldsymbol{R}$ = integrateResidual($\boldsymbol{V}^{n+\alpha_f}$, $\dot{\boldsymbol{V}}^{n+\alpha_m}$, $\boldsymbol{P}^{n+1}$, $\Delta t$)

            // Solve for update (eq. (4.28))
            $\Delta\dot{\boldsymbol{V}}, \Delta\boldsymbol{P}$ = solveLinearSystem($\boldsymbol{T}$, $\boldsymbol{R}$)

            // Update solution variables (eq. (4.29))
            $\dot{\boldsymbol{V}}^{n+1} = \dot{\boldsymbol{V}}^{n+1} + \Delta\dot{\boldsymbol{V}}$
            $\boldsymbol{V}^{n+1} = \boldsymbol{V}^{n+1} + \gamma\Delta t\Delta\dot{\boldsymbol{V}}$
            $\boldsymbol{P}^{n+1} = \boldsymbol{P}^{n+1} + \Delta\boldsymbol{P}$

            if nonlinearIterationsConverged( ) then
                break $l$
            end if
        end for

        // Now update and proceed to the next time step
        $\dot{\boldsymbol{V}}^n = \dot{\boldsymbol{V}}^{n+1}$
        $\boldsymbol{V}^n = \boldsymbol{V}^{n+1}$
        $\boldsymbol{P}^n = \boldsymbol{P}^{n+1}$
    end for

    // Return latest solution
    return $\boldsymbol{V}^n$, $\dot{\boldsymbol{V}}^n$, $\boldsymbol{P}^n$

end function
```

as follows

$$\dot{\boldsymbol{V}}^{n+1,l} = \dot{\boldsymbol{V}}^{n+1,l-1} + \Delta\dot{\boldsymbol{V}}^{n+1,l} \tag{4.29a}$$

$$\boldsymbol{V}^{n+1,l} = \boldsymbol{V}^{n+1,l-1} + \gamma\Delta t\Delta\dot{\boldsymbol{V}}^{n+1,l} \tag{4.29b}$$

$$\boldsymbol{P}^{n+1,l} = \boldsymbol{P}^{n+1,l-1} + \Delta\boldsymbol{P}^{n+1,l}. \tag{4.29c}$$

The iterations are stopped once convergence is achieved. Then, the solution fields at time step n are updated with the newly obtained values and the algorithm proceeds to the next time step.

The convergence of the Newton-Raphson scheme can be measured in several ways. In the current work, the scalar product of residual and increment and the norm of the residual were considered:

$$\begin{aligned} e_1 &= \left| \, \boldsymbol{R}_l \cdot (\Delta\dot{\boldsymbol{V}}^{n+1,l}, \Delta\boldsymbol{P}^{n+1,l}) \, \right| \\ e_2 &= \|\boldsymbol{R}_l\|. \end{aligned}$$

After the error in the chosen norm was decreased by several orders of magnitude, the scheme is considered to be converged.

To start the time integration, it is often desirable to use several first-order implicit steps. Especially oscillations in time arising from non-matching initial and boundary conditions can be eliminated this way (see algorithm 2). Furthermore, the efficiency of the scheme can be increased by an adaptive selection of the time step Δt . One method has been proposed by Vignal et al. [2017] that estimates the error using a first-order backward approximation and does not require the computation of an additional solution.

Algorithm 2 Solution scheme

```
// Get discrete initial conditions (e.g. by an $L^2$-projection)
$\boldsymbol{V}^0$, $\dot{\boldsymbol{V}}^0$, $\boldsymbol{P}^0$ = getInitialConditions( )

$n_0 = 2$
$\Delta t_0 = \Delta t / n_0$

// Integrate first time step with $n_0$ backward-Euler steps
$\boldsymbol{V}^1$, $\dot{\boldsymbol{V}}^1$, $\boldsymbol{P}^1$ = performTimeIntegration($\boldsymbol{V}^0$, $\dot{\boldsymbol{V}}^0$, $\boldsymbol{P}^0$, $\Delta t_0$, $n_0$)

// Start the actual time integration
$\boldsymbol{V}^n$, $\dot{\boldsymbol{V}}^n$, $\boldsymbol{P}^n$ = performTimeIntegration($\boldsymbol{V}^1$, $\dot{\boldsymbol{V}}^1$, $\boldsymbol{P}^1$, $\Delta t$, $n_{max} - 1$)
```

Chapter 5

Results

To show the performance of the theory introduced in the previous chapters several benchmarks are computed and analyzed in the following. To this end, the lid driven cavity example and the flow around a cylinder are considered. Although the computations are only two-dimensional, the methods and the implementation are not restricted to this.

5.1 Lid driven cavity

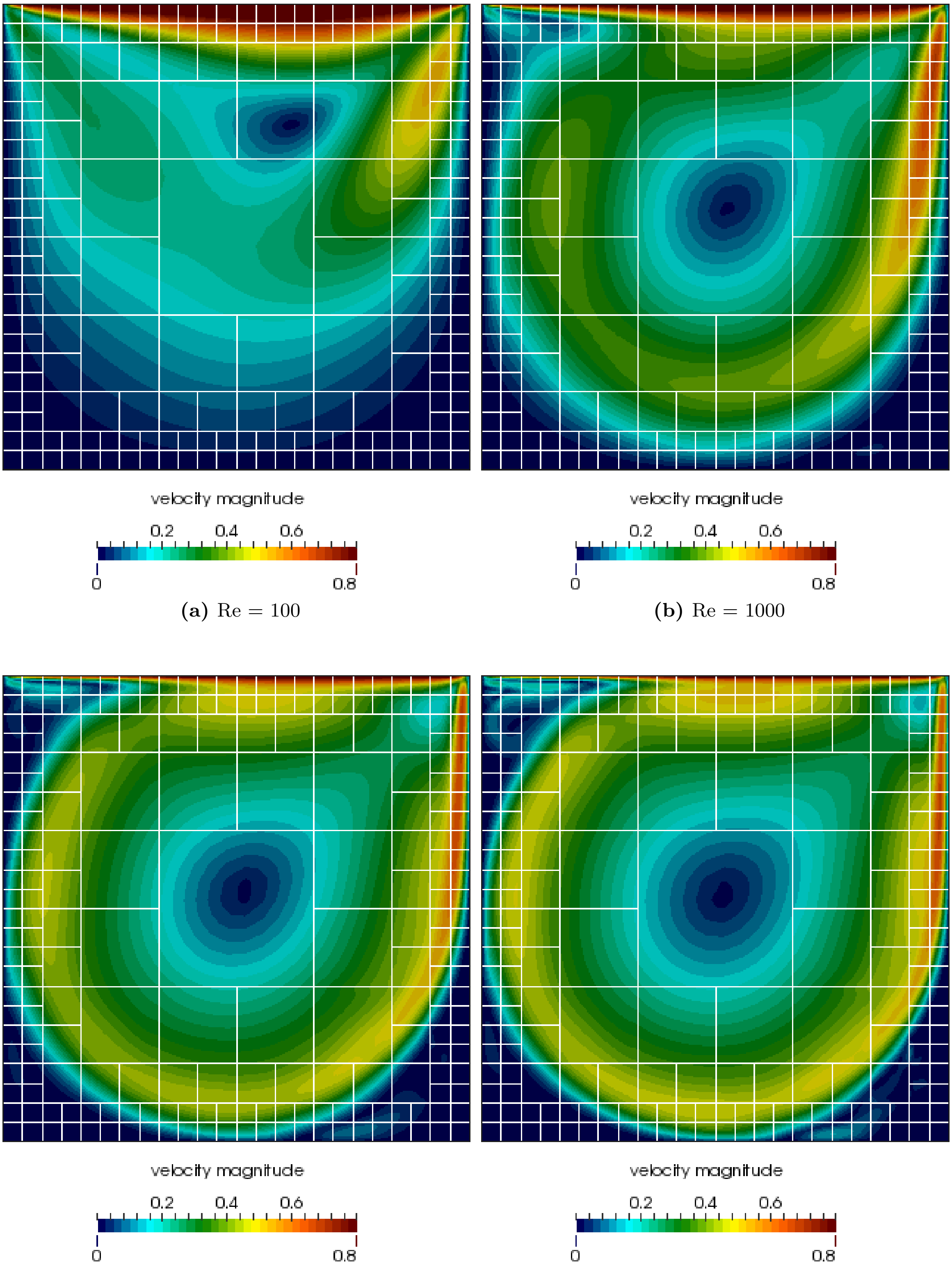
In the lid-driven cavity benchmark, a square domain with fluid inside is excited by a prescribed tangential velocity component on the top. Together with no-slip conditions on the remaining boundary a circulation emerges. The flow characteristics change depending on the Reynolds number defined as

$$Re = \frac{u_c L^c}{\nu}, \tag{5.1}$$

where u_c and L^c are the characteristic velocity and length scales. They were chosen as the prescribed velocity (on the top) and the side length of the domain. The problem is discretized for different Reynolds numbers using a conforming mesh with three levels of multi-level hp-refinement towards the boundaries. The setting is summarized in table 5.1. Figure 5.1 shows the velocity contours and fig. 5.2 - 5.3 compare the solution along the middle lines at $x = 0.5$ and $y = 0.5$ to a reference obtained by Ghia et al. [1982]. Despite the relatively

space	number of elements		3×3
	maximum refinement depth		3
	polynomial degree		3
	number of dofs		6240
time	Δt		0.025
	number of time steps	$Re = 100$	160
		$Re = 1000$	1600
		$Re = 5000$	8000
		$Re = 10000$	8000

Table 5.1: Discretization of lid driven cavity example.



(a) Re = 100

(b) Re = 1000

(c) Re = 5000

(d) Re = 10000

Figure 5.1: Velocity contours of the lid driven cavity example for different Reynolds numbers.

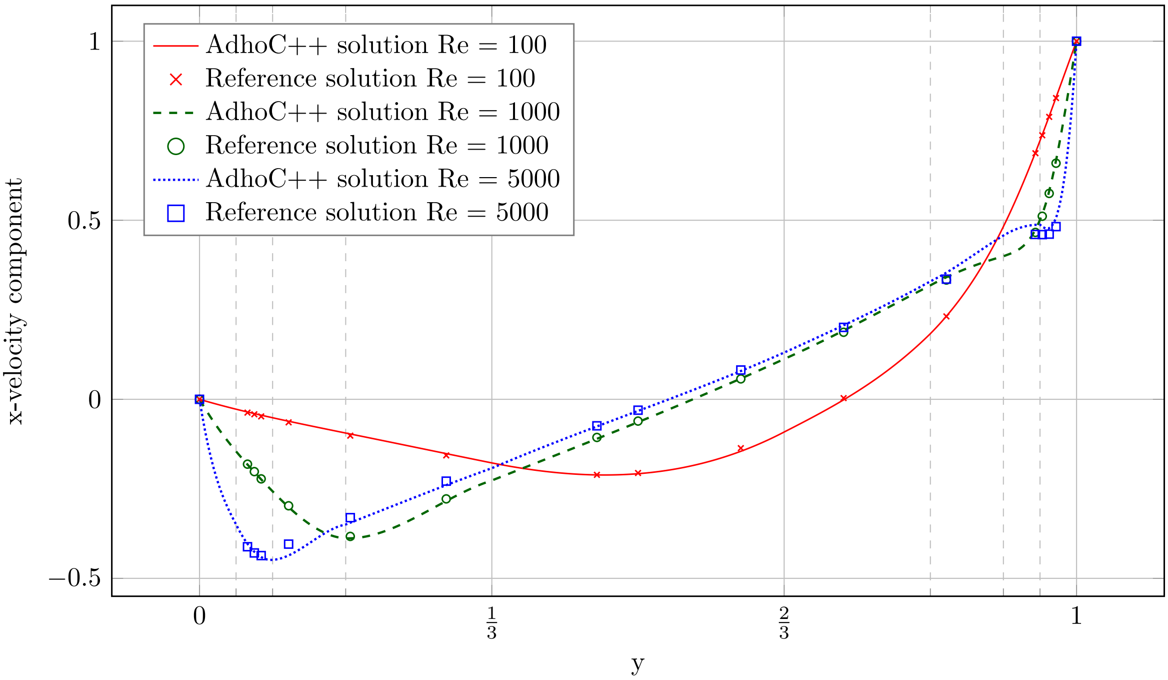


Figure 5.2: Velocity profile comparison at x = 0.5 with Ghia et al. [1982] for the lid driven cavity benchmark.

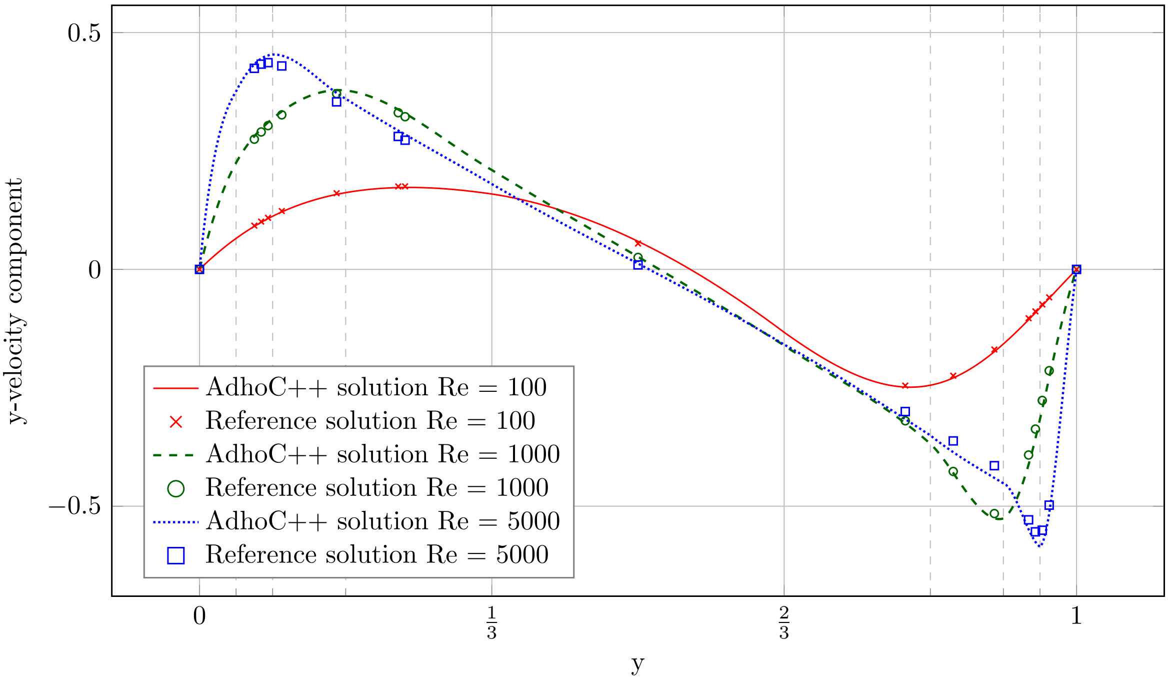


Figure 5.3: Velocity profile comparison at y = 0.5 with Ghia et al. [1982] for the lid driven cavity benchmark.

coarse discretization, the solution coincides very well with the reference. This underlines the effectiveness and flexibility of the multi-level hp-method. In combination with C^0 Finite Elements, the local features such the corner singularity in this example can be decoupled from the rest of the discretization. This could be even further improved by extending the adaptive scheme to an anisotropic refinement where elements are bisected only in one spatial direction.

5.2 Flow around a cylinder

To investigate the performance of the Finite Cell Method, a two-dimensional flow around a cylinder as defined in Schäfer and Turek [1996] is considered. Above $Re \approx 50$, a periodic vortex shedding (Von Kármán vortex street) arises. The assumption of a two-dimensional nature of the problem is only valid up to a transition zone from $Re \approx 190$ to $Re \approx 400$ after which the flow becomes turbulent and the velocity does not remain uniform in z-direction. A numerical study of this transition can be found in Rajani et al. [2009] where the development of the vertical profile for increasing Reynolds numbers is shown. The problem definition is sketched in fig. 5.4. To introduce a small asymmetry, the cylinder is placed slightly below $l_y/2$. In the first benchmark, the inflow velocity is chosen as $u_0 = 0.3$, resulting in a steady flow profile with $Re = 20$. The Reynolds number is calculated according to eq. (5.1) using as characteristic velocity the mean value of the inflow profile:

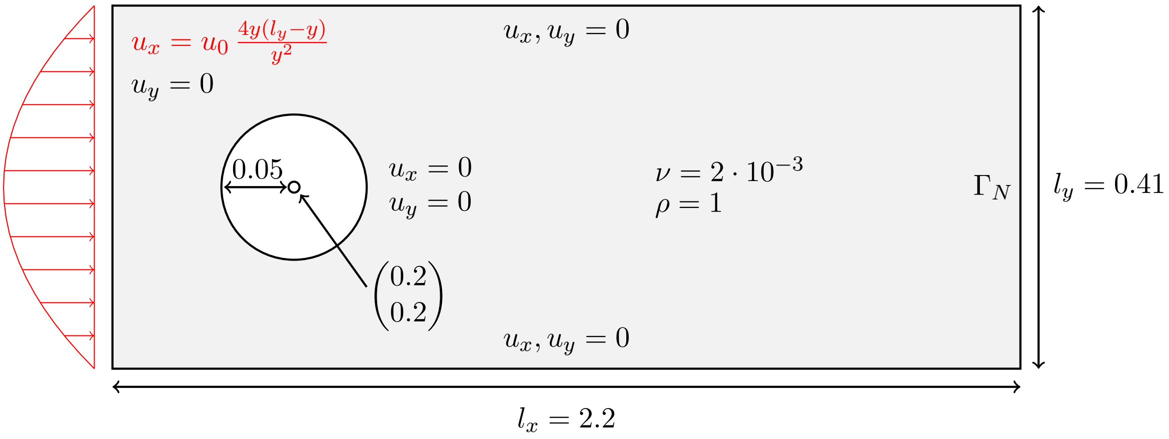


Figure 5.4: Flow around cylinder following Schäfer and Turek [1996].

$$u_c = \frac{2}{3} u_0.$$

The second benchmark in Schäfer and Turek [1996] investigates the transient behavior at $Re = 100$, obtained by selecting $u_0 = 1.5$.

For both scenarios the reference values are compared to the solution obtained with a conforming grid and to the solution obtained with the Finite Cell Method. For both methods the time step was chosen as $\Delta t = 0.01$ in the steady case and $\Delta t = 0.0025$ in the unsteady case. The initial velocity and pressure fields are assumed to be zero. As the computation of the Hessian matrix of the element mapping has not been implemented, the second derivatives are omitted in computations with conforming meshes. Furthermore, the drag and lift coefficients are computed as follows:

$$C^D = C_f \int_S \tau n_y - p n_x \, dS \qquad C^L = C_f \int_S \tau n_x - p n_y \, dS, \tag{5.2}$$

with

$$\begin{aligned} C_f &= \frac{1}{\rho r u_c^2} \\ \tau &= \rho\nu \frac{\partial u_t}{\partial \boldsymbol{n}} = \rho\nu\Big(\boldsymbol{n} \cdot \sum_i t_i \nabla u_i\Big) \\ \boldsymbol{t} &= \begin{pmatrix} n_y \\ -n_x \end{pmatrix}. \end{aligned}$$

The line integral over the circle is carried out numerically using a Gauss-Legendre scheme with $p+1$ integration points.

5.2.1 Conforming mesh

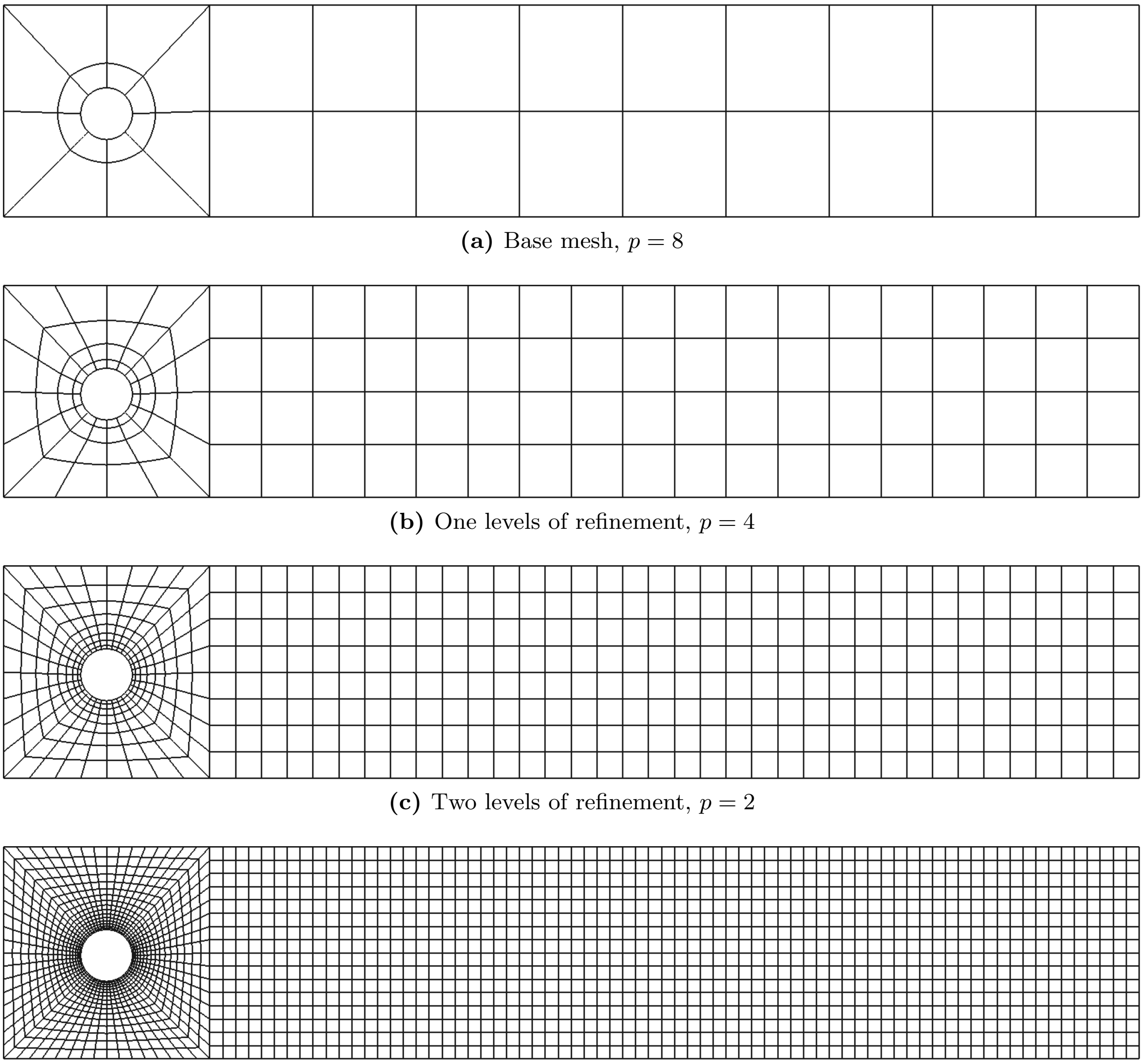
A conforming Finite Element mesh is obtained utilizing blended quadrilaterals. The element edges corresponding to the boundary of the cylinder consist of parametric arcs. The two benchmark scenarios are solved on four discretizations. Starting from a base mesh with polynomial degree 8 layers of multi-level hp-refinement are added successively. To retain a total number of 6936 degrees of freedom, the polynomial degree is multiplied by $1/2$ each time. The resulting meshes are plotted in fig. 5.5. In all computations a penalty value of $\beta = 10^6$ was used to apply boundary conditions.

The results in the steady case are summarized in table 5.2. Moreover, fig. 5.6 shows the temporal evolution of drag and lift coefficients until a steady state is reached. For $p = 1$ both, drag and lift coefficients deviate significantly from the reference values. Increasing the polynomial degree results in a much better agreement.

	C^D	C^L
$p = 1$	5.788	0.0055
$p = 2$	5.600	0.0102
$p = 4$	5.541	0.0094
$p = 8$	5.561	0.0103
Schäfer & Turek	5.57 – 5.59	0.0104 – 0.0110

Table 5.2: Drag and lift coefficients for $Re = 20$ evaluated at $t = 10$ using a conforming discretization.

The velocity and pressure fields for $Re = 100$ are shown in fig. 5.7 and fig. 5.8 at the respective time step with maximum lift value in fig. 5.10. The gray lines represent the reference solution bounds from Schäfer and Turek [1996]. Figure 5.9 shows the temporal evolution of drag and lift coefficients for $p = 4$. While in the solution fields no significant difference is seen, the drag and lift forces decrease with p, showing also a considerable difference in phase. In contrast



(a) Base mesh, $p = 8$

(b) One levels of refinement, $p = 4$

(c) Two levels of refinement, $p = 2$

(d) Three levels of refinement, $p = 1$

Figure 5.5: Conforming spatial discretizations of the flow around a cylinder.

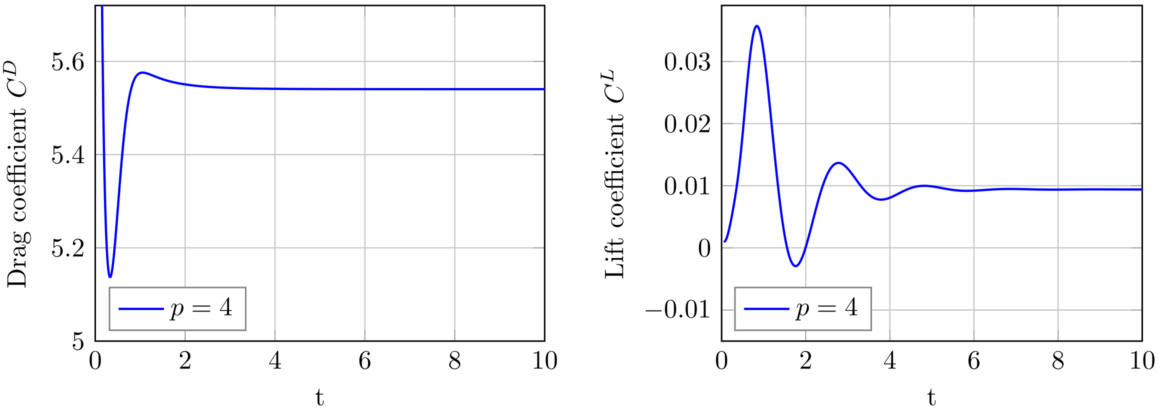
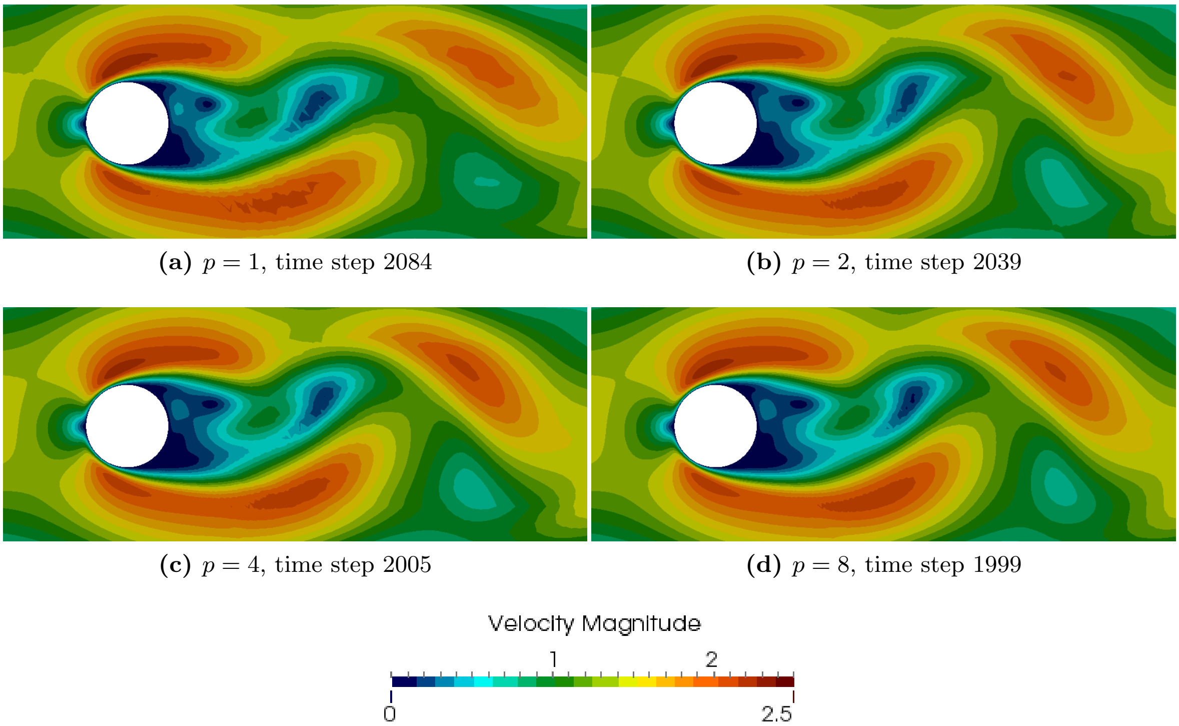


Figure 5.6: Evolution of C^D and C^L for $Re = 20$ using a conforming discretization.



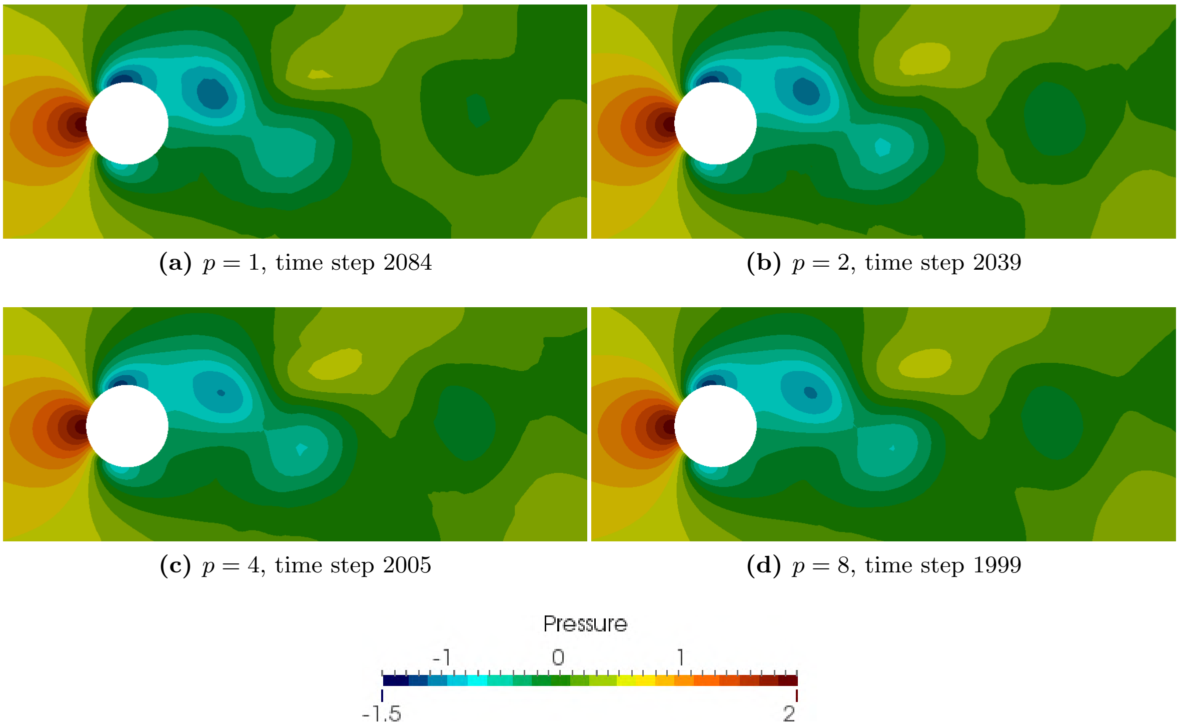
(a) $p = 1$, time step 2084

(b) $p = 2$, time step 2039

(c) $p = 4$, time step 2005

(d) $p = 8$, time step 1999

Figure 5.7: Velocity magnitude contours for conforming discretizations of the flow around a cylinder.



(a) $p = 1$, time step 2084

(b) $p = 2$, time step 2039

(c) $p = 4$, time step 2005

(d) $p = 8$, time step 1999

Figure 5.8: Pressure contours for conforming discretizations of the flow around a cylinder.

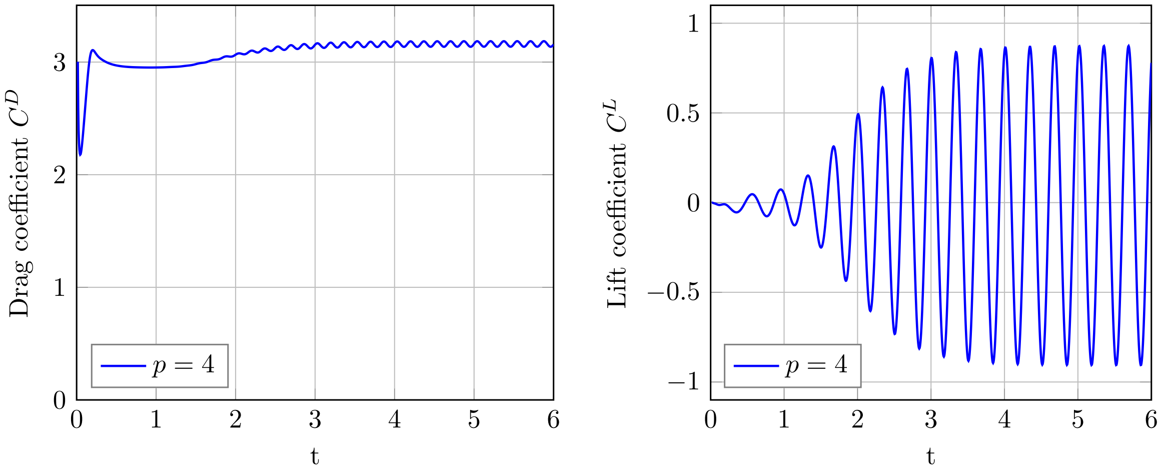


Figure 5.9: Evolution of drag and lift coefficients for a conforming discretization with $p = 4$.

to the steady computations the quality of the results does not increase for higher polynomial degrees. As in the unsteady example the influence of shear forces becomes more important, the deviation can be attributed more to an error in the velocity fields. In general the results have shown to be sensitive to the time step. That is, a larger time step causes an over-diffusive behavior and therefore leads to an underestimation in the drag and lift forces.

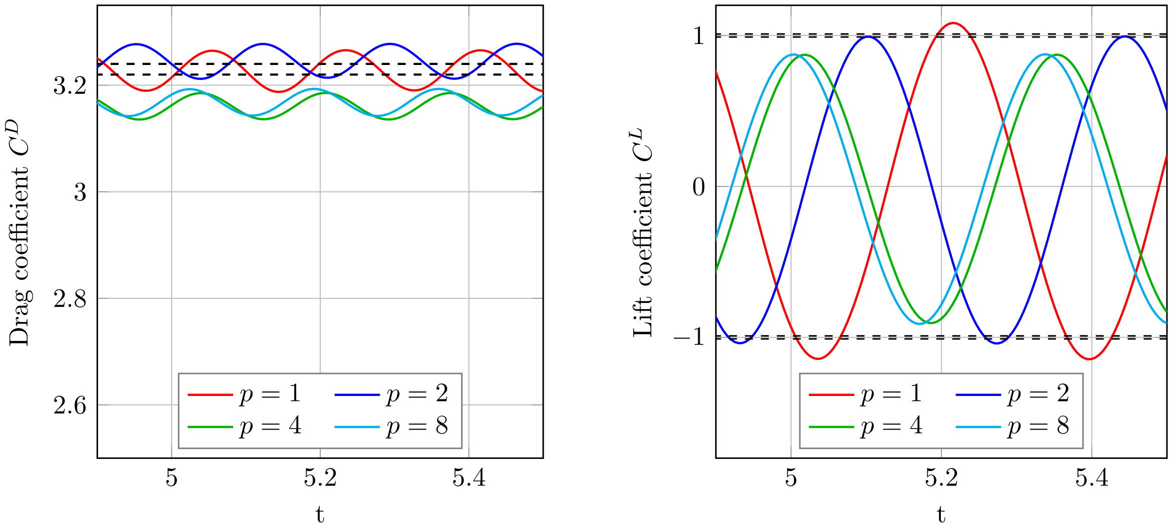


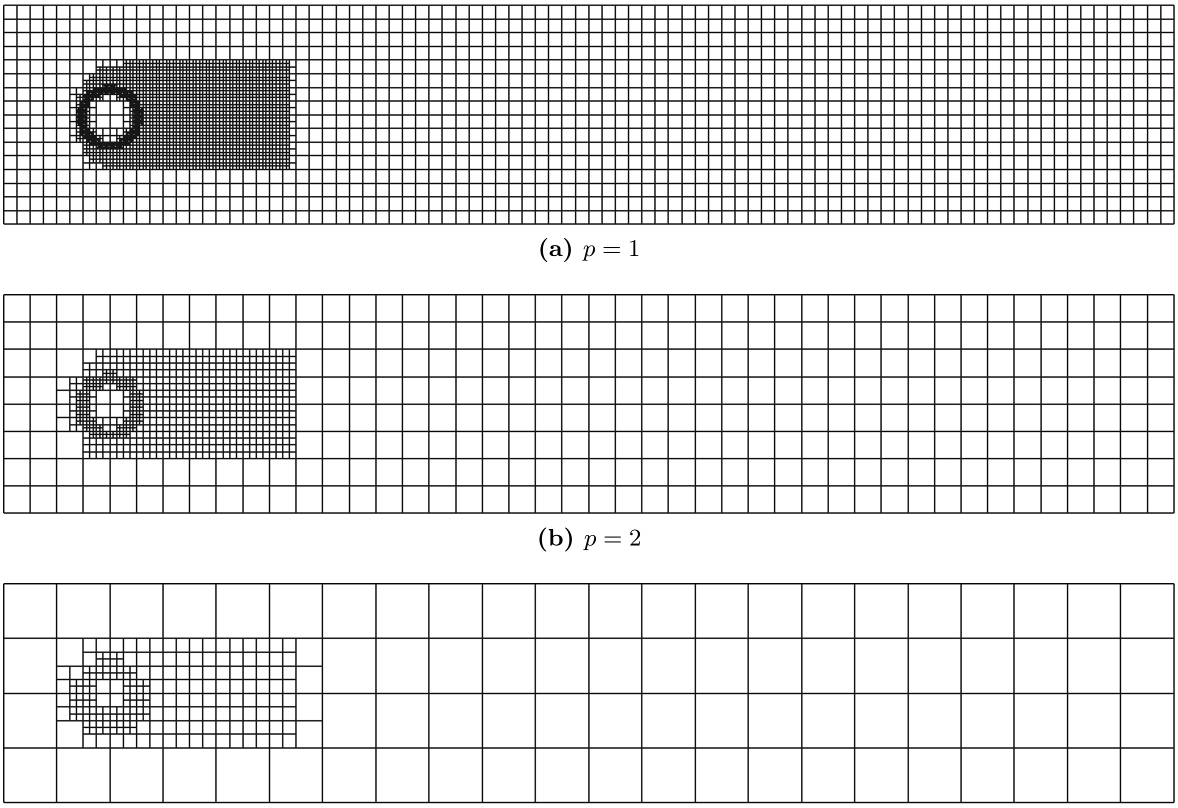
Figure 5.10: Drag and lift coefficients for different conforming discretizations.

5.2.2 Finite Cell discretization

In this section, the flow around a cylinder is discretized using the Finite Cell method. Additionally, the mesh is refined towards the cylinder inside a larger box with two levels and towards the cylinder boundary with three levels of multi-level *hp*-refinement. The different configurations are summarized in table 5.3 and plotted in fig. 5.11. Throughout this section the blended partitioning approach (*cf.* Kudela et al. [2015]) was used to handle the discontinuity in the FCM. The differences to computations using a space tree approach with depth $p+1$ were not significant. The configuration with a base mesh of 11×2 elements and $p = 8$ is not presented here, as it exhibits major oscillations around the cylinder arising from numerical instabilities related to a high penalty value and a small ϵ in the Finite Cell method. Moreover, the difference between using a Finite Cell discretization, *i.e.* eliminating the influence from Ω_{fict}, and only applying boundary conditions in an immersed sense is investigated. To this end, the discretization with $p = 4$ is also computed with $\alpha = 1$ and using the full VMS Navier-Stokes formulation on Ω_{fict}.

p	base mesh	β	ϵ	number of unknowns
1	88×16	10^6	10^{-7}	10593
2	44×8	10^5	10^{-6}	11685
4	22×4	10^6	10^{-7}	13899
4	22×4	10^6	1	13899

Table 5.3: Finite Cell discretizations of the flow around a cylinder.


(a) $p = 1$

(b) $p = 2$

(c) $p = 4$

Figure 5.11: Finite Cell discretizations of the flow around a cylinder.

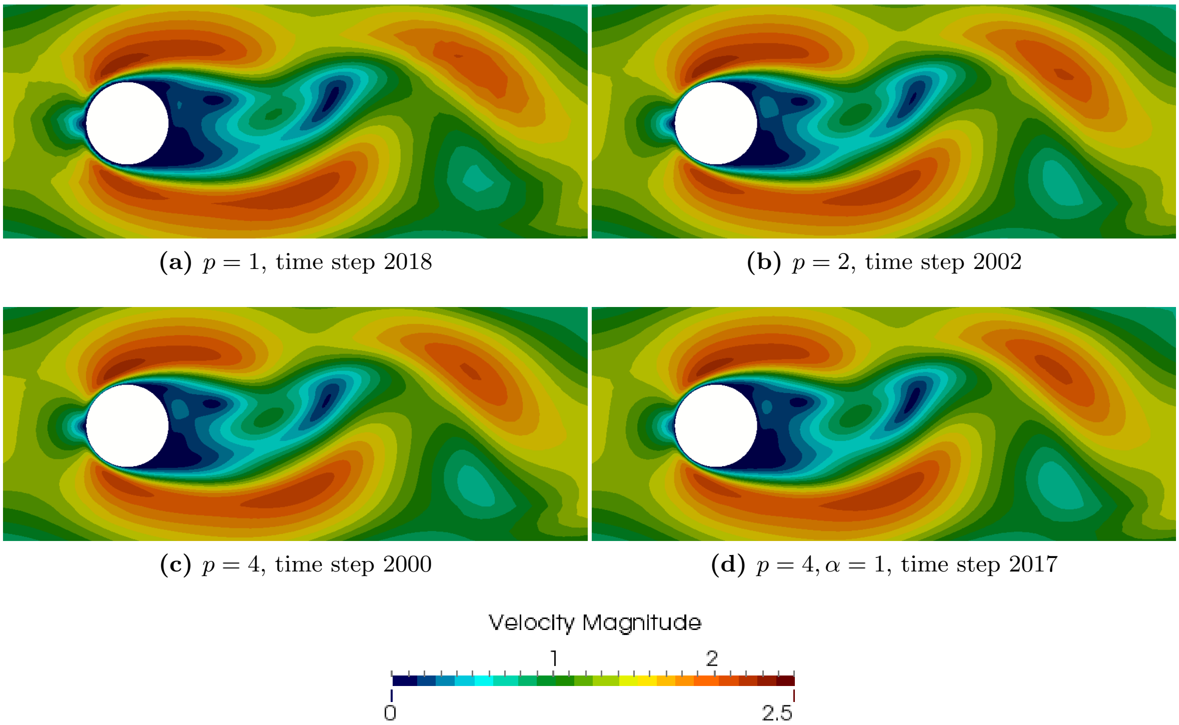
Again, we analyze the steady setting first. Table 5.4 compares the drag and lift forces obtained with the Finite Cell Method. Similar to the conforming discretization, the results for lower polynomial degrees do not agree well with the reference. This is even more severe as for $p = 1$ in particular the lift forces are completely wrong.

Now, the unsteady case is analyzed. Figures 5.12 - 5.13 show the velocity and pressure fields.

	C^D	C^L
$p = 1$	4.735	-0.0418
$p = 2$	5.576	0.0444
$p = 4$	5.580	0.0095
$p = 4, \alpha = 1$	2.821	0.0060
Schäfer & Turek	$5.57 - 5.59$	$0.0104 - 0.0110$

Table 5.4: Drag and lift coefficients for $Re = 20$ evaluated at $t = 10$ using FCM.

The drag and lift coefficients are depicted in fig. 5.14. Again, the time step in fig. 5.12 - 5.13 was chosen such that the lift value in fig. 5.14 is maximal. The results confirm the conclusion from the steady computations that increasing p increases the accuracy significantly in the Finite Cell Method. Despite this, an optimal agreement was not achieved with the given discretization. In the computations, a clear improvement of the results can be achieved by choosing a smaller time step. As an example, the drag and lift forces for $p = 4$ agree very well with the reference computations for a time step of $\Delta t = 0.001$. The origin of this error is therefore not yet clear but can be attributed either to the time integration or the VMS stabiliztion, which also introduces a time step dependency.



(a) $p = 1$, time step 2018

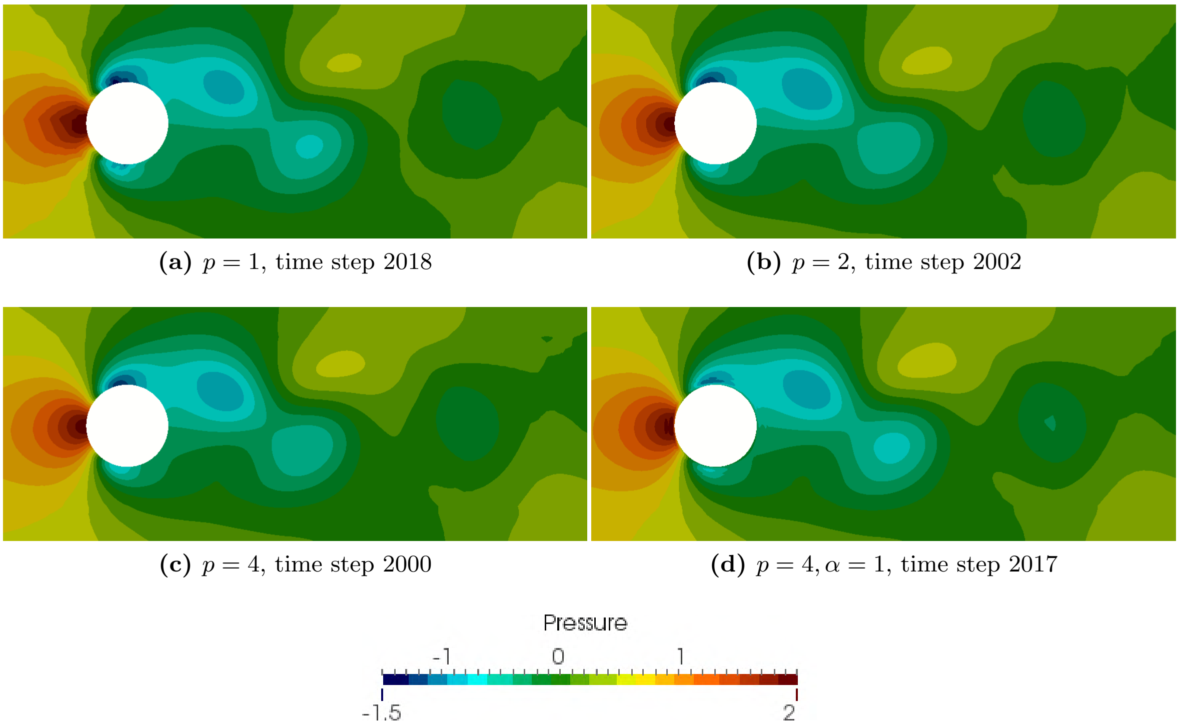
(b) $p = 2$, time step 2002

(c) $p = 4$, time step 2000

(d) $p = 4, \alpha = 1$, time step 2017

Figure 5.12: Velocity magnitude contours for FCM discretizations of the flow around a cylinder.

Figures 5.15 - 5.16 show the velocity and pressure solution fields around and inside the fictitious domain for $p = 4$. The smooth extension enables optimal convergence in the Finite Cell Method (*cf.* Dauge et al. [2015]) only if the influence originating from the fictitious domain is eliminated. As a consequence, the solution in fig. 5.15b and fig. 5.16b exhibits oscillations around the interface. While the influence on the large scale characteristics on the



(a) $p = 1$, time step 2018

(b) $p = 2$, time step 2002

(c) $p = 4$, time step 2000

(d) $p = 4, \alpha = 1$, time step 2017

Figure 5.13: Pressure contours for FCM discretizations of the flow around a cylinder.

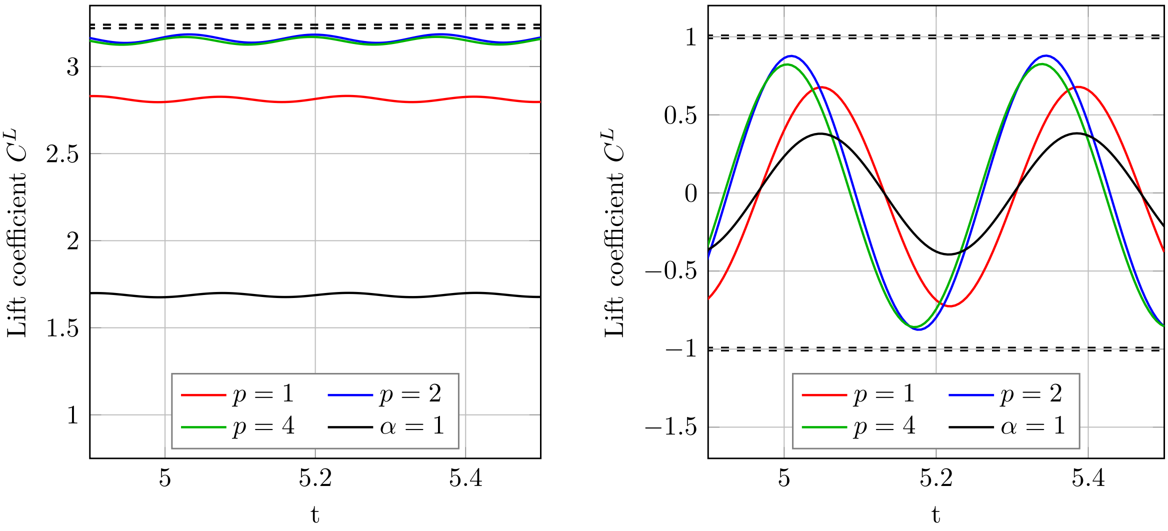
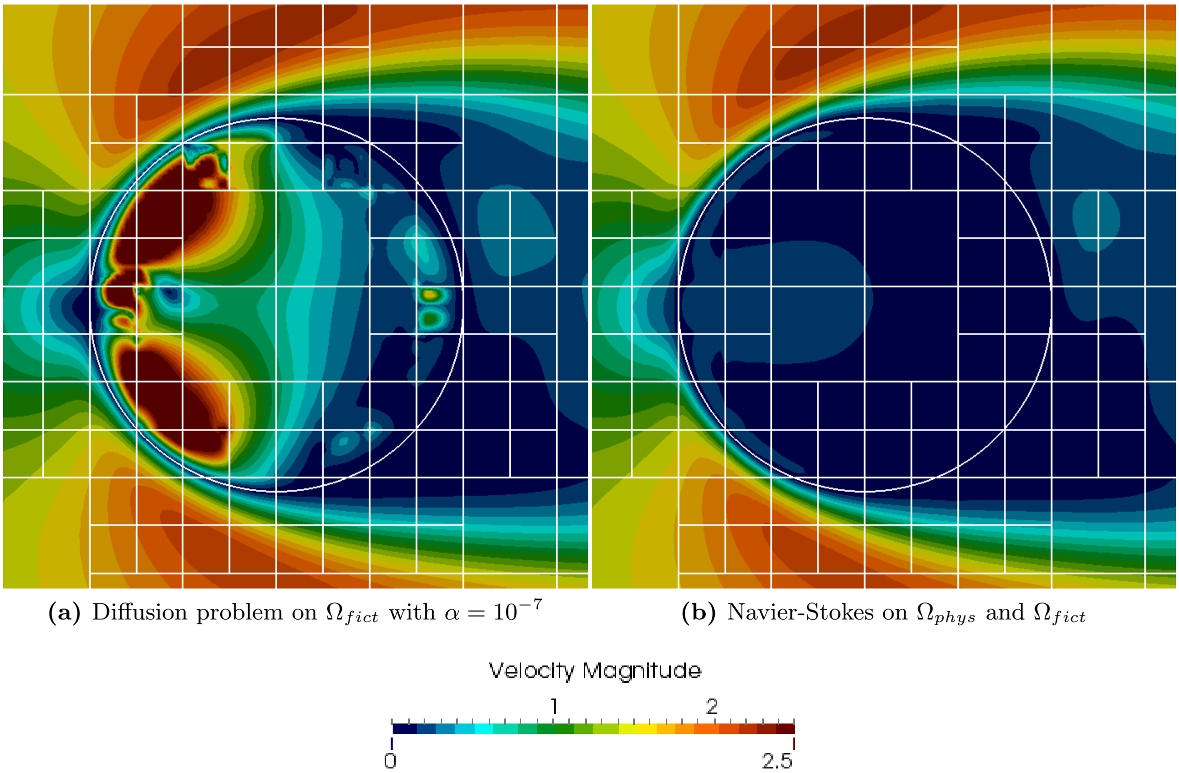


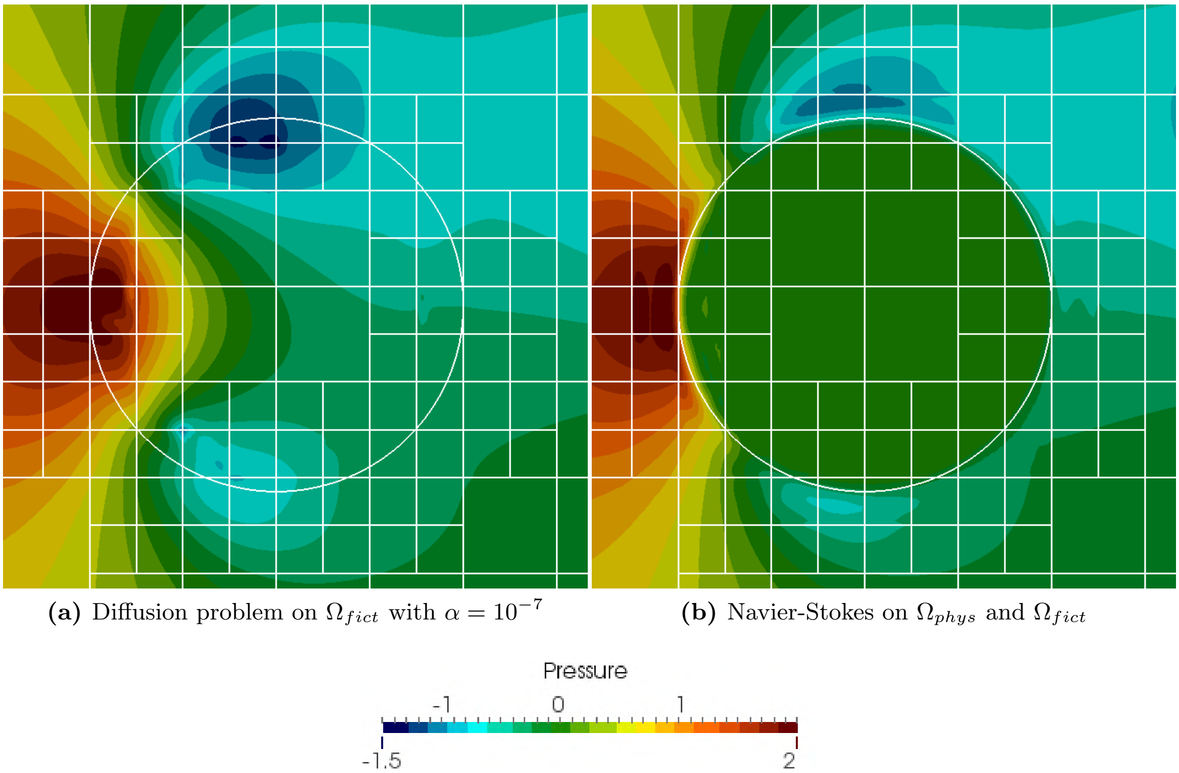
Figure 5.14: Drag and lift coefficients for different FCM discretizations.

rest of the domain is small, the effect on the solution and its derivatives on the boundary can not be neglected. As a result the drag and lift forces in both, steady (*cf.* table 5.4) and unsteady computations (*cf.* fig. 5.14) are underestimated. As these quantities are of great interest in practice, future investigations need to aim also at finding a particular ϵ that is small enough to accurately compute forces. This includes in particular the problem that a modified equation on Ω_{fict} introduces a dependency of ϵ on problem specific quantities, as mentioned in section 4.3.


(a) Diffusion problem on Ω_{fict} with $\alpha = 10^{-7}$

(b) Navier-Stokes on Ω_{phys} and Ω_{fict}

Figure 5.15: Comparison of velocity fields for different models in Ω_{fict} with $p = 4$.


(a) Diffusion problem on Ω_{fict} with $\alpha = 10^{-7}$

(b) Navier-Stokes on Ω_{phys} and Ω_{fict}

Figure 5.16: Comparison of pressure fields for different models in Ω_{fict} with $p = 4$.

Chapter 6

Conclusion and Outlook

In the first part of this thesis, it has been demonstrated that the multi-level *hp*-method represents a flexible tool to resolve sharp solution characteristics that often arise in the field of fluid mechanics. The application to advection-diffusion problems shows that spurious oscillations at boundary layers can be suppressed by refining the mesh locally. Moreover, the combination of high-order methods with stabilization schemes such as the Variational Multiscale Method was addressed. A simple expression for choosing the stabilization parameter τ for arbitrary polynomial degrees has been found in a simple one-dimensional setting that needs yet to be validated on more complex scenarios.

In the second part, the residual-based Variational Multiscale method was extended towards the Finite Cell method. The only major modification required is the adaption of the integration scheme towards a discontinuous integrand. However, an improvement in computation time and numerical stability can be achieved by choosing a different weak form in the fictitious domain. The results presented in chapter 5 show that this approach delivers good results in combination with the blended partitioning of integration cells. Although drag and lift coefficients in the flow around a cylinder example deviate from the reference values to a significant degree, this problem is present also in computations using a conforming mesh. Therefore, the source of this error can not be related purely to the Finite Cell discretization. An additional investigation in this direction is required before more complex scenarios can be addressed. Furthermore, it has been shown that the combination with a multi-level *hp*-refinement provides the required flexibility to adapt Finite Cell discretizations locally in regions of interest.

On top of this, several tasks remain open. This includes, in particular, extending the weak imposition of boundary conditions as suggested in chapter 4.2. The possibility to incorporate a wall model is very attractive as it transfers to immersed boundaries in a straightforward manner and therefore greatly extends the method's range of applicability. Furthermore, the quality of different definitions of τ for the stabilized Navier-Stokes system needs to be investigated. It is not yet clear, how sensible the method is with respect to the selection of τ and under which conditions the influence is significant. As mentioned in section 4.4, it is possible to decrease the overall computation time by extending the time integration to use an adaptive step size. An automatic selection of Δt would also go well with the philosophy of automatizing the solution procedure and therefore reducing the effort for the engineer in practice.

Besides the discretization method, it is also of interest to address the solution time of the linear system of equations. As direct solvers tend to require too much memory on larger problems, iterative solvers need to be used. As the number of iterations is strongly dependent on the conditioning of the matrices, the use of preconditioners is indispensable. One possibility represents the element-by-element inverse preconditioner introduced in Bazilevs et al. [2010]. The concept is to extract element matrices from the final system and assemble the inverse into the preconditioner. Despite its attractive simplicity concerning the implementation, it is not clear how the performance changes for high-order elements. A similar approach represents the SIPIC preconditioner, introduced by de Prenter et al. [2017], where the bad conditioning is attributed to a small support and a linear dependence of basis functions in badly cut cells.

Last, but not least, the tests performed in chapters 3 and 5 need to be complemented by more benchmarks and real world examples. In particular, studying the influence of the integration scheme used to capture the discontinuity in the Finite Cell Method is important to guide the future development of the method.

Appendix A

The best approximation property and Variational Multiscale method

The question, how the optimality of the Bubnov-Galerkin method relates to the Variational Multiscale method the Galerkin orthogonality is addressed here. Following the introduction of chapter 2, a symmetric weak form of the type

$$B(w,u) = F(w) \qquad \forall w \in \mathcal{W} \tag{A.1}$$

is considered, where the solution $u \in \mathcal{W}$ was already split, such that the right hand side F contains the Dirichlet boundary contribution $B(w,\overline{g})$. Moreover, $B(w,u)$ defines an inner product $\langle \cdot, \cdot \rangle_B$ due to the symmetry in u and w. The discrete Bubnov-Galerkin solution is obtained by choosing $\mathcal{W}^h \subset \mathcal{W}$ and solving

$$B(w^h,u^h) = F(w^h) \qquad \forall w^h \in \mathcal{W}^h. \tag{A.2}$$

The exact solution u also satisfies eq. (A.2). Subtraction of both equations yields

$$B(w^h, u^h - u) = 0 \qquad \forall w^h \in \mathcal{W}^h,$$

or using the error $e = u^h - u$:

$$\langle w^h, e \rangle_B = 0 \qquad \forall w^h \in \mathcal{W}^h. \tag{A.3}$$

In other words, the error is B-orthogonal to the weighting space, which was chosen to be also the solution space, because of this property. To show, that the solution is an optimal fit, we consider any $q^h \in \mathcal{W}^h$ and proceed as follows:

$$\begin{aligned} B(e+q^h, e+q^h) &= B(e,e) + B(e,q^h) + B(q,e) + B(q^h,q^h) \\ &= B(e,e) + \underbrace{2B(q^h,e)}_{=0, \text{ by eq. (A.3)}} + B(q^h,q^h). \end{aligned}$$

As $B(q^h,q^h) \geq 0$, we get

$$B(e,e) \leq B(e+q^h, e+q^h) = B(u-u^h+q^h, u-u^h+q^h).$$

Because q is arbitrary, $v^h = u^h + q^h$ is arbitrary and therefore

$$B(e,e) \leq B(u - v^h, u - v^h) \qquad \forall v^h \in \mathcal{W}^h,$$

the solution u^h delivers the best error possible with respect to $\|e\|_B = \sqrt{B(e,e)}$:

$$u^h = \inf_{v^h \in \mathcal{W}^h} \|u - v^h\|_B. \tag{A.4}$$

Equation A.4 is precisely the definition of an orthogonal projection onto $\mathcal{W}^h$ in $\| \cdot \|_B$.

Now, we construct a Variational Multiscale method and choose this projector. The coarse scale equation yields

$$\langle \overline{w}, \overline{u} \rangle + \langle \overline{w}, u' \rangle = \langle \overline{w}, f \rangle. \tag{A.5}$$

We know, that already the equation $\langle \overline{w}, \overline{u} \rangle = \langle \overline{w}, f \rangle$ gives an optimal result in $\| \cdot \|_B$. From this, we can conclude that $\langle \overline{w}, u' \rangle = \langle \overline{w}, e \rangle = 0$ as it does not contribute to the weak form. Therefore, from a symmetric operator follows the orthogonality of coarse and unresolved scales in the B-norm, where the unresolved scales are exactly the error $u^h - u$. Furthermore, the Variational Multiscale method provides a theoretical framework to optimize Finite Element methods with respect to a different norm.

List of Figures

List of Tables

Bibliography

Arnold, D. N., Babuška, I., and Osborn, J. (1984). Finite element methods: Principles for their selection. *Computer Methods in Applied Mechanics and Engineering*, 45(1):57–96.

Babuska, I., Szabo, B., and Katz, I. (1981). The p-Version of the Finite Element Method. *SIAM Journal on Numerical Analysis*, 18(3):515–545.

Bazilevs, Y., Calo, V. M., Cottrell, J. A., Hughes, T. J. R., Reali, A., and Scovazzi, G. (2007a). Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1–4):173–201.

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

Bazilevs, Y., Michler, C., Calo, V. M., and Hughes, T. J. R. (2007b). Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196(49–52):4853–4862.

Bazilevs, Y., Michler, C., Calo, V. M., and Hughes, T. J. R. (2010). Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199(13-16):780–790.

Bog, T., Zander, N., Kollmannsberger, S., and Rank, E. (2015). Normal contact with high order finite elements and a fictitious contact material. *Computers & Mathematics with Applications*, 70(7):1370–1390.

Brooks, A. N. and Hughes, T. J. R. (1982). Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1–3):199–259.

Cai, Q. (2013). *Finite Cell Method for Transport Problems in Porous Media.* PhD thesis, Technische Universität München, Munich.

Cai, Q., Kollmannsberger, S., Sala-Lardies, E., Huerta, A., and Rank, E. (2014). On the natural stabilization of convection dominated problems using high order Bubnov–Galerkin finite elements. *Computers & Mathematics with Applications*, 66(12):2545–2558.

Codina, R. (2000). On stabilized finite element methods for linear systems of convection–diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 188(1-3):61–82.

Codina, R., Principe, J., Guasch, O., and Badia, S. (2007). Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 196(21–24):2413–2430.

Cottrell, J. A., Hughes, T. J. R., and Bazilevs, Y. (2009). *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley Publishing, 1st edition.

Dauge, M., Düster, A., and Rank, E. (2015). Theoretical and Numerical Investigation of the Finite Cell Method. *J. Sci. Comput.*, 65(3):1039–1064.

de Prenter, F., Verhoosel, C., van Zwieten, G., and van Brummelen, E. (2017). Condition number analysis and preconditioning for the finite cell method. *Computer Methods in Applied Mechanics and Engineering*, 316:297–327.

Demkowicz, L., Oden, J. T., Rachowicz, W., and Hardy, O. (1989). Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering*, 77(1–2):79–112.

Donéa, J. and Huerta, A. (2003). *Finite element methods for flow problems*. Wiley, Chichester, New York.

Durán, R. G. (2005). Galerkin Approximations and Finite Element Methods - lecture notes. Departamento de Matemática, Universidad de Buenos Aires.

Elhaddad, M., Zander, N., Kollmannsberger, S., Shadavakhsh, A., Nübel, V., and Rank, E. (2015). Finite Cell Method: High-Order Structural Dynamics for Complex Geometries. *International Journal of Structural Stability and Dynamics*, 15(7):1540018.

Franca, L. P., Frey, S. L., and Hughes, T. J. R. (1992). Stabilized finite element methods: I. Application to the advective-diffusive model. *Computer Methods in Applied Mechanics and Engineering*, 95(2):253–276.

Ghia, U., Ghia, K. N., and Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411.

Gui, W. and Babuška, I. (1986a). The h, p and h-p versions of the finite element method in 1 dimension Part I: The error analysis of the p-version. *Numerische Mathematik*, 49(6):577–612.

Gui, W. and Babuška, I. (1986b). The h, p and h-p versions of the finite element method in 1 dimension Part II: The error analysis of the h-and h-p versions. *Numerische Mathematik*, 49(6):613–657.

Hughes, T. J. R. (1995). Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127(1):387–401.

Hughes, T. J. R. (2000). *The finite element method: linear static and dynamic finite element analysis*. Dover Publications, Mineola, NY.

Hughes, T. J. R., Feijóo, G. R., Mazzei, L., and Quincy, J.-B. (1998). The variational multiscale method—a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1):3–24.

Hughes, T. J. R., Mazzei, L., and Jansen, K. E. (2000). Large Eddy Simulation and the variational multiscale method. *Computing and Visualization in Science*, 3(1-2):47–59.

Hughes, T. J. R. and Sangalli, G. (2007). Variational Multiscale Analysis: the Fine-scale Green's Function, Projection, Optimization, Localization, and Stabilized Methods. *SIAM Journal on Numerical Analysis*, 45(2):539–557.

Hughes, T. J. R., Scovazzi, G., and Franca, L. P. (2004). Multiscale and Stabilized Methods. In *Encyclopedia of Computational Mechanics*. John Wiley & Sons, Ltd.

Jansen, K. E., Whiting, C. H., and Hulbert, G. M. (2000). A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190(3–4):305–319.

Joulaian, M., Hubrich, S., and Düster, A. (2016). Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Computational Mechanics*, 57(6):979–999.

Kamensky, D., Hsu, M.-C., Schillinger, D., Evans, J. A., Aggarwal, A., Bazilevs, Y., Sacks, M. S., and Hughes, T. J. R. (2015). An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005–1053.

Kudela, L., Zander, N., Bog, T., Kollmannsberger, S., and Rank, E. (2015). Efficient and accurate numerical quadrature for immersed boundary methods. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):1–22.

Kudela, L., Zander, N., Kollmannsberger, S., and Rank, E. (2016). Smart octrees: Accurately integrating discontinuous functions in 3d. *Computer Methods in Applied Mechanics and Engineering*, 306:406–426.

Nitsche, J. (1971). Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15.

Pardo, D., Paszynski, M., Collier, N., Alvarez, J., Dalcin, L., and Calo, V. M. (2012). A survey on direct solvers for Galerkin methods. *SeMA Journal*, 57(1):107–134.

Parvizian, J., Düster, A., and Rank, E. (2007). Finite cell method. *Computational Mechanics*, 41(1):121–133.

Parvizian, J., Düster, A., and Rank, E. (2011). Topology optimization using the finite cell method. *Optimization and Engineering*, 13(1):57–78.

Quarteroni, A. and Valli, A. (1994). *Numerical Approximation of Partial Differential Equations*, volume 23 of *Springer Series in Computational Mathematics*. Springer, Berlin, Heidelberg.

Rajani, B. N., Kandasamy, A., and Majumdar, S. (2009). Numerical simulation of laminar flow past a circular cylinder. *Applied Mathematical Modelling*, 33(3):1228–1247.

Rank, E. (1992). Adaptive remeshing and h-p domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 101(1–3):299–313.

Schillinger, D., Ruess, M., Zander, N., Bazilevs, Y., Düster, A., and Rank, E. (2012). Small and large deformation analysis with the p- and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478.

Schwab, C. and Suri, M. (1996). The p and hp versions of the finite element method for problems with boundary layers. *Mathematics of Computation*, 65(216):1403–1429.

Schäfer, M. and Turek, S. (1996). Benchmark Computations of Laminar Flow Around a Cylinder. In *Flow Simulation with High-Performance Computers II*, number 48 in Notes on Numerical Fluid Mechanics (NNFM), pages 547–566. Vieweg & Teubner Verlag.

Shakib, F., Hughes, T. J. R., and Johan, Z. (1991). A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89(1-3):141–219.

Szabó, B. and Babuška, I. (1991). *Finite element analysis*. John Wiley & Sons, New York.

Tezduyar, T. E. and Osawa, Y. (2000). Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190(3–4):411–430.

Vignal, P., Collier, N., Dalcin, L., Brown, D. L., and Calo, V. M. (2017). An energy-stable time-integrator for phase-field models. *Computer Methods in Applied Mechanics and Engineering*, 316:1179–1214.

Xu, F., Schillinger, D., Kamensky, D., Varduhn, V., Wang, C., and Hsu, M.-C. (2015). The tetrahedral finite cell method for fluids: Immersogeometric analysis of turbulent flow around complex geometries. *Computers & Fluids*, in press.

Zander, N. (2016). *Multi-level hp-FEM: dynamically changing high-order mesh refinement with arbitrary hanging nodes*. in preparation, Technische Universität München, Munich.

Zander, N., Bog, T., Elhaddad, M., Frischmann, F., Kollmannsberger, S., and Rank, E. (2016). The multi-level hp-method for three-dimensional problems: Dynamically changing high-order mesh refinement with arbitrary hanging nodes. *Computer Methods in Applied Mechanics and Engineering*, 310:252–277.

Zander, N., Bog, T., Kollmannsberger, S., Schillinger, D., and Rank, E. (2015). Multi-level hp-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, 55(3):499–517.

Zienkiewicz, O. C., Taylor, R. L., and Nithiarasu, P. (2005). *The Finite Element Method for Fluid Dynamics*. Butterworth-Heinemann, Oxford, 6th edition.