Inaugural-Dissertation

zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von
Diplom-Mathematiker Michael Schmich
aus Mannheim

Tag der mündlichen Prüfung: 15. Dezember 2009

Adaptive Finite Element Methods for Computing Nonstationary Incompressible Flows

Gutachter: Prof. Dr. Dr. h. c. Rolf Rannacher

Prof. Dr. Peter Bastian

Abstract

Subject of this work is the development of numerical methods for efficiently solving nonstationary incompressible flow problems. In contrast to stationary flow problems, here errors due to discretization in time and space occur. Furthermore, especially three-dimensional simulations lead to huge computational costs. Thus, adaptive discretization methods have to be used in order to reduce the computational costs while still maintaining a certain accuracy.

The main focus of this thesis is the development of an a posteriori error estimator which is computable and able to assess both discretization errors separately. Thereby, the error is measured in an arbitrary quantity of interest (such as the drag-coefficient, for example) because measuring errors in global norms is often of minor importance in practical applications. The basis for this is a finite element discretization in time and space. The techniques presented here also provide local error indicators which are used to adaptively refine the temporal and spatial discretization. A key ingredient in setting up an efficient discretization method is balancing the error contributions due to temporal and spatial discretization. To this end, a quantitative assessment of the individual discretization errors is required.

The described methods are validated by several numerical tests. These also include established Navier-Stokes benchmarks as well as a two-phase flow problem with complex three-dimensional geometry.

Zusammenfassung

Gegenstand dieser Arbeit ist die Entwicklung numerischer Verfahren zur effizienten Lösung instationärer inkompressibler Strömungsprobleme. Im Gegensatz zu stationären Strömungsproblemen entstehen hier Diskretisierungsfehler sowohl durch die Diskretisierung in der Zeit als auch durch die Diskretisierung im Ort. Außerdem führen insbesondere dreidimensionale Simulationen zu einem hohen Rechenaufwand. Dies erfordert die Verwendung adaptiver Diskretisierungen, um den Rechenaufwand zu reduzieren und gleichzeitig eine gewisse Genauigkeit beizubehalten.

Der Schwerpunkt dieser Dissertation besteht in der Entwicklung eines auswertbaren a posteriori-Fehlerschätzers, der die getrennte Erfassung beider Diskretisierungsfehler ermöglicht. Der Fehler wird dabei in einer beliebigen Größe (wie etwa dem Widerstandsbeiwert) gemessen, da Fehlerangaben in globalen Normen in praktischen Anwendungen meist von geringerer Bedeutung sind. Grundlage dafür ist die Verwendung von Finite-Elemente-Diskretisierungen in Ort und Zeit. Die vorgestellten Techniken liefern außerdem lokale Fehlerindikatoren, die zur adaptiven Verfeinerung der Zeit- bzw. Ortsdiskretisierung verwendet werden. Zur Gestaltung eines effizienten Diskretisierungsverfahren ist die Balancierung der Fehlerbeiträge durch Zeit- bzw. Ortsdiskretisierung nötig, was eine zuverlässige quantitative Erfassung der einzelnen Diskretisierungsfehler erfordert.

Die präsentierten Methoden werden anhand verschiedener numerischer Tests validiert. Dabei werden auch etablierte Navier-Stokes-Benchmarks sowie ein Zweiphasenströmungsproblem mit komplexer, dreidimensionaler Geometrie betrachtet.

Contents

1 Introduction

This work is devoted to the development of efficient discretization techniques for numerically solving nonstationary incompressible flow problems. Since in contrast to stationary problems we have to deal with the discretization in time as well as in space, one of the main topics in setting up such an efficient algorithm is to obtain quantitative information about the temporal and spatial discretization error. This is a key ingredient because within an efficient algorithm one has to decide which discretization has to be refined to reduce the discretization error in the most efficient way.

The computational costs of numerically solving nonstationary flow problems are comparatively high due to the complex structure of such problems, especially when dealing with nonstationary three-dimensional flow problems. Thus, it is crucial to apply adaptive refinement techniques to reduce the size of the approximative problems without reducing the accuracy of the approximation.

Adaptive methods are widely used in the context of finite element discretizations of partial differential equations, see, for example, Verfürth [102] or Eriksson, Estep, Hansbo, and Johnson [41] for an overview. In Bänsch [6], an adaptive strategy for the nonstationary Navier-Stokes equations is developed which is based on a posteriori error estimates in the *energy-norm*.

However, error estimation with respect to global norms such as the *energy-norm* sometimes is not very efficient since in flow problems one is often only interested in a specific functional value of the solution, the so-called *quantity of interest*. Hence, the goal of the numerical simulation of a flow problem is the efficient computation of this single number. This quantity might, for instance, be the mean drag- or lift-coefficient of an obstacle which is surrounded by the fluid. In this case, the efficiency of an algorithm for numerically computing this quantity has to be measured by means of the reduction of the discretization error in the quantity of interest rather than in global norms since the latter usually do not provide useful bounds for the error in the quantity of interest.

The basis for such a posteriori error estimation was given in Becker and Rannacher [13]. Besides the simulation of elliptic problems, this result has been successfully applied to parameter estimation (Becker and Vexler [14]), optimal control problems (Becker [8]), stationary flow problems (Richter [93]), chemically reacting flows (Braack [17]), and many others. Considering time-dependent problems, Hartmann [64] derived a posteriori error estimators for the heat equation. In Schmich and Vexler [96], this approach was extended to general nonlinear parabolic problems. Meidner [78] developed efficient adaptive algorithms for optimal control problems governed by nonlinear parabolic problems. An application

of the abstract theory to fluid-structure interaction problems can be found in Dunne [38] who however only considered spatial adaptivity.

This work extends the methodology developed in Schmich and Vexler [96] to nonstationary flow problems allowing for the simultaneous adaptation of the temporal and spatial discretization. Furthermore, we will derive a posteriori error estimators which quantitatively assess the discretization error measured in the quantity of interest and separate the influence of the temporal and the spatial discretization. This separation will allow us to set up an efficient algorithm for the adaptive refinement of the temporal and the spatial discretization. Applying the approach derived in this thesis, we are able to compute the mean drag-coefficient in a three-dimensional time-dependent benchmark configuration from Schäfer and Turek [95] up to an accuracy of a few percent on a standard personal computer.

The key to rigorous a posteriori error estimation is a coupled variational formulation of the underlying equations. It allows to apply Galerkin finite element methods not only for the discretization in space, but also for the discretization in time. The use of space-time finite element discretizations enables the application of residual based a posteriori error estimation. Discontinuous Galerkin methods for the discretization of ordinary differential equations have been used by Delfour, Hager, and Trochu [37] whereas Estep and French [49] applied continuous Galerkin methods to ordinary differential equations. In the context of parabolic problems, the works of Eriksson and Johnson [43, 44, 45, 46], Eriksson, Johnson, and Larsson [47], Eriksson, Johnson, and Thomée [48], and Thomée [100] as well as Akrivis, Makridakis, and Nochetto [1] and Aziz and Monk [4] have to be mentioned. Space-time Galerkin methods have already been applied successfully to the simulation of incompressible flows, see, for example, Mittal, Ratner, Hastreiter, and Tezduyar [79], Mittal and Tezduyar [80], Behr and Tezduyar [15], or N'dri, Garon, and Fortin [84] as well as Hoffman [68] (referred to as *General Galerkin G2*). While the first references do not consider adaptivity, Hoffman [68] also develops an adaptive algorithm for nonstationary flow problems based on a posteriori error estimation. However, he does not separate the temporal and spatial discretization error. Instead, the temporal refinement is linked to the spatial refinement.

The novelty of the approach presented in this thesis is the development of a posteriori error estimators for nonstationary incompressible flow problems which separate and quantitatively assess the temporal and spatial discretization error. This allows for the construction of efficient discretization methods because the temporal and spatial discretization error can be balanced.

To be most efficient in capturing the dynamics of a nonstationary flow problem, it seems desirable to use so-called *dynamic meshes* for the discretization in space. That is, one uses possibly different meshes for different time points. Thus one can efficiently resolve and track fronts marching through the domain, for example. In the context of parabolic partial differential equations one is easily led to the fully discrete problem (that is discretized in time and space) by taking the variational formulation of the semi-discrete problem (that is discretized in time, but still continuous in space) and simply restricting the corresponding function spaces to the ones involving the finite-dimensional fully discrete spaces. Proceeding in a similar way for incompressible flow problems leads to appropriate approximations of

the velocity while the approximation of the pressure deteriorates. The numerical analysis of the phenomenon as well as a theoretical investigation is a main subject of this thesis.

In the sequel, we give a summary of the remaining chapters of this thesis.

Theoretical Results

In Chapter 2, we first give an overview of the basic notation used throughout this thesis. Then, we formulate the incompressible Navier-Stokes equations which are based on the physical conservation laws of mass and momentum. Finally, we present the variational formulation of the incompressible Navier-Stokes equations which forms the basis for the Galerkin finite element discretization in space and time.

Space-Time Finite Element Discretization

Chapter 3 presents the discretization of the incompressible Navier-Stokes equations in space and time. For the temporal discretization we consider continuous or discontinuous Galerkin finite element methods. The spatial discretization on dynamic meshes is based on continuous Galerkin finite element methods. Since we use equal order polynomials for the velocity and the pressure, the resulting fully discrete formulations need to be stabilized. This is done by means of the so-called *local projection stabilization*, see Becker and Braack [10, 11]. We conclude this chapter by giving some information on implementational aspects.

A Posteriori Error Estimation

One of the main topics of this thesis is the derivation of an a posteriori error estimator which assesses both the temporal and the spatial discretization error with respect to a given quantity of interest. This is based on an abstract result of Becker and Rannacher [12, 13]. We also present an adaptive algorithm which uses the a posteriori error estimators for successive refinement of the temporal and spatial discretization. This algorithm leads to equilibrated spatial and temporal discretization errors which is crucial for the efficiency of the solution strategy. However, this requires quantitative error estimators. In contrast to the error estimators developed in this thesis, simple heuristic error indicators based on smoothness of the solution do in general not fulfill this requirement. The same holds true for error estimators involving unknown interpolation or stability constants. We close this chapter with a numerical validation of the developed quantitative error estimators and a numerical comparison with heuristic error indicators.

Issues on Dynamic Meshes

This chapter is dedicated to the numerical and theoretical investigation of a phenomenon that occurs when solving the incompressible Navier-Stokes equations on dynamically changing meshes. We present a numerical simulation of a two-dimensional benchmark

configuration from Schäfer and Turek [95] in which we aim at computing the (mean) lift-coefficient. We note oscillations in the lift-coefficient when switching the spatial mesh which can be shown to originate from a disturbed approximation of the pressure. We numerically investigate two model problems involving the Stokes equations in which the same effect can be observed. To exclude stabilization artifacts, we introduce the inf-sup-stable Taylor-Hood element, see Hood and Taylor [69]. For completeness, the second-order fractional-step-θ scheme (Müller-Urbaniak [81]) for the discretization in time is also considered which is a popular method in computational fluid dynamics. The final section of this chapter gives a theoretical investigation of this phenomenon.

All computations in this thesis have been done with the software package Gascoigne [56] except for the computations with the Taylor-Hood element which have been carried out using the software package deal.II [36]. For visualizing the numerical solutions, the visualization tool VisIt [103] was used.

Applications

In Chapter 6, we apply the developed a posteriori error estimators and adaptive refinement techniques to some applications.

The first examples are the two- and three-dimensional time-dependent benchmark configurations from Schäfer and Turek [95] in which we compute mean drag- and lift-coefficients.

The last example simulates the filling process of a so-called *lab-on-a-chip*, like the microdiagnosis chip Lilliput®, see Figure 1.1. This reaction platform, which is of size $20\,\text{mm} \times 37\,\text{mm} \times 3\,\text{mm}$, consists of 96 cavities like ordinary platforms, but uses less than $\frac{1}{100}$ of the usual volume, namely $1.8\,\mu\text{l}$.

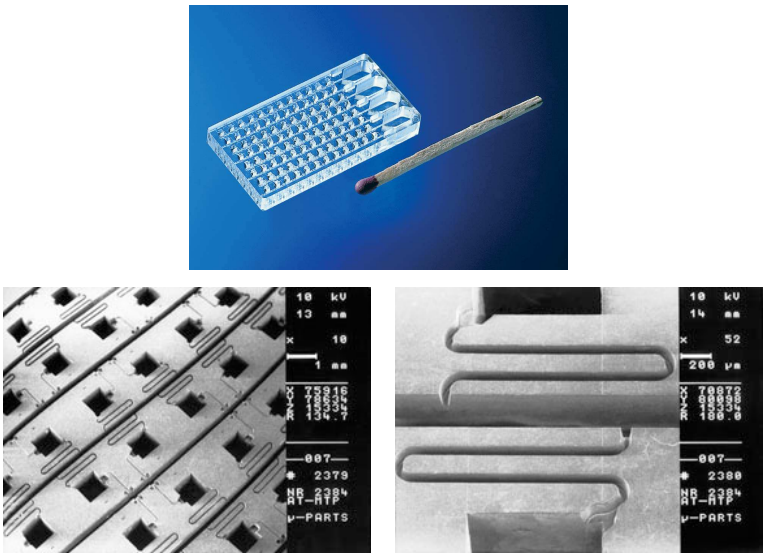


Figure 1.1. Microdiagnosis chip Lilliput® (photos by courtesy of Boehringer Ingelheim microParts GmbH)

At the beginning of the simulation, the whole chip is filled with gas. Then the liquid enters the chip via the larger channels, is distributed through the thin channels, and finally reaches one of the reaction chambers. This process is modeled as a two-phase flow with one liquid phase and one gaseous phase.

Before showing numerical results of the simulation of the filling process, we first present the governing equations as well as the fully discrete model including a level set approach for separating the two phases.

Conclusions and Outlook

In the final chapter of this thesis, we summarize the presented results and discuss possible extensions and future work.

2 Theoretical Results

In this chapter, we derive the fundamental equations describing the motion of fluids and discuss some theoretical results.

In Section 2.1, we present some basic notation used throughout this thesis. Section 2.2 is dedicated to the formulation of the *incompressible Navier-Stokes equations*. We also show their variational formulation which is the basis for the discretization in space and time presented in Chapter 3.

2.1 Basic notations

Throughout this thesis, $\Omega \subseteq \mathbb{R}^d$, $d \in \{ 2, 3 \}$, denotes a bounded domain with boundary $\partial\Omega$. If no further information is given, we will generally assume the boundary to be Lipschitzian, see Grisvard [60] for a precise definition. The outer unit normal vector to $\partial\Omega$ is denoted by $\boldsymbol{n}$.

The standard Lebesgue space $L^p(\Omega)$, $1 \leq p \leq \infty$, consists of measurable functions u which are Lebesgue-integrable to the p-th power. This is a Banach space with the norm

$$\begin{aligned} \|u\|_{L^p(\Omega)} &\coloneqq \left(\int_\Omega |u(\boldsymbol{x})|^p \, \mathrm{d}\boldsymbol{x} \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty, \\ \|u\|_{L^\infty(\Omega)} &\coloneqq \operatorname*{ess\,sup}_{\boldsymbol{x} \in \Omega} |u(\boldsymbol{x})| \,. \end{aligned}$$

For $p = 2$, $L^2(\Omega)$ is a Hilbert space with the inner product

$$(u, v)_{L^2(\Omega)} \coloneqq \int_\Omega u(\boldsymbol{x}) v(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}.$$

The Sobolev space $W^{m,p}(\Omega)$, $m \in \mathbb{N}$, $1 \leq p \leq \infty$, is the space of functions in $L^p(\Omega)$ whose distributional derivatives of order up to m belong to $L^p(\Omega)$. It is a Banach space with the norm

$$\begin{aligned} \|u\|_{W^{m,p}(\Omega)} &\coloneqq \left(\sum_{|\alpha| \leq m} \|\partial^\alpha u\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty, \\ \|u\|_{W^{m,\infty}(\Omega)} &\coloneqq \max_{|\alpha| \leq m} \|\partial^\alpha u\|_{L^\infty(\Omega)} \,. \end{aligned}$$

Here, $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$ denotes a multi-index. We set

$$|\alpha| \coloneqq \sum_{j=1}^{d} \alpha_j, \quad \partial^\alpha \coloneqq \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}.$$

For $p = 2$, $H^m(\Omega) \coloneqq W^{m,2}(\Omega)$ is a Hilbert space with the inner product

$$(u, v)_{H^m(\Omega)} \coloneqq \sum_{|\alpha| \leq m} (\partial^\alpha u, \partial^\alpha v)_{L^2(\Omega)}.$$

Let us also introduce the semi-norms

$$|u|_{W^{m,p}(\Omega)} \coloneqq \left(\sum_{|\alpha|=m} \|\partial^\alpha u\|_{L^p(\Omega)}^p \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty,$$
$$|u|_{W^{m,\infty}(\Omega)} \coloneqq \max_{|\alpha|=m} \|\partial^\alpha u\|_{L^\infty(\Omega)}.$$

For sake of brevity, we use the following shorter notation:

$$(u, v) \coloneqq (u, v)_{L^2(\Omega)}, \quad \|u\| \coloneqq \|u\|_{L^2(\Omega)}.$$

The corresponding spaces of d-dimensional vector functions are denoted by $L^p(\Omega)^d$, $W^{m,p}(\Omega)^d$, and $H^m(\Omega)^d$. They are equipped with the usual product norm. The norms and inner products on these spaces are denoted in the same way as for scalar functions.

The space $C^k(\Omega)$, $k \in \mathbb{N}$, denotes the space of functions whose derivatives up to order k are continuous on Ω. We set

$$C^\infty(\Omega) \coloneqq \bigcap_{k \in \mathbb{N}} C^k(\Omega).$$

For simplicity, we set $C(\Omega) \coloneqq C^0(\Omega)$.

The space $C^k(\overline{\Omega})$ consists of all functions from $C^k(\Omega)$ whose derivatives up to order k possess continuous extensions onto $\overline{\Omega}$. It is a Banach space with the norm

$$\|f\|_{C^k(\overline{\Omega})} \coloneqq \max_{|\alpha| \leq k} \sup_{\boldsymbol{x} \in \overline{\Omega}} \left| \partial^\alpha f(\boldsymbol{x}) \right|.$$

The space $C_0^\infty(\Omega) \subseteq C^\infty(\Omega)$ is the subspace of functions which are non-zero only in a compact subset of Ω. The closure of $C_0^\infty(\Omega)$ in $W^{m,p}(\Omega)$ is denoted by $W_0^{m,p}(\Omega)$ or $H_0^m(\Omega)$ if $p = 2$.

The dual space of $H_0^m(\Omega)$ is denoted by $H^{-m}(\Omega)$. It is a Banach space with its norm

$$\|u\|_{H^{-m}(\Omega)} \coloneqq \sup_{\varphi \in H_0^m(\Omega)} \frac{\langle u, \varphi \rangle}{|\varphi|_{H^m(\Omega)}},$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing between $H^{-m}(\Omega)$ and $H_0^m(\Omega)$.

When dealing with the incompressible Navier-Stokes equations (see Section 2.2), the following function spaces are of fundamental importance:

$$\begin{aligned} \mathcal{V} &\coloneqq \left\{ \boldsymbol{v} \in C_0^\infty(\Omega)^d \;\middle|\; \nabla \cdot \boldsymbol{v} = 0 \right\}, \\ V &\coloneqq \overline{\mathcal{V}}^{\,\|\cdot\|_{H_0^1(\Omega)}}, \\ H &\coloneqq \overline{\mathcal{V}}^{\,\|\cdot\|_{L^2(\Omega)}}. \end{aligned} \tag{2.1}$$

The spaces H and V are Hilbert spaces with the inner products induced by $L^2(\Omega)^d$ and $H_0^1(\Omega)^d$, respectively.

Moreover, let $I \coloneqq (0,T)$ with $0 < T < \infty$ be a bounded time interval. For any Banach space X and $1 \leq p \leq \infty$, $L^p(I,X)$ denotes the space of L^p-integrable functions f from I into X. This is a Banach space with the norm

$$\begin{aligned} \|f\|_{L^p(I,X)} &\coloneqq \left(\int_I \|f(t)\|_X^p \, \mathrm{d}t \right)^{\frac{1}{p}}, \quad 1 \leq p < \infty, \\ \|f\|_{L^\infty(I,X)} &\coloneqq \operatorname*{ess\,sup}_{t \in I} \|f(t)\|_X . \end{aligned}$$

For a detailed derivation of these spaces by means of the Bochner integral, we refer to Dautray and Lions [35] and Wloka [108].

The space $C^k(\bar{I},X)$, $k \in \mathbb{N}$, consists of functions from $\bar{I}$ into X that are k times continuously differentiable on I and whose derivatives $\partial_t^j f(t)$ of order $0 \leq j \leq k$ possess continuous extensions onto $\bar{I}$. This is a Banach space with the norm

$$\|f\|_{C^k(\bar{I},X)} \coloneqq \max_{0 \leq j \leq k} \sup_{t \in \bar{I}} \left\| \partial_t^j f(t) \right\|_X .$$

We set $C(\bar{I},X) \coloneqq C^0(\bar{I},X)$.

2.2 The incompressible Navier-Stokes equations

This section is devoted to the incompressible Navier-Stokes equations and their variational formulation.

The incompressible Navier-Stokes equations describe the motion of incompressible fluids and are based on the physical conservation laws of mass and momentum. A detailed derivation can be found in the standard literature, see, for instance, Chorin and Marsden [31] or Feistauer [51].

The conservation of mass is equivalent to the so-called *continuity equation*

$$\partial_t \rho + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{2.2}$$

where ρ describes the density and $\boldsymbol{v}$ the velocity of the fluid. The balance of momentum leads to the *momentum equation*

$$\rho\partial_t\boldsymbol{v} + \rho(\boldsymbol{v}\cdot\nabla)\boldsymbol{v} - \nabla\cdot\boldsymbol{\sigma}(\boldsymbol{v},p) = \rho\boldsymbol{f} \tag{2.3}$$

with the stress tensor $\boldsymbol{\sigma}(\boldsymbol{v},p)$ and external forces $\boldsymbol{f}$. In the sequel, we will only consider Newtonian fluids. For such fluids, the stress tensor $\boldsymbol{\sigma}(\boldsymbol{v},p)$ is given by

$$\boldsymbol{\sigma}(\boldsymbol{v},p) = (-p + \lambda\nabla\cdot\boldsymbol{v})\mathbf{I} + \mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T),$$

where p denotes the pressure and λ and μ are constants.

Incompressible fluids are characterized by the fact that the volume of subregions occupied by the fluid does not change in time. Using Reynolds' transport theorem, this leads to the incompressibility condition

$$\nabla\cdot\boldsymbol{v} = 0. \tag{2.4}$$

Inserting this into the momentum equation (2.3) yields

$$\rho\partial_t\boldsymbol{v} - \mu\Delta\boldsymbol{v} + \rho(\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p = \rho\boldsymbol{f}. \tag{2.5}$$

Inserting the incompressibility condition (2.4) into the continuity equation (2.2), we obtain

$$\partial_t\rho + \boldsymbol{v}\cdot\nabla\rho = 0. \tag{2.6}$$

If the fluid under consideration is *homogeneous*, i. e., the density ρ is constant in space, then the density is also constant in time due to (2.6). In this case, (2.5) can be simplified to

$$\partial_t\boldsymbol{v} - \nu\Delta\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \rho_0^{-1}\nabla p = \boldsymbol{f},$$

where $\nu \coloneqq \rho_0^{-1}\mu$ denotes the kinematic viscosity and $\rho \equiv \rho_0$ denotes the constant density. The corresponding dimensionless form reads

$$\partial_t\boldsymbol{v} - \frac{1}{\mathrm{Re}}\Delta\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p = \boldsymbol{f}. \tag{2.7}$$

Here, the so-called *Reynolds number* Re is given by

$$\mathrm{Re} = \frac{LU}{\nu} = \frac{LU\rho_0}{\mu}$$

with a characteristic length L and a characteristic velocity U.

In the remaining part of this section, we present the variational formulation of the incompressible Navier-Stokes equations (2.7) involving the pressure. This will be the basis for the discretization presented in Chapter 3.

Let us assume that $\boldsymbol{v} \in C^1(I, C(\Omega)) \cap C(I, C^2(\Omega)) \cap C(\bar{I}, C(\overline{\Omega}))$ and $p \in C(I, C^1(\Omega))$ are classical solutions of the incompressible Navier-Stokes equations (2.7). For any $\boldsymbol{\psi} \in \mathcal{V}$ we then have after integration by parts ($\nu = \mathrm{Re}^{-1}$)

$$(\partial_t\boldsymbol{v}, \boldsymbol{\psi}) + \nu(\nabla\boldsymbol{v}, \nabla\boldsymbol{\psi}) + ((\boldsymbol{v}\cdot\nabla)\boldsymbol{v}, \boldsymbol{\psi}) = (\boldsymbol{f}, \boldsymbol{\psi}). \tag{2.8}$$

Due to the Hölder inequality and the continuous embedding $H_0^1(\Omega) \hookrightarrow L^4(\Omega)$ for $\Omega \subseteq \mathbb{R}^d$, $d \in \{ 2, 3 \}$,

$$((\boldsymbol{v_1} \cdot \nabla)\boldsymbol{v_2}, \boldsymbol{v_3})$$

is a continuous trilinear form on V and hence the identity (2.8) is true also for all $\boldsymbol{\psi} \in V$ by continuity. Clearly there holds

$$(\partial_t \boldsymbol{v}, \boldsymbol{\psi}) = \frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{v}, \boldsymbol{\psi}).$$

Obviously, a classical solution $\boldsymbol{v}$ also fulfills $\boldsymbol{v} \in L^2(I, V)$. This gives rise to the following weak formulation:

Problem 2.1. *For $\boldsymbol{f} \in L^2(I, V^*)$ and $\boldsymbol{v^0} \in H$ find $\boldsymbol{v} \in L^2(I, V)$ with*

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{v}, \boldsymbol{\psi}) + \nu(\nabla \boldsymbol{v}, \nabla \boldsymbol{\psi}) + ((\boldsymbol{v} \cdot \nabla)\boldsymbol{v}, \boldsymbol{\psi}) = \langle \boldsymbol{f}, \boldsymbol{\psi} \rangle \quad \forall \boldsymbol{\psi} \in V, \tag{2.9a}$$

$$\boldsymbol{v}(0) = \boldsymbol{v^0}. \tag{2.9b}$$

Before introducing the pressure, let us first state the following fundamental lemma:

Lemma 2.1 ("Inf-Sup" Condition). *For each $\boldsymbol{f} \in H^{-1}(\Omega)^d$ with $\langle \boldsymbol{f}, \boldsymbol{\psi} \rangle = 0 \ \forall \boldsymbol{\varphi} \in V$ there is a unique $p \in L^2(\Omega)/\mathbb{R}$ such that*

$$\boldsymbol{f} = \nabla p, \quad \textit{i. e.,} \quad \langle \boldsymbol{f}, \boldsymbol{\psi} \rangle = \langle \nabla p, \boldsymbol{\psi} \rangle = -(p, \nabla \cdot \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_0^1(\Omega)^d.$$

Furthermore, the "inf-sup" stability condition holds:

$$\|p\|_{L^2(\Omega)/\mathbb{R}} \leq C \, \|\nabla p\|_{H^{-1}(\Omega)} = C \sup_{\boldsymbol{\psi} \in H_0^1(\Omega)^d} \frac{\left|(p, \nabla \cdot \boldsymbol{\psi})\right|}{\|\nabla \boldsymbol{\psi}\|}.$$

Proof. Proofs of this fundamental property can be found, for instance, in Girault and Raviart [58] or Temam [99]. □

By integrating (2.9a) with respect to t and applying Lemma 2.1, we obtain the pressure

$$p := \partial_t P$$

as the distributional derivative of a function $P \in C(\bar{I}, L^2(\Omega)/\mathbb{R})$ (see Temam [99] for more details on the introduction of the pressure). Together with $\boldsymbol{v} \in L^2(I, V)$ there holds

$$\partial_t \boldsymbol{v} - \nu \Delta \boldsymbol{v} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + \nabla p = \boldsymbol{f}$$

in the distribution sense on $I \times \Omega$.

Under additional requirements on the data one can show that the weak solution of Problem 2.1 possesses further regularity and that the pressure can be viewed as an almost everywhere defined function rather than only as a distribution. The results differ between the two-dimensional and the three-dimensional case and are collected in the following proposition:

Proposition 2.2. *Let Ω be a bounded domain in $\mathbb{R}^d$ with boundary of class C^2 and $\boldsymbol{v}^{\mathbf{0}} \in V$.*

(i) $d = 2$: For $\boldsymbol{f} \in L^2(I, H)$ there exists a unique solution to Problem 2.1 which satisfies

$$\boldsymbol{v} \in L^2(I, H^2(\Omega)^2 \cap H_0^1(\Omega)^2) \cap L^\infty(I, V) \quad \text{and} \quad \partial_t \boldsymbol{v} \in L^2(I, H).$$

For the corresponding pressure we obtain $p \in L^2(I, H^1(\Omega)^2)$.

(ii) $d = 3$: For $\boldsymbol{f} \in L^\infty(I, H)$ there exists $T^* \leq T$ such that Problem 2.1 possesses a unique solution $\boldsymbol{v}$ on $I^* \coloneqq (0, T^*)$. Moreover,

$$\boldsymbol{v} \in L^2(I^*, H^2(\Omega)^3 \cap H_0^1(\Omega)^3) \cap L^\infty(I^*, V) \quad \text{and} \quad \partial_t \boldsymbol{v} \in L^2(I^*, H).$$

For the corresponding pressure we obtain $p \in L^2(I^*, H^1(\Omega)^3)$.

Proof. We refer to Cattabriga [29] or Temam [99] for a proof of this regularity and uniqueness result. □

Remark 2.1. The prerequisites concerning the regularity of the boundary $\partial\Omega$ can be weakened. The statement of Proposition 2.2 remains true if $\Omega \subseteq \mathbb{R}^d$ is a polygonally bounded convex domain or its boundary is of class $C^{1,1}$, see Heywood and Rannacher [65] as well as Kellogg and Osborn [76] for the case $d = 2$ and Dauge [34] for the case $d = 3$.

This allows us to set up a variational formulation incorporating the pressure: Under the assumptions of Proposition 2.2 (or Remark 2.1) the solution $\boldsymbol{u} \coloneqq (\boldsymbol{v}, p)^T \in X$ of Problem 2.1 satisfies

$$\int_I \{(\partial_t \boldsymbol{v}, \boldsymbol{\psi}) + \nu(\nabla \boldsymbol{v}, \nabla \boldsymbol{\psi}) + ((\boldsymbol{v} \cdot \nabla)\boldsymbol{v}, \boldsymbol{\psi}) - (p, \nabla \cdot \boldsymbol{\psi}) + (\nabla \cdot \boldsymbol{v}, \chi)\} \, \mathrm{d}t + (\boldsymbol{v}(0) - \boldsymbol{v}^{\mathbf{0}}, \boldsymbol{\psi}(0)) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t \quad \forall \boldsymbol{\varphi} \coloneqq (\boldsymbol{\psi}, \chi)^T \in X, \quad (2.10)$$

where

$$X \coloneqq \left\{ \boldsymbol{u} = (\boldsymbol{v}, p)^T \;\middle|\; \boldsymbol{v} \in L^2(I, H_0^1(\Omega)^d),\ \partial_t \boldsymbol{v} \in L^2(I, L^2(\Omega)^d),\ p \in L^2(I, L^2(\Omega)/\mathbb{R}) \right\}$$

with I replaced by I^* for $d = 3$. For simplicity, we will assume $I^* = I$ in what follows.

Remark 2.2. It is well known that the space X is continuously embedded in $C(\bar{I}, L^2(\Omega)^d) \times L^2(I, L^2(\Omega)/\mathbb{R})$, see, for instance, Dautray and Lions [35]. Hence, the expression $\boldsymbol{v}(0)$ makes sense for functions $\boldsymbol{u} = (\boldsymbol{v}, p)^T \in X$.

Remark 2.3. In applications, we will sometimes be confronted with configurations in which Dirichlet boundary conditions for the velocity are not prescribed on the whole boundary. Instead, there will be some part Γ_{out} of the boundary representing an outlet. Then, we apply natural boundary conditions on Γ_{out}:

$$\nu \partial_{\boldsymbol{n}} \boldsymbol{v} - p\boldsymbol{n} = \mathbf{0}.$$

This type of boundary condition implicitly normalizes the pressure such that it is already uniquely determined without the usual mean value constraint. Hence, the spaces in which the solutions are sought have to be modified to

$$\boldsymbol{v} \in \left\{ \boldsymbol{v} \in H^1(\Omega)^d \;\middle|\; \boldsymbol{v}\big|_{\partial\Omega \setminus \Gamma_{\text{out}}} = \boldsymbol{0} \right\} \quad \text{and} \quad p \in L^2(\Omega).$$

However, the description of the discretization in the next chapter will be based on the case where Dirichlet boundary conditions for $\boldsymbol{v}$ are posed on the whole boundary $\partial\Omega$.

For more information on this free outflow boundary condition as well as results concerning existence and uniqueness of solutions, we refer to Heywood, Rannacher, and Turek [67].

To shorten the notation, we will frequently use the semi-linear form defined by

$$a(\boldsymbol{u})(\boldsymbol{\varphi}) \coloneqq \int_I \bar{a}(\boldsymbol{u}(t))(\boldsymbol{\varphi}(t)) \, \mathrm{d}t$$

with

$$\bar{a}(\boldsymbol{u})(\boldsymbol{\varphi}) \coloneqq \nu(\nabla \boldsymbol{v}, \nabla \boldsymbol{\psi}) + ((\boldsymbol{v} \cdot \nabla)\boldsymbol{v}, \boldsymbol{\psi}) - (p, \nabla \cdot \boldsymbol{\psi}) + (\nabla \cdot \boldsymbol{v}, \chi).$$

3 Space-Time Finite Element Discretization

In this chapter, we describe the discretization of the weak formulation of the incompressible Navier-Stokes equations (2.10) involving the pressure. The discretization in space as well as in time will be done by means of Galerkin finite element methods.

In Section 3.1, we present the semi-discretization in time by *continuous Galerkin* (cG) and *discontinuous Galerkin* (dG) methods. Section 3.2 then deals with the discretization in space of the arising semi-discrete problems. This is done by continuous Galerkin finite element methods. For technical reasons, we use piecewise polynomial functions of the same degree for the velocity and the pressure component. Hence, the Babuška-Brezzi stability condition is not fulfilled. Therefore, we have to apply stabilization techniques. This is done by the so-called *local projection stabilization* (LPS) and is described in Section 3.3 in more detail. In Section 3.4, we give the precise time-stepping formulation of the fully discrete problems for the cG(s)dG(0), cG(s)dG(1), and cG(s)cG(1) discretization. Section 3.5 finally deals with implementational aspects of the discretization such as the computation on dynamic meshes, i. e., spatial meshes that change in time.

3.1 Discretization in time

This section is devoted to the semi-discretization in time of the incompressible Navier-Stokes equations (2.10) by means of Galerkin finite element methods. A more detailed introduction and motivation of these concepts in the context of ordinary differential equations or parabolic partial differential equations is presented in the textbooks of Eriksson, Estep, Hansbo, and Johnson [42] or Thomée [100].

The first type of discretization which we call *discontinuous Galerkin method of degree* r or dG(r) method uses discontinuous trial and test functions for the velocity and the pressure which are piecewise polynomials of degree r; see Section 3.1.1. The second method uses continuous trial functions (piecewise polynomials of degree r) for the velocity and discontinuous trial functions for the pressure as well as discontinuous test functions for the velocity and the pressure (piecewise polynomials of degree $r-1$). We call this method *continuous Galerkin method of degree* r or simply cG(r) method; see Section 3.1.2.

To introduce these semi-discretizations in time, we partition the time interval $\bar{I} = [0, T]$ into

$$\bar{I} = \{\, 0 \,\} \cup I_1 \cup \cdots \cup I_m \cup \cdots \cup I_M$$

with subintervals $I_m := (t_{m-1}, t_m]$ of length $k_m := t_m - t_{m-1}$ using time points

$$0 = t_0 < t_1 < \cdots < t_m < \cdots < t_M = T.$$

The discretization parameter k is given as a piecewise constant function by setting $k\big|_{I_m} := k_m$ for $m = 1, \ldots, M$.

Using the subintervals I_m, let us define the following semi-discrete spaces X_k^r and $\widetilde{X}_k^r$ for $r \in \mathbb{N}$:

$$\begin{aligned}
X_k^r := \Big\{ \boldsymbol{u_k} = (\boldsymbol{v_k}, p_k)^T \in C(\bar{I}, L^2(\Omega)^d) \times L^2(I, L^2(\Omega)/\mathbb{R}) \;\Big|\; \boldsymbol{v_k}\big|_{I_m} \in \mathcal{P}_r(I_m, H_0^1(\Omega)^d), \\
p_k\big|_{I_m} \in \mathcal{P}_{r-1}(I_m, L^2(\Omega)/\mathbb{R}),\ m = 1, \ldots, M \Big\}, \\
\widetilde{X}_k^r := \Big\{ \boldsymbol{u_k} = (\boldsymbol{v_k}, p_k)^T \;\Big|\; \boldsymbol{v_k}(0) \in L^2(\Omega)^d,\ \boldsymbol{v_k}\big|_{I_m} \in \mathcal{P}_r(I_m, H_0^1(\Omega)^d), \\
p_k\big|_{I_m} \in \mathcal{P}_r(I_m, L^2(\Omega)/\mathbb{R}),\ m = 1, \ldots, M \Big\} \subseteq L^2(I, H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R}),
\end{aligned}$$

where $\mathcal{P}_r(I_m, Y)$ denotes the space of polynomials up to degree r on I_m with values in Y. The space X_k^r is used as trial space in the continuous Galerkin method whereas the space $\widetilde{X}_k^r$ is used as trial and test space in the discontinuous Galerkin method and as test space in the continuous Galerkin method.

Remark 3.1. The spaces for the semi-discrete pressure p_k are chosen to allow discontinuous functions in both discontinuous and continuous Galerkin methods due to the missing initial conditions for p.

To account for the possible discontinuity of a function u_k at time points t_m, we introduce the notation

$$u_{k,m}^+ := \lim_{\varepsilon \downarrow 0} u_k(t_m + \varepsilon), \quad u_{k,m}^- := \lim_{\varepsilon \downarrow 0} u_k(t_m - \varepsilon), \quad [u_k]_m := u_{k,m}^+ - u_{k,m}^-.$$

Thus, $u_{k,m}^+$ is the limit "from above" while $u_{k,m}^-$ denotes the limit "from below". $[u_k]_m$ then is the "jump" of $u_k(t)$ at $t = t_m$, see Figure 3.1.

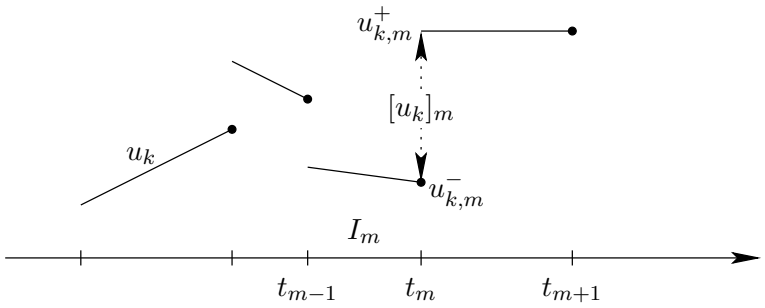


Figure 3.1. Notation of discontinuous functions u_k in the case $r = 1$

3.1.1 Discontinuous Galerkin methods

The dG(r) semi-discretization of the incompressible Navier-Stokes equations (2.10) reads: Find $\boldsymbol{u_k} = (\boldsymbol{v_k}, p_k)^T \in \widetilde{X}_k^r$ such that

$$\sum_{m=1}^{M} \int_{I_m} (\partial_t \boldsymbol{v_k}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_k})(\boldsymbol{\varphi}) + \sum_{m=0}^{M-1} ([\boldsymbol{v_k}]_m, \boldsymbol{\psi}_m^+) + (\boldsymbol{v}_{\boldsymbol{k},0}^-, \boldsymbol{\psi}_0^-) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_k^r. \quad (3.1)$$

Remark 3.2. Many authors use the formulation

$$\sum_{m=1}^{M} \int_{I_m} (\partial_t \boldsymbol{v_k}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_k})(\boldsymbol{\varphi}) + \sum_{m=2}^{M} ([\boldsymbol{v_k}]_{m-1}, \boldsymbol{\psi}_{m-1}^+) + (\boldsymbol{v}_{\boldsymbol{k},0}^+, \boldsymbol{\psi}_0^+) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^+) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_k^r \quad (3.2)$$

and eliminate $\boldsymbol{v_k}(0)$ in the definition of $\widetilde{X}_k^r$. The equivalence of (3.1) and (3.2) can be seen as follows:

By subtracting equation (3.2) from (3.1), we obtain

$$(\boldsymbol{v}_{\boldsymbol{k},0}^-, \boldsymbol{\psi}_0^- - \boldsymbol{\psi}_0^+) = (\boldsymbol{v^0}, \boldsymbol{\psi}_0^- - \boldsymbol{\psi}_0^+). \quad (3.3)$$

If this equation is valid, the equivalence of both formulations is shown. But this is either directly fulfilled due to formulation (3.1) since the terms containing $\boldsymbol{\psi}_0^-$ can be separated from the remainder and hence

$$(\boldsymbol{v}_{\boldsymbol{k},0}^-, \boldsymbol{\psi}) = (\boldsymbol{v^0}, \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in L^2(\Omega)^d,$$

or by simply setting the undefined values $\boldsymbol{v}_{\boldsymbol{k},0}^-$ in the case of formulation (3.2).

However, we prefer formulation (3.1) especially for implementational reasons. This is because the same data structures can then be used as for the continuous Galerkin methods introduced in the next section.

3.1.2 Continuous Galerkin methods

With the semi-discrete spaces defined at the beginning of this section, the cG(r) formulation can directly be stated as follows: Find $\boldsymbol{u_k} = (\boldsymbol{v_k}, p_k)^T \in X_k^r$ such that

$$\int_I (\partial_t \boldsymbol{v_k}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_k})(\boldsymbol{\varphi}) + (\boldsymbol{v_k}(0), \boldsymbol{\psi}_0^-) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_k^{r-1}. \quad (3.4)$$

Remark 3.3. In the formulation (3.4), the polynomial degree of the test functions for the velocity is reduced by one compared to the corresponding polynomial degree for the trial functions. This is done to obtain the well-posedness of the problem which requires the compensation of the additional degrees of freedom in $\widetilde{X}_k^r$ for the velocities due to their possible discontinuity.

3.2 Discretization in space

So far, we have only considered semi-discretization in time. Hence, the spaces X_k^r and $\widetilde{X}_k^r$ still contain the continuous spatial spaces $H_0^1(\Omega)^d$ and $L^2(\Omega)/\mathbb{R}$. In this section, we present the discretization in space of the semi-discrete problems obtained in the previous section. This is again done by means of continuous Galerkin finite element methods. To this end, we introduce finite dimensional subspaces $V_h^s \subseteq H^1(\Omega)$ of piecewise polynomial functions up to order s.

3.2.1 Finite element spaces

The definition of finite element spaces is closely linked to the decomposition of the computational domain $\Omega \subseteq \mathbb{R}^d$, $d \in \{ 2, 3 \}$. For sake of simplicity, let us for the moment assume the boundary $\partial\Omega$ to be polygonal. The case of non-polygonally bounded domains is discussed later on.

The computational domain Ω is partitioned into open cells K which are depending on the spatial dimension d either quadrilaterals ($d = 2$) or hexahedrons ($d = 3$). All cells together form the mesh $\mathcal{T}_h = \{ K \}$ of the domain, where the parameter h is given as a cell-wise constant function $h\big|_K = h_K \coloneqq \operatorname{diam}(K)$ with the diameter h_K of a cell. The symbol h also denotes the maximum cell diameter, that is

$$h \coloneqq \max_{K \in \mathcal{T}_h} h_K.$$

Following standard literature like Braess [23], Brenner and Scott [24], or Ciarlet [32], we formulate the following definition:

Definition 3.1 (Regularity). *A mesh $\mathcal{T}_h = \{ K \}$ is called* regular *if the following conditions are fulfilled:*

i) $\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} \overline{K}$.

ii) $K_1 \cap K_2 = \emptyset$ *for all cells* $K_1, K_2 \in \mathcal{T}_h$ *with* $K_1 \neq K_2$.

iii) Any face of any cell $K_1 \in \mathcal{T}_h$ is either a subset of the boundary $\partial\Omega$ or a face of another cell $K_2 \in \mathcal{T}_h$.

The last condition is weakened for two reasons: In order to facilitate adaptive mesh refinement and to avoid connecting elements, we introduce so-called *hanging nodes*. Cells are allowed to have nodes which lie on midpoints of faces or edges of neighboring cells. At the most one hanging node is allowed on each face or edge. Also note that there are two types of hanging nodes in three dimensions (on faces or edges), while in two dimensions there is only one type of hanging nodes (only on edges, see Figure 3.2).

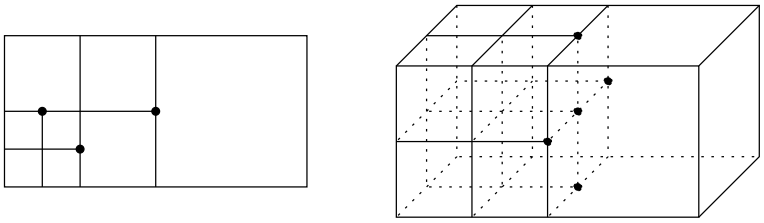


Figure 3.2. Two- and three-dimensional meshes with hanging nodes.

The second reason concerns the case of a non-polygonal boundary $\partial\Omega$. In such a case, we alleviate the third condition of a face being a subset of the boundary to only requiring the vertices of such a face (and possibly some inner points) to be located on the boundary. This is discussed in more detail later.

In addition, we will require that the mesh is organized in a patch-wise manner. That is, $\mathcal{T}_h$ is obtained by uniform refinement of a coarser mesh $\mathcal{T}_{2h}$, such that we can always combine four ($d = 2$) or eight ($d = 3$) adjacent cells of $\mathcal{T}_h$ to obtain one cell of $\mathcal{T}_{2h}$. Such macro-cells are called *patches* (see Figure 3.3). This construction will be of essential importance in the context of a posteriori error estimation, see Chapter 4.

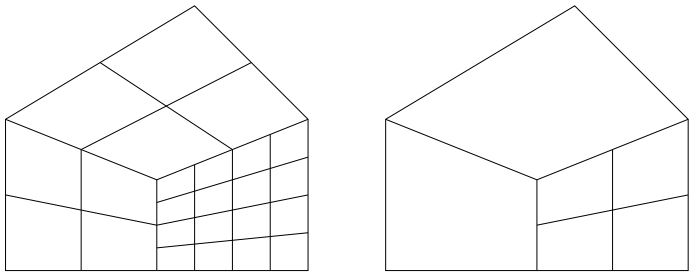


Figure 3.3. Two-dimensional mesh $\mathcal{T}_h$ (with hanging nodes) organized in a patch-wise manner with corresponding coarser mesh $\mathcal{T}_{2h}$

Following Brenner and Scott [24], Ciarlet [32], or Johnson [74], we define continuous H^1-conforming finite element spaces V_h^s by

$$V_h^s \coloneqq \left\{ u_h \in C(\overline{\Omega}) \;\middle|\; u_h\big|_K \in \mathcal{Q}_s(K)\ \forall K \in \mathcal{T}_h \right\} \subseteq H^1(\Omega).$$

Here, $\mathcal{Q}_s(K)$ denotes the space of polynomial-like functions on $K \in \mathcal{T}_h$. To give a more precise definition of $\mathcal{Q}_s(K)$, we introduce the space $\hat{\mathcal{Q}}_s(\hat{K})$ of tensor product polynomials up to degree s on the reference cell $\hat{K} = (0,1)^d$ given as

$$\hat{\mathcal{Q}}_s(\hat{K}) \coloneqq \operatorname{span}\left\{ \prod_{i=1}^{d} \hat{x}_i^{\alpha_i} \;\middle|\; \alpha_i \in \{\, 0, 1, \ldots, s \,\} \right\}.$$

The lowest order case $s = 1$ results in the space of bi- ($d = 2$) or tri-linear ($d = 3$) functions

$$\begin{aligned} \hat{\mathcal{Q}}_1(\hat{K}) &= \operatorname{span}\{\, 1, \hat{x}_1, \hat{x}_2, \hat{x}_1\hat{x}_2 \,\} && (d = 2) \text{ or} \\ \hat{\mathcal{Q}}_1(\hat{K}) &= \operatorname{span}\{\, 1, \hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_1\hat{x}_2, \hat{x}_1\hat{x}_3, \hat{x}_2\hat{x}_3, \hat{x}_1\hat{x}_2\hat{x}_3 \,\} && (d = 3). \end{aligned}$$

Then, the space $\mathcal{Q}_s(K)$ is obtained using transformations $\boldsymbol{T_K} : \hat{K} \to K$ (see Figure 3.4) by

$$\mathcal{Q}_s(K) \coloneqq \left\{ u_h : K \to \mathbb{R} \;\middle|\; u_h \circ \boldsymbol{T_K} \in \hat{\mathcal{Q}}_s(\hat{K}) \right\}.$$

If the transformation $\boldsymbol{T_K}$ itself is an element of $\hat{\mathcal{Q}}_s(\hat{K})^d$, the resulting finite element space is called *isoparametric*.

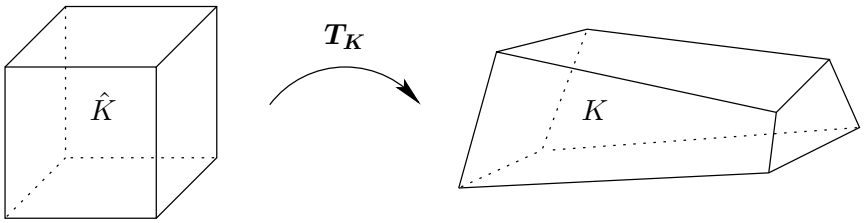
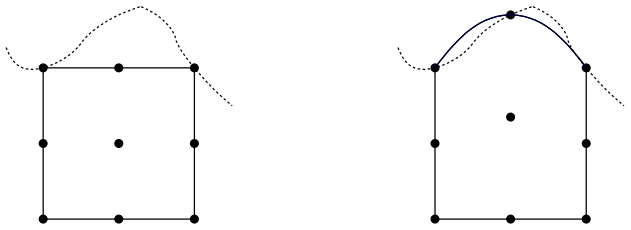


Figure 3.4. Transformation $\boldsymbol{T_K}$ from the reference cell $\hat{K}$ to a computational cell K

Let us now consider the case of domains with non-polygonal boundary. Regarding higher order elements ($s > 1$), we have degrees of freedom associated to points on edges or faces of elements. In the context of curved boundaries, there are two possibilities to distribute these degrees of freedom (see Figure 3.5 for the case $s = 2$). Using only bi- or trilinear transformations (which means only the vertices of the "physical" cell are located on the boundary, see Figure 3.5(a)), may lead to the wrong approximation order and reduced accuracy. In contrast, using isoparametric finite element spaces allows us to choose the transformation in such a way that also the degrees of freedom located on edges or faces are located on the real boundary. We also adjust the position of the inner degrees of freedom (see Figure 3.5(b)). In this thesis, we always use isoparametric finite element spaces.

The definition of the finite element spaces in the case of hanging nodes needs some remarks. To enforce global continuity and hence global conformity, the degrees of freedom located on the interface between cells of different refinement levels have to fulfill additional constraints, roughly spoken they are determined by interpolation of neighboring degrees of freedom.



(a) Bilinear transformation (b) Isoparametric transformation

Figure 3.5. Two possible distributions of the degrees of freedom in connection with curved boundaries for $\mathcal{Q}_2(K)$ elements in two spatial dimensions

Hence, hanging nodes actually do not carry any degrees of freedom. For details on this concept, we refer to Carey and Oden [27].

For ensuring approximation properties of the finite element spaces, additional conditions on the geometry of the cells are required. We state two classical assumptions in this context, namely the so-called *uniformity* and the weaker *quasi-uniformity*, see, for example, Braess [23]:

Definition 3.2 (Quasi-Uniformity). *A family of meshes* $\{ \mathcal{T}_h \mid h \downarrow 0 \}$ *is called* quasi-uniform *if there is a constant* κ *such that the following two conditions are fulfilled:*

i) For each transformation $\boldsymbol{T_K} \colon \hat{K} \to K$ *it holds*

$$\frac{\sup \left\{ \left\| \nabla \boldsymbol{T_K}(\hat{\boldsymbol{x}}) \boldsymbol{x} \right\| \;\middle|\; \hat{\boldsymbol{x}} \in \hat{K},\ \|\boldsymbol{x}\| = 1 \right\}}{\inf \left\{ \left\| \nabla \boldsymbol{T_K}(\hat{\boldsymbol{x}}) \boldsymbol{x} \right\| \;\middle|\; \hat{\boldsymbol{x}} \in \hat{K},\ \|\boldsymbol{x}\| = 1 \right\}} \leq \kappa \quad \forall K \in \bigcup_h \mathcal{T}_h.$$

ii) With the diameter ρ_K *of the biggest ball inscribed into the cell* K *there holds*

$$\frac{h_K}{\rho_K} \leq \kappa \quad \forall K \in \bigcup_h \mathcal{T}_h.$$

Definition 3.3 (Uniformity). *A quasi-uniform family of meshes* $\{ \mathcal{T}_h \mid h \downarrow 0 \}$ *is called* uniform *if there is a constant* κ *such that*

$$\frac{h}{\rho_K} \leq \kappa \quad \forall K \in \bigcup_h \mathcal{T}_h.$$

By using the lemmas of Lax-Milgram and Céa (see, for example, Alt [2] or Braess [23]), it is possible to estimate the approximation error of finite elements by interpolation errors. For continuous functions $u \in C(\overline{\Omega})$ the point-wise interpolation operator $I_h \colon C(\overline{\Omega}) \to V_h^s$ is well-defined. It satisfies the approximation properties stated in the following lemma:

Lemma 3.1. *Let $\mathcal{T}_h$ be a quasi-uniform family of triangulations of the domain Ω and V_h^s be the spaces of isoparametric finite elements of order s. Then, there exists a constant C, only depending on Ω and s, such that for $u \in H^m(\Omega)$ with $2 \leq m \leq s+1$ and $0 \leq j \leq m$ there holds*

$$|u - I_h u|_{H^j(\Omega)} \leq C h^{m-j} \|u\|_{H^m(\Omega)} .$$

Proof. The proof of this classical result can be found, for instance, in Braess [23]. □

Clément [33] introduced a generalized interpolation operator $\tilde{I}_h \colon H^1(\Omega) \to V_h^1$ which is also applicable for functions $u \in H^1(\Omega)$ which are not continuous. Its approximation properties are summarized in the following lemma:

Lemma 3.2. *Let $\mathcal{T}_h$ be a quasi-uniform family of triangulations of the domain Ω and V_h^1 be the space of isoparametric finite elements of order 1. Then, there exists a constant C, only depending on Ω, such that for $u \in H^1(\Omega)$ and $0 \leq j \leq 1$ there holds*

$$\left| u - \tilde{I}_h u \right|_{H^j(\Omega)} \leq C h^{1-j} \|u\|_{H^1(\Omega)} .$$

Proof. For a proof of this lemma, we refer to the literature cited above. □

On uniform meshes, we are able to estimate stronger norms of finite element functions in terms of weaker norms. Such estimates are called *inverse estimates*:

Lemma 3.3 (Inverse Estimate). *Let $\mathcal{T}_h$ be a uniform family of triangulations of the domain Ω and V_h^s be the space of isoparametric finite elements of order s. Then, there exists a constant C such that for any $u_h \in V_h^s \cap H^m(\Omega)$ and $0 \leq i \leq j \leq m$ there holds*

$$|u_h|_{H^j(\Omega)} \leq C h^{i-j} |u_h|_{H^i(\Omega)} .$$

The constant C only depends on Ω, s, i, and j.

Proof. The proof can be found in standard finite element textbooks like Braess [23] or Ciarlet [32]. □

3.2.2 Discretization on dynamic meshes

This section is devoted to the formulation of the fully discrete problems arising by spatial discretization of the semi-discrete problems presented in Section 3.1. For efficiency reasons, it is desirable to allow the spatial meshes to change in time. To this end, we introduce the concept of dynamic meshes. This is done in the same way as in Schmich and Vexler [96] where these concepts are successfully used in the numerical simulation of nonlinear parabolic problems.

For implementational simplicity, we allow the spatial meshes to change in time whereas the time steps k_m are kept constant in space. Thus, let us associate with each time point t_m a spatial mesh $\mathcal{T}_h^m$ and corresponding conforming finite element spaces $V_h^{s_{\boldsymbol{v}},m}$ and $V_h^{s_p,m}$

which will be used as trial and test spaces on the time interval I_m. This allows us to define the following fully discrete space-time finite element space

$$\widetilde{X}_{kh}^{r,s} := \Big\{ \boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \;\Big|\; \boldsymbol{v_{kh}}(0) \in (H_h^0)^d,\ \boldsymbol{v_{kh}}\Big|_{I_m} \in \mathcal{P}_r(I_m, (H_h^m)^d),$$
$$p_{kh}\Big|_{I_m} \in \mathcal{P}_r(I_m, L_h^m),\ m = 1, \ldots, M \Big\} \subseteq L^2(I, H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R}),$$

where

$$H_h^m := V_h^{s_{\boldsymbol{v}},m} \cap H_0^1(\Omega) \quad \text{and} \quad L_h^m := V_h^{s_p,m} \cap L^2(\Omega)/\mathbb{R}.$$

Because of the conformity of H_h^m and L_h^m, we have $\widetilde{X}_{kh}^{r,s} \subseteq \widetilde{X}_k^r$.

Hence, the cG(s)dG(r) formulation of problem (2.10) is easily obtained from the dG(r) semi-discretization in time by simply adding the additional index h to the variables and by replacing the semi-discrete space $\widetilde{X}_k^r$ by $\widetilde{X}_{kh}^{r,s}$. Thus, the cG(s)dG(r) formulation of problem (2.10) reads: Find $\boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in \widetilde{X}_{kh}^{r,s}$ such that

$$\sum_{m=1}^{M} \int_{I_m} (\partial_t \boldsymbol{v_{kh}}, \boldsymbol{\psi})\, \mathrm{d}t + a(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + \sum_{m=0}^{M-1} ([\boldsymbol{v_{kh}}]_m, \boldsymbol{\psi}_m^+) + (\boldsymbol{v}_{\boldsymbol{kh},0}^-, \boldsymbol{\psi}_0^-)$$
$$= \int_I (\boldsymbol{f}, \boldsymbol{\psi})\, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_{kh}^{r,s}. \quad (3.5)$$

Remark 3.4. The notation cG(s)dG(r), representing a space-time finite element discretization with continuous piecewise polynomials of degree s in space and discontinuous piecewise polynomials of degree r in time, is taken from Eriksson, Estep, Hansbo, and Johnson [42]. In a similar way, we use the notation cG(s)cG(r) for a space-time finite element discretization which uses continuous piecewise polynomials of degree s in space and continuous piecewise polynomials of degree r in time for the velocity and discontinuous piecewise polynomials of degree $r-1$ in time for the pressure.

When setting up the corresponding fully discrete formulation for the cG(r) semi-discretization, we have to ensure the global continuity of the trial functions for the velocity. In the sequel, we describe an approach already presented in Schmich and Vexler [96] which is similar to the one of Becker [8]. We also refer to Bangerth [5] for an application of this approach to the wave equation.

Let $\{ \tau_0, \ldots, \tau_r \}$ be a basis of $\mathcal{P}_r(I_m, \mathbb{R})$ with

$$\tau_0(t_{m-1}) = 1, \quad \tau_0(t_m) = 0, \quad \tau_i(t_{m-1}) = 0,\ i = 1, \ldots, r.$$

By means of this basis, we define the spaces $X_{kh}^{r,s,m} \subseteq \mathcal{P}_r(I_m, H_0^1(\Omega)^d)$ by

$$X_{kh}^{r,s,m} := \operatorname{span} \Big\{ \tau_i \boldsymbol{\psi_i} \;\Big|\; \boldsymbol{\psi_0} \in (H_h^{m-1})^d,\ \boldsymbol{\psi_i} \in (H_h^m)^d,\ i = 1, \ldots, r \Big\}.$$

Using these spaces, we give the definition of the fully discrete space $X_{kh}^{r,s}$ used as trial space in the cG(s)cG(r) formulation of problem (2.10) by

$$X_{kh}^{r,s} := \Big\{ \boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in C(\bar{I}, L^2(\Omega)^d) \times L^2(I, L^2(\Omega)/\mathbb{R}) \;\Big|\; \boldsymbol{v_{kh}}\Big|_{I_m} \in X_{kh}^{r,s,m}, \\ p_{kh}\Big|_{I_m} \in \mathcal{P}_{r-1}(I_m, L_h^m),\ m = 1, \ldots, M \Big\}. \tag{3.6}$$

This definition ensures the global continuity of the trial functions for the velocity. This is due to the fact that the spatial degrees of freedom which vanish when stepping from H_h^{m-1} to H_h^m are coupled only with the temporal basis function τ_0 which is zero at t_m. On the other hand, the spatial degrees of freedom in H_h^m appearing when coming from H_h^{m-1} are coupled only with temporal basis functions τ_i that disappear at t_{m-1}.

For convenience of the reader, the actual distribution of degrees of freedom in the case of a cG(1)cG(1) discretization in one spatial dimension is illustrated in Figure 3.6.

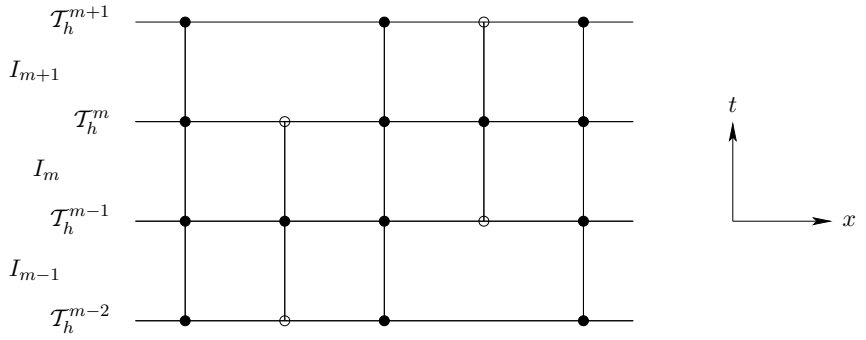


Figure 3.6. Hanging nodes in time: Distribution of degrees of freedom for the cG(1)cG(1) discretization in one spatial dimension (•: degree of freedom, ∘: no degree of freedom)

Remark 3.5. A possible choice for a basis $\{ \tau_0, \ldots, \tau_r \}$ of $\mathcal{P}_r(I_m, \mathbb{R})$ which fulfills the prerequisites is the Lagrange basis. In the case $r = 1$, this results in

$$\tau_0(t) = \frac{t_m - t}{k_m} \quad \text{and} \quad \tau_1(t) = \frac{t - t_{m-1}}{k_m}.$$

Remark 3.6. If all meshes are identical and hence all spatial finite element spaces coincide, i. e., $H_h^m = H_h$ and $L_h^m = L_h$, $m = 0, \ldots, M$, the definition of $X_{kh}^{r,s}$ given in (3.6) reduces to the following more familiar one:

$$X_{kh}^{r,s} := \Big\{ \boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in C(\bar{I}, L^2(\Omega)^d) \times L^2(I, L^2(\Omega)/\mathbb{R}) \;\Big|\; \boldsymbol{v_{kh}}\Big|_{I_m} \in \mathcal{P}_r(I_m, H_h^d), \\ p_{kh}\Big|_{I_m} \in \mathcal{P}_{r-1}(I_m, L_h),\ m = 1, \ldots, M \Big\}$$

We now state the cG(s)cG(r) formulation of problem (2.10) as follows: Find $\boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in X_{kh}^{r,s}$ such that

$$\int_I (\partial_t \boldsymbol{v_{kh}}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + (\boldsymbol{v_{kh}}(0), \boldsymbol{\psi}_0^-) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_{kh}^{r-1,s}. \quad (3.7)$$

Both fully discrete formulations cG(s)dG(r) and cG(s)cG(r) do not lead to a stable approximation of problem (2.10) unless the spatial finite element spaces H_h^m and L_h^m fulfill the Babuška-Brezzi inf-sup-stability condition. This condition states (see, for example, Girault and Raviart [58]) that there is a constant β independent of h such that

$$\inf_{p_h \in L_h^m} \sup_{\boldsymbol{v_h} \in (H_h^m)^d} \frac{(p_h, \nabla \cdot \boldsymbol{v_h})}{\|p_h\| \, \|\nabla \boldsymbol{v_h}\|} \geq \beta > 0. \quad (3.8)$$

Especially the cases of equal-order trial spaces, i. e., $s_{\boldsymbol{v}} = s_p = s$, which are favorable from the implementational point of view, do not fulfill condition (3.8). To obtain stable approximations using the presented fully discrete formulations, one has to use mixed finite element methods like the Taylor-Hood element ($s_{\boldsymbol{v}} = 2$, $s_p = 1$, for example). For more details on this topic, we refer to Hood and Taylor [69] and Brezzi and Fortin [25] or Girault and Raviart [58].

3.3 Stabilization

As already mentioned, it is desirable for implementational reasons to use the same trial functions for both velocity and pressure. Such combinations of trial spaces do not fulfill the Babuška-Brezzi condition (3.8). In this section, we present two modifications of the cG(s)dG(r) and cG(s)cG(r) formulation which will allow us to obtain a stable approximation of the continuous problem (2.10) using equal-order trial functions. This is done by adding stabilization terms to the fully discrete formulations. In addition to the already mentioned instability for equal-order trial functions for velocity and pressure, we also have to add stabilization terms in the case of higher Reynolds number, i. e., convection dominant flows, since the pure Galerkin formulation there also suffers from instability.

3.3.1 Residual based stabilization

The first modification goes back to ideas of Brooks and Hughes [26] and Hughes, Franca, and Balestra [70] who added mesh-dependent residual terms to the Galerkin formulation in order to obtain a stable discretization. Since this modification includes streamline diffusion terms for the stabilization of the convection terms as well as pressure stabilization due to the missing inf-sup-stability for equal-order trial functions, it is often referred to as

streamline upwind Petrov-Galerkin (SUPG)/pressure stabilization Petrov-Galerkin (PSPG) method. In the case of the nonstationary Navier-Stokes equations, the additional terms are given as

$$s_{\mathrm{SUPG/PSPG}}(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) \coloneqq \sum_{m=1}^{M} \Bigg\{ \int_{I_m} \sum_{K \in \mathcal{T}_h^m} (\partial_t \boldsymbol{v_{kh}} - \nu \Delta \boldsymbol{v_{kh}} + (\boldsymbol{v_{kh}} \cdot \nabla)\boldsymbol{v_{kh}} + \nabla p_{kh} - \boldsymbol{f}, \alpha_{K,m} \nabla \chi + \delta_{K,m} (\boldsymbol{v_{kh}} \cdot \nabla)\boldsymbol{\psi})_K \Bigg\}.$$

The term $(\nabla p_{kh}, \nabla \chi)_K$ represents the stabilization due to equal-order trial functions, whereas the term $((\boldsymbol{v_{kh}} \cdot \nabla)\boldsymbol{v_{kh}}, (\boldsymbol{v_{kh}} \cdot \nabla)\boldsymbol{\psi})_K$ is a streamline diffusion term. All other terms are just present to provide consistency, i. e., the additional terms vanish if we insert the continuous solution $\boldsymbol{u} = (\boldsymbol{v}, p)^T$. The parameters $\alpha_{K,m}$ and $\delta_{K,m}$ are given cell-wise as

$$\alpha_{K,m} = \alpha_0 \frac{h_K^2}{6\nu + h_K \left\| \boldsymbol{v_{kh}} \right\|_K} \quad \text{and} \quad \delta_{K,m} = \delta_0 \frac{h_K^2}{6\nu + h_K \left\| \boldsymbol{v_{kh}}_K \right\| + \frac{h_K}{k_m}} \tag{3.9}$$

with some constants α_0 and δ_0. For details on the choice of these parameters, we refer, for instance, to Franca and Frey [52] or Braack, Burman, John, and Lube [19].

The modified fully discrete formulations then read as follows:

cG(s)dG(r): Find $\boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in \widetilde{X}_{kh}^{r,s}$ such that

$$\sum_{m=1}^{M} \int_{I_m} (\partial_t \boldsymbol{v_{kh}}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + s_{\mathrm{SUPG/PSPG}}(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + \sum_{m=0}^{M-1} ([\boldsymbol{v_{kh}}]_m, \boldsymbol{\psi}_m^+) + (\boldsymbol{v}_{\boldsymbol{kh},0}^-, \boldsymbol{\psi}_0^-) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_{kh}^{r,s}. \tag{3.10}$$

cG(s)cG(r): Find $\boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in X_{kh}^{r,s}$ such that

$$\int_I (\partial_t \boldsymbol{v_{kh}}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + s_{\mathrm{SUPG/PSPG}}(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + (\boldsymbol{v_{kh}}(0), \boldsymbol{\psi}_0^-) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_{kh}^{r-1,s}. \tag{3.11}$$

A well known drawback of this kind of stabilization is the introduction of boundary layers in the pressure approximation due to the artificial boundary condition

$$\partial_{\boldsymbol{n}} p_{kh} = 0 \quad \text{on } \partial\Omega$$

which arises from the stabilization terms and is fulfilled in the discrete sense. This leads to a decrease in accuracy near the boundary.

From the computational point of view, another disadvantage is the necessity to compute second derivatives appearing in the additional terms because for higher order trial functions ($s > 1$) or even for bi- or tri-linear finite elements on arbitrary quadrilaterals or hexahedrons they do not vanish. This is especially serious because they are only needed for the consistency terms in the stabilization and not even for the pure Galerkin formulation. However, neglecting them would result in a decreased accuracy.

Furthermore, these additional terms possess a quite nasty algebraic structure: Artificial non-symmetric terms are introduced as well as artificial couplings between velocity and pressure. This has a negative influence on the behavior of the solver of the algebraic equations arising from these discretizations.

3.3.2 Local projection stabilization

In Becker and Braack [10, 11], a new stabilization based on local projections was proposed. To give a precise definition of the modified fully discrete formulations, we introduce a spatial interpolation operator $I_h : V_h^{s,m} \to \widetilde{V}_h^{s,m}$ into a subspace $\widetilde{V}_h^{s,m} \subseteq V_h^{s,m}$ which is given as

$$\widetilde{V}_h^{s,m} := \begin{cases} V_{2h}^{1,m} & \text{for } s = 1, \\ V_h^{1,m} & \text{for } s = 2. \end{cases}$$

The interpolation onto the mesh $\mathcal{T}_{2h}$ in the case $s = 1$ is easily computable if the mesh possesses the patch structure introduced in Section 3.2. Using the interpolation operator I_h, we define the filtering operator $\pi : V_h^{s,m} \to V_h^{s,m}$ by

$$\pi := \mathrm{id} - I_h.$$

The filtering operator $\boldsymbol{\pi} : (V_h^{s,m})^d \to (V_h^{s,m})^d$ is defined analogously component-wise. Let us further extend these operators in time by defining point-wise

$$(\pi p_{kh})(t) := \pi p_{kh}(t) \quad \text{and} \quad (\boldsymbol{\pi u_{kh}})(t) := \boldsymbol{\pi u_{kh}}(t).$$

This allows us to state the following modified fully discrete formulations of problem (2.10):

cG(s)dG(r): Find $\boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in \widetilde{X}_{kh}^{r,s}$ such that

$$\sum_{m=1}^{M} \int_{I_m} (\partial_t \boldsymbol{v_{kh}}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + s_{\mathrm{LPS}}(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + \sum_{m=0}^{M-1} ([\boldsymbol{v_{kh}}]_m, \boldsymbol{\psi}_m^+) + (\boldsymbol{v}_{\boldsymbol{kh},0}^-, \boldsymbol{\psi}_0^-) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_{kh}^{r,s}. \quad (3.12)$$

cG(s)cG(r): Find $\boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T \in X_{kh}^{r,s}$ such that

$$\int_I (\partial_t \boldsymbol{v_{kh}}, \boldsymbol{\psi}) \, \mathrm{d}t + a(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + s_{\mathrm{LPS}}(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + (\boldsymbol{v_{kh}}(0), \boldsymbol{\psi}_0^-) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t + (\boldsymbol{v^0}, \boldsymbol{\psi}_0^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_{kh}^{r-1,s}. \quad (3.13)$$

Here, the additional terms are given by

$$s_{\mathrm{LPS}}(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) \coloneqq \sum_{m=1}^{M} \int_{I_m} s_h^m(\boldsymbol{u_{kh}}(t))(\boldsymbol{\varphi}(t)) \, \mathrm{d}t,$$

where

$$s_h^m(\boldsymbol{U})(\boldsymbol{\Phi}) \coloneqq \sum_{K \in \mathcal{T}_h^m} \left\{ (\nabla \pi P, \alpha_{K,m} \nabla \pi X)_K + ((\boldsymbol{V} \cdot \nabla) \boldsymbol{\pi V}, \delta_{K,m} (\boldsymbol{V} \cdot \nabla) \boldsymbol{\pi \Psi})_K \right\}$$

with $\boldsymbol{U} = (\boldsymbol{V}, P)^T$ and $\boldsymbol{\Phi} = (\boldsymbol{\Psi}, X)^T$. The cell-wise stabilization parameters $\alpha_{K,m}$ and $\delta_{K,m}$ are given in the same way as in (3.9).

The structure of the stabilization terms is very easy. No artificial couplings are introduced and we do not need to compute second derivatives of the trial functions. The boundary layer in the approximation of the pressure mentioned in the last section is not present in this case. However, this stabilization is not consistent, i. e., the continuous solution does not fulfill the modified fully discrete formulations, but the additional error is of the same order as the discretization error.

This is why, in our computations, we always use this local projection stabilization technique. For convenience, we introduce the abbreviations

$$a_h(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) \coloneqq \sum_{m=1}^{M} \int_{I_m} a_h^m(\boldsymbol{u_{kh}}(t))(\boldsymbol{\varphi}(t)) \, \mathrm{d}t$$

with

$$a_h^m(\boldsymbol{U})(\boldsymbol{\Phi}) \coloneqq \bar{a}(\boldsymbol{U})(\boldsymbol{\Phi}) + s_h^m(\boldsymbol{U})(\boldsymbol{\Phi})$$

such that

$$a_h(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) = a(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) + s_{\mathrm{LPS}}(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}).$$

3.4 Formulation as time-stepping schemes

In this section, we present the precise formulation as time-stepping schemes of the space-time Galerkin finite element discretizations of the incompressible Navier-Stokes equations presented in the last section. In this thesis, we will consider the cases cG(s)dG(0) (which

corresponds to a variant of the backward Euler scheme), cG(s)dG(1), and cG(s)cG(1) (which actually is a variant of the Crank-Nicolson scheme). For the spatial discretization, we will consider $s = 1$ (bi- or tri-linear finite elements) and $s = 2$ (bi- or tri-quadratic finite elements) in combination with the local projection stabilization.

Each time step results in a quasi-stationary nonlinear problem. These nonlinear problems are solved with a quasi-Newton iteration. For details on solving the linear subproblems occurring within the quasi-Newton iteration, we refer to Section 3.5.3.

3.4.1 cG(s)dG(0) discretization

In the case $r = 0$, we use trial and test functions which are piecewise constant in time. Hence, we have $\partial_t \boldsymbol{v_{kh}}\big|_{I_m} = 0$. Introducing the notation

$$\boldsymbol{U_m} = (\boldsymbol{V_m}, P_m)^T \quad \text{with} \quad \boldsymbol{V_m} \coloneqq \boldsymbol{v}^-_{\boldsymbol{kh},m}, \quad P_m \coloneqq p^-_{kh,m},$$

we obtain the following set of equations that should be valid for all $\boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in (H_h^m)^d \times L_h^m$:

$\boldsymbol{m = 0}$:

$$(\boldsymbol{V_0}, \boldsymbol{\psi}) = (\boldsymbol{v^0}, \boldsymbol{\psi})$$

$\boldsymbol{m = 1, \ldots, M}$:

$$\begin{aligned}
&(\boldsymbol{V_m}, \boldsymbol{\psi}) + k_m \Big\{ \nu(\nabla \boldsymbol{V_m}, \nabla \boldsymbol{\psi}) + ((\boldsymbol{V_m} \cdot \nabla)\boldsymbol{V_m}, \boldsymbol{\psi}) - (P_m, \nabla \cdot \boldsymbol{\psi}) \\
&\qquad + \sum_{K \in \mathcal{T}_h^m} \delta_{K,m} ((\boldsymbol{V_m} \cdot \nabla)\boldsymbol{\pi V_m}, (\boldsymbol{V_m} \cdot \nabla)\boldsymbol{\pi \psi})_K \Big\} = (\boldsymbol{V_{m-1}}, \boldsymbol{\psi}) + \int_{I_m} (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t \\
&k_m \Big\{ (\nabla \cdot \boldsymbol{V_m}, \chi) + \sum_{K \in \mathcal{T}_h^m} \alpha_{K,m} (\nabla \pi P_m, \nabla \pi \chi)_K \Big\} = 0
\end{aligned}$$

If we approximate the temporal integral by the box rule, i. e.,

$$\int_{I_m} (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t \approx k_m (\boldsymbol{f}(t_m), \boldsymbol{\psi}),$$

we obtain the backward Euler scheme.

The backward Euler scheme is known to be of first order and strongly A-stable. In Eriksson and Johnson [46], the authors show for a scalar linear ordinary differential equation that the approximation of the temporal integral by the box rule has disadvantages compared to the exact evaluation of the integral. Thus, especially for long time integrations, the use of higher order quadrature is advisable.

3.4.2 cG(*s*)dG(1) discretization

In the case $r = 1$, $\boldsymbol{v_{kh}}$ and p_{kh} can be expressed on each subinterval I_m as

$$\boldsymbol{v_{kh}}\Big|_{I_m} = \frac{t - t_{m-1}}{k_m}\boldsymbol{V_m^-} + \frac{t_m - t}{k_m}\boldsymbol{V_{m-1}^+} \quad \text{and} \quad p_{kh}\Big|_{I_m} = \frac{t - t_{m-1}}{k_m}P_m^- + \frac{t_m - t}{k_m}P_{m-1}^+$$

with the abbreviations

$$\boldsymbol{V_m^-} \coloneqq \boldsymbol{v_{kh,m}^-}, \quad \boldsymbol{V_{m-1}^+} \coloneqq \boldsymbol{v_{kh,m-1}^+}, \quad P_m^- \coloneqq p_{kh,m}^-, \quad P_{m-1}^+ \coloneqq p_{kh,m-1}^+.$$

Hence

$$\partial_t \boldsymbol{v_{kh}}\Big|_{I_m} = \frac{1}{k_m}(\boldsymbol{V}_m^- - \boldsymbol{V}_{m-1}^+).$$

Let us now derive the time-stepping formulation. The following equations should again be valid for all $\boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in (H_h^m)^d \times L_h^m$:

$\boldsymbol{m = 0}$: As in the case $r = 0$, we obtain for $t = 0$:

$$(\boldsymbol{V_0^-}, \boldsymbol{\psi}) = (\boldsymbol{v^0}, \boldsymbol{\psi}).$$

$\boldsymbol{m = 1, \ldots, M}$: Choosing the test function

$$\boldsymbol{\varphi_1}(t) \coloneqq \begin{cases} \dfrac{t - t_{m-1}}{k_m}\boldsymbol{\varphi} & \text{for } t \in I_m, \\ 0 & \text{otherwise} \end{cases}$$

with $\boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in (H_h^m)^d \times L_h^m$, we obtain the equation

$$\frac{1}{2}(\boldsymbol{V_m^-} - \boldsymbol{V_{m-1}^+}, \boldsymbol{\psi}) + \int_{I_m} a_h^m(\boldsymbol{u_{kh}})(\boldsymbol{\varphi_1})\,\mathrm{d}t = \int_{I_m} \left(\boldsymbol{f}, \frac{t - t_{m-1}}{k_m}\boldsymbol{\psi}\right)\mathrm{d}t. \tag{3.14a}$$

Similarly for

$$\boldsymbol{\varphi_2}(t) \coloneqq \begin{cases} \dfrac{t_m - t}{k_m}\boldsymbol{\varphi} & \text{for } t \in I_m, \\ 0 & \text{otherwise,} \end{cases}$$

we obtain

$$\frac{1}{2}(\boldsymbol{V_m^-} + \boldsymbol{V_{m-1}^+}, \boldsymbol{\psi}) + \int_{I_m} a_h^m(\boldsymbol{u_{kh}})(\boldsymbol{\varphi_2})\,\mathrm{d}t = (\boldsymbol{V_{m-1}^-}, \boldsymbol{\psi}) + \int_{I_m} \left(\boldsymbol{f}, \frac{t_m - t}{k_m}\boldsymbol{\psi}\right)\mathrm{d}t. \tag{3.14b}$$

The cG(s)dG(1) discretization is of second order in time and strongly A-stable. Actually, in the time points t_m, this method shows a super-convergence property of order three. For a detailed analysis in the context of ordinary differential equations, see, for example, Johnson [75].

For implementational aspects of assembling and solving the system of equations (3.14), we refer to Section 3.5.

3.4.3 cG(*s*)cG(1) discretization

Here, we use trial functions which are continuous piecewise linear in time for the velocity and discontinuous piecewise constant for the pressure. The test functions for both unknowns are discontinuous piecewise constant in time. Hence, also this globally formulated discretization reduces to a time-stepping scheme. Using the notation

$$\boldsymbol{V_m} \coloneqq \boldsymbol{v_{kh}}(t_m) \quad \text{and} \quad P_m \coloneqq p^-_{kh,m}$$

and noting that

$$\partial_t \boldsymbol{v_{kh}}\Big|_{I_m} = \frac{1}{k_m}(\boldsymbol{V_m} - \boldsymbol{V_{m-1}}),$$

we obtain the following system of equations which should be valid for all $\boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in (H_h^m)^d \times L_h^m$:

$m = 0$:

$$(\boldsymbol{V_0}, \boldsymbol{\psi}) = (\boldsymbol{v^0}, \boldsymbol{\psi})$$

$m = 1, \ldots, M$:

$$(\boldsymbol{V_m}, \boldsymbol{\psi}) + \int_{I_m} a_h^m(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) \, \mathrm{d}t = (\boldsymbol{V_{m-1}}, \boldsymbol{\psi}) + \int_{I_m} (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t$$

If we approximate the temporal integrals by the trapezoidal rule

$$\int_{I_m} f(t) \, \mathrm{d}t \approx \frac{k_m}{2} \left\{ f(t_{m-1}) + f(t_m) \right\},$$

we obtain a variant of the well known Crank-Nicolson scheme which reads as follows:

$$\begin{aligned}
(\boldsymbol{V_m}, \boldsymbol{\psi}) + \frac{k_m}{2} \Big\{ & \nu(\nabla \boldsymbol{V_m}, \nabla \boldsymbol{\psi}) + ((\boldsymbol{V_m} \cdot \nabla)\boldsymbol{V_m}, \boldsymbol{\psi}) \\
& + \sum_{K \in \mathcal{T}_h^m} \delta_{K,m} ((\boldsymbol{V_m} \cdot \nabla)\boldsymbol{\pi V_m}, (\boldsymbol{V_m} \cdot \nabla)\boldsymbol{\pi \psi})_K \Big\} - k_m (P_m, \nabla \cdot \boldsymbol{\psi}) \\
& = (\boldsymbol{V_{m-1}}, \boldsymbol{\psi}) - \frac{k_m}{2} \Big\{ \nu(\nabla \boldsymbol{V_{m-1}}, \nabla \boldsymbol{\psi}) + ((\boldsymbol{V_{m-1}} \cdot \nabla)\boldsymbol{V_{m-1}}, \boldsymbol{\psi}) \\
& + \sum_{K \in \mathcal{T}_h^m} \delta_{K,m} ((\boldsymbol{V_{m-1}} \cdot \nabla)\boldsymbol{\pi V_{m-1}}, (\boldsymbol{V_{m-1}} \cdot \nabla)\boldsymbol{\pi \psi})_K \Big\} + \frac{k_m}{2} (\boldsymbol{f}(t_{m-1}) + \boldsymbol{f}(t_m), \boldsymbol{\psi}) \\
& \frac{k_m}{2} (\nabla \cdot \boldsymbol{V_m}, \chi) + k_m \sum_{K \in \mathcal{T}_h^m} \alpha_{K,m} (\nabla \pi P_m, \nabla \pi \chi)_K = -\frac{k_m}{2} (\nabla \cdot \boldsymbol{V_{m-1}}, \chi)
\end{aligned}$$

The Crank-Nicolson scheme is known to be of second order, but in contrast to time-stepping schemes resulting from dG(r) semi-discretizations in time it is only A-stable, not strongly A-stable.

3.5 Implementational aspects

This section is devoted to the discussion of some implementational aspects coming up when dealing with the discretizations presented in the previous sections. These are mainly the questions on how to do computations on dynamic meshes, on assembling the system of equations arising in the cG(s)dG(1) time-stepping formulation, and on how to solve the linear subproblems arising within the quasi-Newton iteration. Each topic is discussed in a separate section below.

3.5.1 Computations on dynamic meshes

The discussion in this section is done exemplarily for the cG(s)dG(r) discretizations. However, the cG(s)cG(r) discretizations can be handled in a similar way.

When doing computations on dynamic meshes, the main difficulty is the evaluation of the inner product $(\boldsymbol{V}_{\boldsymbol{m-1}}^{-}, \boldsymbol{\psi})$. Since $\boldsymbol{V}_{\boldsymbol{m-1}}^{-}$ is given as coefficient vector with respect to the nodal basis of $(V_h^{s,m-1})^d$, this problem reduces to the evaluation of inner products of basis functions $\psi^{m-1} \in V_h^{s,m-1}$ with basis functions $\psi^m \in V_h^{s,m}$. These integrals cannot be evaluated cell-wise on $\mathcal{T}_h^m$ by applying quadrature rules, since ψ^{m-1} is not necessarily smooth on each cell $K \in \mathcal{T}_h^m$.

To overcome this difficulty, we require that all meshes $\mathcal{T}_h^m$ originate from one common mesh $\widetilde{\mathcal{T}_h}$ by hierarchical refinement or coarsening. Thus, we can build a temporary mesh $\mathcal{T}_h^{m-\frac{1}{2}}$ as the common refinement of $\mathcal{T}_h^{m-1}$ and $\mathcal{T}_h^m$, see Figure 3.7.

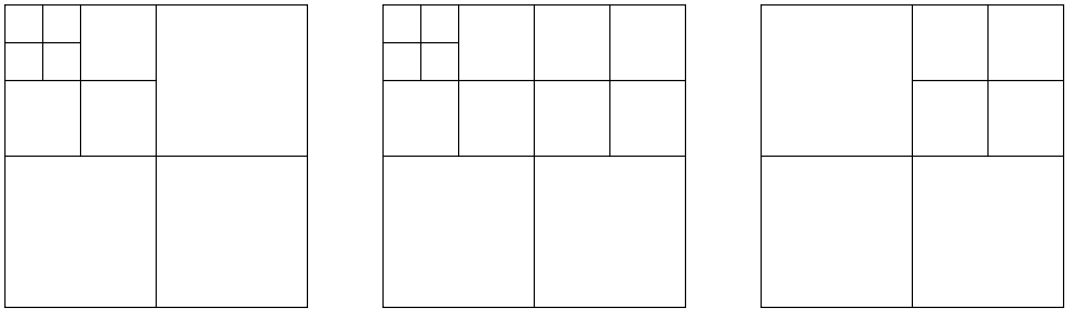


Figure 3.7. Construction of temporary mesh $\mathcal{T}_h^{m-\frac{1}{2}}$

Looking at the corresponding finite element spaces, there holds by construction $V_h^{s,m-1} \subseteq V_h^{s,m-\frac{1}{2}}$ as well as $V_h^{s,m} \subseteq V_h^{s,m-\frac{1}{2}}$. Therefore, we can express both ψ^{m-1} and ψ^m as linear combinations of nodal basis functions $\psi_i^{m-\frac{1}{2}}$ of $V_h^{s,m-\frac{1}{2}}$:

$$\psi^{m-1} = \sum_{i=1}^{N} \xi_i^{(m-1)} \psi_i^{m-\frac{1}{2}}, \quad \psi^m = \sum_{i=1}^{N} \xi_i^{(m)} \psi_i^{m-\frac{1}{2}},$$

where $\xi_i^{(m-1)}, \xi_i^{(m)} \in \mathbb{R}$ and $N = \dim V_h^{s,m-\frac{1}{2}}$. Since $V_h^{s,m-1}$, $V_h^{s,m}$, and $V_h^{s,m-\frac{1}{2}}$ are finite element spaces, these linear combinations can be computed locally on patches because every basis function has only local support. For example, in the case of bi-linear trial functions on the reference cell $\hat{K} = (0,1)^2$, we have

$$\begin{pmatrix} \psi_1^{m-1} \\ \psi_2^{m-1} \\ \psi_3^{m-1} \\ \psi_4^{m-1} \end{pmatrix} = \begin{pmatrix} 1 & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 1 & 0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{4} & 0 & 1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} \psi_1^{m-\frac{1}{2}} \\ \psi_2^{m-\frac{1}{2}} \\ \psi_3^{m-\frac{1}{2}} \\ \psi_4^{m-\frac{1}{2}} \\ \psi_5^{m-\frac{1}{2}} \\ \psi_6^{m-\frac{1}{2}} \\ \psi_7^{m-\frac{1}{2}} \\ \psi_8^{m-\frac{1}{2}} \\ \psi_9^{m-\frac{1}{2}} \end{pmatrix}.$$

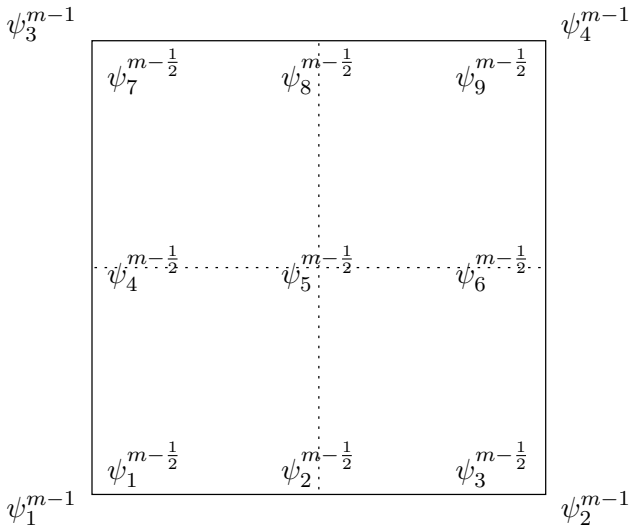


Figure 3.8. Representation of basis functions on a coarse mesh by basis functions of a finer mesh in the case of bi-linear finite elements

See Figure 3.8 for the numbering of the corresponding basis functions. Of course, this matrix—which is identical for all elements in the mesh—is not computed, but the representation is rather assembled by, e. g.,

$$\psi_2^{m-1} = \frac{1}{2}\psi_2^{m-\frac{1}{2}} + \psi_3^{m-\frac{1}{2}} + \frac{1}{4}\psi_5^{m-\frac{1}{2}} + \frac{1}{2}\psi_6^{m-\frac{1}{2}}.$$

After these preparations, we can evaluate the inner product (ψ^{m-1}, ψ^m) as

$$(\psi^{m-1}, \psi^m) = \sum_{i=1}^{N} \xi_i^{(m-1)} (\psi_i^{m-\frac{1}{2}}, \psi^m) = \sum_{i,j=1}^{N} \xi_i^{(m-1)} \xi_j^{(m)} (\psi_i^{m-\frac{1}{2}}, \psi_j^{m-\frac{1}{2}}),$$

where the inner products $(\psi_i^{m-\frac{1}{2}}, \psi_j^{m-\frac{1}{2}})$ can be evaluated cell-wise on $\mathcal{T}_h^{m-\frac{1}{2}}$ by applying quadrature rules since $\psi_i^{m-\frac{1}{2}}$ and $\psi_j^{m-\frac{1}{2}}$ are smooth on each cell $K \in \mathcal{T}_h^{m-\frac{1}{2}}$.

3.5.2 Assembling and solving the system of equations in the time-stepping formulation of the cG(*s*)dG(1) method

In this section, some details on the process of assembling and solving the arising system of equations in the time-stepping formulation of the cG(s)dG(1) discretization are given.

Let us first recall the system of equations to be solved on the subinterval I_m: Given $\boldsymbol{V}_{\boldsymbol{m-1}}^-$, find $\boldsymbol{u_{kh}}\big|_{I_m} = (\boldsymbol{v_{kh}}, p_{kh})^T\big|_{I_m} \in \mathcal{P}_1(I_m, (H_h^m)^d \times L_h^m)$ such that for all $\boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in (H_h^m)^d \times L_h^m$ there holds

$$\frac{1}{2}(\boldsymbol{V}_{\boldsymbol{m}}^- - \boldsymbol{V}_{\boldsymbol{m-1}}^+, \boldsymbol{\psi}) + \int_{I_m} a_h^m(\boldsymbol{u_{kh}})(\tfrac{t-t_{m-1}}{k_m}\boldsymbol{\varphi})\,\mathrm{d}t = \int_{I_m} (\boldsymbol{f}, \tfrac{t-t_{m-1}}{k_m}\boldsymbol{\psi})\,\mathrm{d}t$$

and

$$\frac{1}{2}(\boldsymbol{V}_{\boldsymbol{m}}^- + \boldsymbol{V}_{\boldsymbol{m-1}}^+, \boldsymbol{\psi}) + \int_{I_m} a_h^m(\boldsymbol{u_{kh}})(\tfrac{t_m-t}{k_m}\boldsymbol{\varphi})\,\mathrm{d}t = (\boldsymbol{V}_{\boldsymbol{m-1}}^-, \boldsymbol{\psi}) + \int_{I_m} (\boldsymbol{f}, \tfrac{t_m-t}{k_m}\boldsymbol{\psi})\,\mathrm{d}t.$$

For solving this system of equations, we apply Newton's method. Given an approximation $\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}$ for $\boldsymbol{u_{kh}}$ on I_m, we compute a correction $\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}$ as the solution of the following system of equations:

$$\begin{aligned}\frac{1}{2}(\boldsymbol{\delta V}_{\boldsymbol{m}}^{(\boldsymbol{i})-} - \boldsymbol{\delta V}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \boldsymbol{\psi}) + \int_{I_m} a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t-t_{m-1}}{k_m}\boldsymbol{\varphi})\,\mathrm{d}t &= \int_{I_m} (\boldsymbol{f}, \tfrac{t-t_{m-1}}{k_m}\boldsymbol{\psi})\,\mathrm{d}t \\ &\quad - \frac{1}{2}(\boldsymbol{V}_{\boldsymbol{m}}^{(\boldsymbol{i})-} - \boldsymbol{V}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \boldsymbol{\psi}) - \int_{I_m} a_h^m(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\tfrac{t-t_{m-1}}{k_m}\boldsymbol{\varphi})\,\mathrm{d}t \quad (3.15a)\end{aligned}$$

$$\begin{aligned}\frac{1}{2}(\boldsymbol{\delta V}_{\boldsymbol{m}}^{(\boldsymbol{i})-} + \boldsymbol{\delta V}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \boldsymbol{\psi}) + \int_{I_m} a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t_m-t}{k_m}\boldsymbol{\varphi})\,\mathrm{d}t &= (\boldsymbol{V}_{\boldsymbol{m-1}}^-, \boldsymbol{\psi}) + \int_{I_m} (\boldsymbol{f}, \tfrac{t_m-t}{k_m}\boldsymbol{\psi})\,\mathrm{d}t \\ &\quad - \frac{1}{2}(\boldsymbol{V}_{\boldsymbol{m}}^{(\boldsymbol{i})-} + \boldsymbol{V}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \boldsymbol{\psi}) - \int_{I_m} a_h^m(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\tfrac{t_m-t}{k_m}\boldsymbol{\varphi})\,\mathrm{d}t \quad (3.15b)\end{aligned}$$

Here, we have used the notation

$$\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}\Big|_{I_m} = \frac{t - t_{m-1}}{k_m}\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-} + \frac{t_m - t}{k_m}\boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}$$

with

$$\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-} = (\boldsymbol{\delta V}_{\boldsymbol{m}}^{(\boldsymbol{i})-}, \delta P_m^{(i)-})^T \quad \text{and} \quad \boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+} = (\boldsymbol{\delta V}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \delta P_{m-1}^{(i)+})^T$$

and the directional derivatives

$$a_h^{m\prime}(\boldsymbol{u})(\boldsymbol{\delta u}, \boldsymbol{\varphi}) \coloneqq \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\Big\{a_h^m(\boldsymbol{u} + \varepsilon\boldsymbol{\delta u})(\boldsymbol{\varphi}) - a_h^m(\boldsymbol{u})(\boldsymbol{\varphi})\Big\} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} a_h^m(\boldsymbol{u} + \varepsilon\boldsymbol{\delta u})(\boldsymbol{\varphi})\Big|_{\varepsilon=0}.$$

Afterwards we set $\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i+1})} \coloneqq \boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})} + \boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}$ and repeat the iteration with i replaced by $i+1$ until convergence.

The temporal integrals in equations (3.15a) and (3.15b) are approximated by quadrature rules:

$$\int_{I_m} a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t-t_{m-1}}{k_m}\boldsymbol{\varphi}) \approx k_m \sum_{l=1}^{n} \omega_l a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1} + k_m x_l))(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1} + k_m x_l), x_l\boldsymbol{\varphi}),$$

where $x_l \in [0,1]$ denote the integration points and $\omega_l \in \mathbb{R}$ the corresponding weights. A similar formula holds for

$$\int_{I_m} a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t_m-t}{k_m}\boldsymbol{\varphi}).$$

Noting that

$$\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1} + k_m x_l) = x_l\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-} + (1 - x_l)\boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+},$$

we see that by linearity

$$\begin{aligned} a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1}+k_m x_l))(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1}+k_m x_l), x_l\boldsymbol{\varphi}) &= x_l^2 a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1}+k_m x_l))(\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-}, \boldsymbol{\varphi}) \\ &\quad + x_l(1 - x_l) a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1} + k_m x_l))(\boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \boldsymbol{\varphi}). \end{aligned}$$

Since $\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-}, \boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+} \in (H_h^m)^d \times L_h^m$, both can be expressed as a linear combination of the basis functions:

$$\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-} = \sum_{j=1}^{N_m} \xi_j^- \boldsymbol{\varphi_j}, \quad \boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+} = \sum_{j=1}^{N_m} \xi_j^+ \boldsymbol{\varphi_j}.$$

Hence, for assembling the matrix which determines the coefficients ξ_j^- and ξ_j^+, one has to compute the terms

$$a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}(t_{m-1} + k_m x_l))(\boldsymbol{\varphi_k}, \boldsymbol{\varphi_j})$$

only once for each l and simply put them into the system matrix with the right scaling.

The computational effort can be further reduced by not using the exact Jacobian within the Newton iteration. If we fix the linearization point and approximate

$$\int_{I_m} a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t-t_{m-1}}{k_m}\boldsymbol{\varphi})\ \mathrm{d}t \approx \int_{I_m} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t-t_{m-1}}{k_m}\boldsymbol{\varphi})\ \mathrm{d}t,$$
$$\int_{I_m} a_h^{m\prime}(\boldsymbol{u}_{\boldsymbol{kh}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t_m-t}{k_m}\boldsymbol{\varphi})\ \mathrm{d}t \approx \int_{I_m} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t_m-t}{k_m}\boldsymbol{\varphi})\ \mathrm{d}t,$$

with $\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})} \coloneqq \frac{1}{2}(\boldsymbol{U}_{\boldsymbol{m}}^{(\boldsymbol{i})-} + \boldsymbol{U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+})$, we can exactly compute

$$\int_{I_m} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t-t_{m-1}}{k_m}\boldsymbol{\varphi})\ \mathrm{d}t = \frac{k_m}{3} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-}, \boldsymbol{\varphi}) + \frac{k_m}{6} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \boldsymbol{\varphi}),$$
$$\int_{I_m} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta u}_{\boldsymbol{kh}}^{(\boldsymbol{i})}, \tfrac{t_m-t}{k_m}\boldsymbol{\varphi})\ \mathrm{d}t = \frac{k_m}{6} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta U}_{\boldsymbol{m}}^{(\boldsymbol{i})-}, \boldsymbol{\varphi}) + \frac{k_m}{3} a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\delta U}_{\boldsymbol{m-1}}^{(\boldsymbol{i})+}, \boldsymbol{\varphi}).$$

Hence, in this case, the terms

$$a_h^{m\prime}(\hat{\boldsymbol{U}}_{\boldsymbol{m}}^{(\boldsymbol{i})})(\boldsymbol{\varphi_k}, \boldsymbol{\varphi_j})$$

have to be computed only once and put into the system matrix with the appropriate scaling.

3.5.3 Solving the linear subproblems

In this section, we give some details on solving the linear subproblems arising in each Newton step.

To this end, we write the nonlinear quasi-stationary problem which corresponds to one time step of the cG(s)dG(0) or cG(s)cG(1) discretization as: Find $\boldsymbol{U_m} \in (H_h^m)^d \times L_h^m$ such that

$$b(\boldsymbol{U_m})(\boldsymbol{\varphi}) = g(\boldsymbol{\varphi}) \quad \forall \boldsymbol{\varphi} \in (H_h^m)^d \times L_h^m$$

where the right-hand side g depends on already known quantities from the last time-step and the original right-hand side $\boldsymbol{f}$. Applying Newton's method to this nonlinear problem reads: Given an initial guess $\boldsymbol{U^{(0)}}$, find for $l = 0, 1, 2, \ldots$ the solution $\boldsymbol{\delta U^{(l)}} \in (H_h^m)^d \times L_h^m$ of

$$b'(\boldsymbol{U}^{(\boldsymbol{l})})(\boldsymbol{\delta U}^{(\boldsymbol{l})}, \boldsymbol{\varphi}) = g(\boldsymbol{\varphi}) - b(\boldsymbol{U}^{(\boldsymbol{l})})(\boldsymbol{\varphi}) \quad \forall \boldsymbol{\varphi} \in (H_h^m)^d \times L_h^m \tag{3.16}$$

and set $\boldsymbol{U^{(l+1)}} \coloneqq \boldsymbol{U^{(l)}} + \boldsymbol{\delta U^{(l)}}$. Here, $b'(\boldsymbol{U})(\boldsymbol{\delta U}, \boldsymbol{\varphi})$ again denotes the directional derivative

$$b'(\boldsymbol{U})(\boldsymbol{\delta U}, \boldsymbol{\varphi}) \coloneqq \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\Big\{ b(\boldsymbol{U} + \varepsilon \boldsymbol{\delta U})(\boldsymbol{\varphi}) - b(\boldsymbol{U})(\boldsymbol{\varphi}) \Big\} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon} b(\boldsymbol{U} + \varepsilon \boldsymbol{\delta U})(\boldsymbol{\varphi})\Big|_{\varepsilon=0}.$$

Let us for the moment ignore the Dirichlet boundary conditions for the velocity and the mean value constraint for the pressure which are both incorporated later. Employing the fact that the finite element spaces used in this thesis involve equal order polynomials for the

velocity and the pressure approximation, a basis of the finite element space $(H_h^m)^d \times L_h^m$ is given by

$$\left\{ \boldsymbol{\varphi}_i^{(p)}, \boldsymbol{\varphi}_i^{(v_1)}, \ldots, \boldsymbol{\varphi}_i^{(v_d)} \;\middle|\; i = 1, \ldots, N \right\}$$

with $N = \dim V_h^m$ and

$$\boldsymbol{\varphi}_i^{(p)} = \begin{pmatrix} \psi_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \boldsymbol{\varphi}_i^{(v_1)} = \begin{pmatrix} 0 \\ \psi_i \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \ldots, \quad \boldsymbol{\varphi}_i^{(v_d)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \psi_i \end{pmatrix},$$

where $\{ \psi_i \}$ is a basis of V_h^m. Using this basis, we can transform (3.16) into an equivalent algebraic system $\boldsymbol{B}\boldsymbol{\xi} = \boldsymbol{c}$. The right-hand side vector $\boldsymbol{c}$ is then given by

$$\boldsymbol{c} = \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix}$$

where the entries c_i itself are small vectors

$$c_i = \begin{pmatrix} g(\boldsymbol{\varphi}_i^{(p)}) - b(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_i^{(p)}) \\ g(\boldsymbol{\varphi}_i^{(v_1)}) - b(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_i^{(v_1)}) \\ \vdots \\ g(\boldsymbol{\varphi}_i^{(v_d)}) - b(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_i^{(v_d)}) \end{pmatrix}.$$

The matrix $\boldsymbol{B}$ also has the following block structure

$$\boldsymbol{B} = \begin{pmatrix} B_{11} & \ldots & B_{1,N} \\ \vdots & & \vdots \\ B_{N,1} & \ldots & B_{NN} \end{pmatrix}$$

where the blocks B_{ij} are given by

$$B_{ij} = \begin{pmatrix} b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(p)}, \boldsymbol{\varphi}_i^{(p)}) & b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(v_1)}, \boldsymbol{\varphi}_i^{(p)}) & \ldots & b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(v_d)}, \boldsymbol{\varphi}_i^{(p)}) \\ b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(p)}, \boldsymbol{\varphi}_i^{(v_1)}) & b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(v_1)}, \boldsymbol{\varphi}_i^{(v_1)}) & \ldots & b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(v_d)}, \boldsymbol{\varphi}_i^{(v_1)}) \\ \vdots & \vdots & \ddots & \vdots \\ b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(p)}, \boldsymbol{\varphi}_i^{(v_d)}) & b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(v_1)}, \boldsymbol{\varphi}_i^{(v_d)}) & \ldots & b'(\boldsymbol{U}^{(l)})(\boldsymbol{\varphi}_j^{(v_d)}, \boldsymbol{\varphi}_i^{(v_d)}) \end{pmatrix}.$$

The degrees of freedom $\xi_i^{(v_j)}$ corresponding to Dirichlet boundary conditions are now enforced strongly by replacing the corresponding rows and columns within the matrix $\boldsymbol{B}$

by

$$\begin{pmatrix} & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ & & & 0 & & & \\ & & & \vdots & & & \\ & & & 0 & & & \end{pmatrix}$$

and the corresponding entries in the right-hand side $\boldsymbol{c}$ by 0. Note that even in the case of inhomogeneous Dirichlet boundary conditions this is correct because the initial guess $\boldsymbol{U^{(0)}}$ already fulfills the right Dirichlet boundary conditions and hence the updates $\boldsymbol{\delta U^{(l)}}$ must satisfy homogeneous Dirichlet boundary conditions.

In the case of a cG(s)dG(1) discretization, we can proceed in a similar way. However, the resulting matrix $\boldsymbol{B}$ consists of $2(d+1) \times 2(d+1)$ blocks B_{ij} instead of $(d+1) \times (d+1)$ blocks because in each time step $\boldsymbol{U_{m-1}^{+}}$ and $\boldsymbol{U_{m}^{-}}$ have to be computed.

For solving the linear subproblems $\boldsymbol{B\xi} = \boldsymbol{c}$, we apply the *Generalized Minimal Residual Method* (GMRES) of Saad [94]. Actually, we solve the preconditioned system

$$\boldsymbol{CB\xi} = \boldsymbol{Cc}$$

where a multigrid iteration is applied as preconditioner $\boldsymbol{C}$. As smoother in the multigrid iteration, we use a fixed-point iteration based on a block-ILU decomposition. The idea is to compute a decomposition

$$\boldsymbol{B} = \boldsymbol{LU} + \boldsymbol{R}$$

where $\boldsymbol{L}$ is a lower and $\boldsymbol{U}$ is an upper triangular matrix. While in a full LU decomposition $\boldsymbol{L}$ and $\boldsymbol{U}$ are dense matrices, in the incomplete version they have the same structure as $\boldsymbol{B}$. Hence, we have $\boldsymbol{R} \neq \boldsymbol{0}$. However, the matrix $\boldsymbol{R}$ is not computed, but neglected within the fixed-point iteration. Hence, the fixed-point iteration performs steps of the following form:

$$\boldsymbol{\xi^{(s+1)}} = (\boldsymbol{I} - \boldsymbol{U}^{-1}\boldsymbol{L}^{-1}\boldsymbol{B})\boldsymbol{\xi^{(s)}} + \boldsymbol{U}^{-1}\boldsymbol{L}^{-1}\boldsymbol{c}.$$

Such incomplete LU factorizations lead to robust smoothers for fluid-mechanical problems, see, for example, Wesseling [105] or Wittum [106, 107]. Compared to the classical ILU factorization, the block-ILU factorization is more expensive, but also more robust, see Hackbusch [63].

The robustness can be further improved by a simple trick: Instead of computing an incomplete decomposition of the matrix $\boldsymbol{B}$, we compute a factorization of a modified matrix $\boldsymbol{B_\omega} \coloneqq \boldsymbol{B} + \omega\boldsymbol{D}$ where $\boldsymbol{D} = \operatorname{diag}(d_i)$ is a diagonal matrix with entries d_i which itself are diagonal blocks. The entries of these blocks are given by

$$(d_i)_l = \sum_{j \neq i} \left| (B_{ij})_{ll} \right|.$$

Numerical tests show $\omega \approx 0.1$ to be a good choice. While computing the block-ILU factorization, the diagonal blocks B_{ii} have to be inverted. This is done exactly by Gaussian elimination.

The mean value constraint for the pressure is re-established in each linear iteration step by subtracting the mean value of the actual pressure approximation.

For optimal complexity of the multigrid iteration, the number of unknowns must be reduced by a certain factor when going from one mesh to the next coarser one. To achieve this, the meshes used in the multigrid iteration are created by global coarsening, see Figure 3.9. For further details on the construction of efficient multigrid methods on locally refined meshes, we refer to Becker and Braack [9].

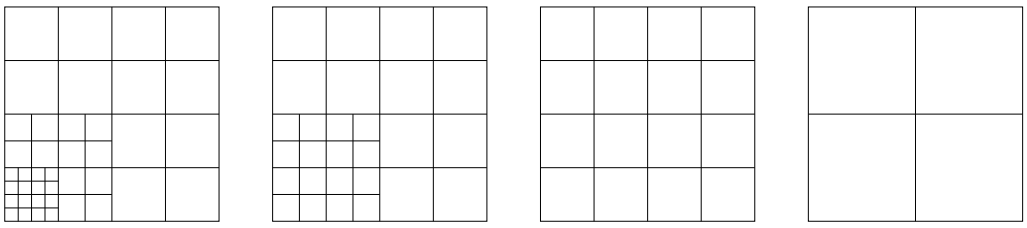


Figure 3.9. Global coarsening for meshes with patch structure

4 A Posteriori Error Estimation

Most often, the aim of simulations in computational fluid dynamics is the efficient computation of functional values of the solution, such as the drag- or lift-coefficient in flow problems around an obstacle, see Schäfer and Turek [95] for benchmark configurations. This chapter is dedicated to the development of an a posteriori error estimator which measures the discretization error exactly in the functional value one is interested in. The application of adaptive methods in the context of finite element discretizations of partial differential equations is well accepted and widely used, see Chapter 1 for references.

The error estimator developed in this chapter is an extension of these concepts to nonstationary flow problems and is mainly based on the results already published in Schmich and Vexler [96] for nonlinear parabolic partial differential equations. It separates the total discretization error into contributions due to the discretization in time and in space. The reliable quantitative error estimation is the key ingredient in setting up an adaptive algorithm during which the temporal and spatial discretization errors are balanced and simultaneously decreased. For the evaluation of the developed error estimator, a linear auxiliary problem has to be solved.

The outline of this chapter is as follows: In Section 4.1, we first recall an abstract error representation formula. Section 4.2 is devoted to the application of the abstract error representation formula to the problem under consideration and the development of the a posteriori error estimator with respect to the functional value of interest for both continuous and discontinuous Galerkin discretizations in time. The derivation is given in detail for the dG(r) case whereas for the cG(r) method we only present the results since they can be derived in a similar way. In Section 4.3, we present the numerical evaluation of the arising terms in the a posteriori error estimator. Section 4.4 then is dedicated to the localization of the obtained error estimator needed for adaptive refinement of the temporal and spatial discretizations. The following Section 4.5 presents an adaptive algorithm for the simultaneous refinement of both discretizations. For comparison, we introduce a simple smoothness-based error indicator in Section 4.6. The final Section 4.7 shows results and a comparison of both a posteriori error estimators derived in Sections 4.2 and 4.6.

4.1 Abstract error representation

In this section, we recall an abstract result concerning error representation from Becker and Rannacher [13]. Here, we present a slightly more general form which is able to cover the case of inconsistently stabilized discretizations and will be used in the sequel to obtain error representation formulas for the discretization error in the quantity of interest.

Lemma 4.1. *Let Y be a function space and L and $\widetilde{L}$ be three times Gâteaux differentiable functionals on Y. We seek a stationary point y_1 of L on a subspace $Y_1 \subseteq Y$: Find $y_1 \in Y_1$ such that*

$$L'(y_1)(\delta y_1) = 0 \quad \forall \delta y_1 \in Y_1. \tag{4.1}$$

This equation is approximated by a Galerkin method using the functional $\widetilde{L}$ on a subspace $Y_2 \subseteq Y$. Hence, the discrete problem seeks $y_2 \in Y_2$ such that

$$\widetilde{L}'(y_2)(\delta y_2) = 0 \quad \forall \delta y_2 \in Y_2. \tag{4.2}$$

If the continuous solution y_1 additionally fulfills

$$L'(y_1)(y_2) = 0 \tag{4.3}$$

with the approximative solution y_2, we have for arbitrary $\tilde{y}_2 \in Y_2$ the error representation

$$L(y_1) - \widetilde{L}(y_2) = \frac{1}{2}L'(y_2)(y_1 - \tilde{y}_2) + \frac{1}{2}(L - \widetilde{L})'(y_2)(\tilde{y}_2 - y_2) + (L - \widetilde{L})(y_2) + \mathcal{R}, \tag{4.4}$$

where the remainder term $\mathcal{R}$ is given by means of $e := y_1 - y_2$ as

$$\mathcal{R} = \frac{1}{2}\int_0^1 L'''(y_2 + se)(e, e, e) \cdot s \cdot (s - 1)\ \mathrm{d}s.$$

Proof. We write by the main theorem of calculus

$$L(y_1) - \widetilde{L}(y_2) = L(y_1) - L(y_2) + (L - \widetilde{L})(y_2) = \int_0^1 L'(y_2 + se)(e)\ \mathrm{d}s + (L - \widetilde{L})(y_2).$$

Using the trapezoidal rule

$$\int_0^1 f(s)\ \mathrm{d}s = \frac{1}{2}f(0) + \frac{1}{2}f(1) + \frac{1}{2}\int_0^1 f''(s) \cdot s \cdot (s - 1)\ \mathrm{d}s$$

for approximating the integral, supplies

$$L(y_1) - \widetilde{L}(y_2) = \frac{1}{2}L'(y_2)(e) + \frac{1}{2}L'(y_1)(e) + \mathcal{R} + (L - \widetilde{L})(y_2).$$

Because of (4.1) and (4.3), we have

$$L'(y_1)(e) = 0.$$

Due to assertion (4.2), we may replace $L'(y_2)(e)$ by

$$L'(y_2)(y_1 - \tilde{y}_2) + L'(y_2)(\tilde{y}_2 - y_2) = L'(y_2)(y_1 - \tilde{y}_2) + (L - \widetilde{L})'(y_2)(\tilde{y}_2 - y_2)$$

for arbitrary $\tilde{y}_2 \in Y_2$. This completes the proof. □

Remark 4.1. The presented version of Lemma 4.1 is more general than the version in Becker and Rannacher [13], where it is formulated using the stronger requirement $Y_1 = Y$ instead of (4.3) as well as $L = \widetilde{L}$. However, since we cannot always ensure $Y_2 \subseteq Y_1$ in our discretizations considered and we want to consider the inconsistent local projection stabilization, this simple modification is necessary.

4.2 Derivation of the a posteriori error estimator

In this section, we want to apply the abstract result of Section 4.1 in the derivation of an a posteriori error estimator which measures the temporal and spatial discretization errors with respect to a given functional value $J(\boldsymbol{u})$ of the solution and separates the contributions due to both discretizations. Hence, we actually want to construct two error estimators η_k and η_h such that

$$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}}) \approx \eta_k + \eta_h.$$

Throughout this thesis, we assume the functional J to be of the form

$$J(\boldsymbol{u}) = \int_I J_1(\boldsymbol{u}(t)) \, \mathrm{d}t + J_2(\boldsymbol{u}(T))$$

where of course J_1 or J_2 may be zero.

Remark 4.2. The derivation of the error estimator is done in a purely formal fashion. The discussion of existence or uniqueness of solutions to the arising continuous, semi-discrete, and fully discrete problems would require more assumptions, e. g., on the functional J.

To this end, we introduce the Lagrangians $\mathcal{L}\colon X \times X \to \mathbb{R}$, $\mathcal{L}_h : X_{kh}^{r,s} \times \widetilde{X}_{kh}^{r-1,s} \to \mathbb{R}$, $\widetilde{\mathcal{L}} : \widetilde{X}_k^r \times \widetilde{X}_k^r \to \mathbb{R}$, and $\widetilde{\mathcal{L}}_h : \widetilde{X}_{kh}^{r,s} \times \widetilde{X}_{kh}^{r,s} \to \mathbb{R}$ by

$$\begin{aligned}
\mathcal{L}(\boldsymbol{u}, \boldsymbol{z}) &\coloneqq J(\boldsymbol{u}) + \int_I (\boldsymbol{f} - \partial_t \boldsymbol{v}, \boldsymbol{w}) \, \mathrm{d}t - a(\boldsymbol{u})(\boldsymbol{z}) - (\boldsymbol{v}(0) - \boldsymbol{v^0}, \boldsymbol{w}(0)), \\
\mathcal{L}_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}) &\coloneqq \mathcal{L}(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}) - \mathcal{S}_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}), \\
\widetilde{\mathcal{L}}(\boldsymbol{u_k}, \boldsymbol{z_k}) &\coloneqq J(\boldsymbol{u_k}) + \sum_{m=1}^{M} \int_{I_m} (\boldsymbol{f} - \partial_t \boldsymbol{v_k}, \boldsymbol{w_k}) \, \mathrm{d}t - a(\boldsymbol{u_k})(\boldsymbol{z_k}) - \sum_{m=0}^{M-1} ([\boldsymbol{v_k}]_m, \boldsymbol{w}_{\boldsymbol{k},m}^+) \\
&\quad - (\boldsymbol{v}_{\boldsymbol{k},0}^- - \boldsymbol{v^0}, \boldsymbol{w}_{\boldsymbol{k},0}^-), \\
\widetilde{\mathcal{L}}_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}) &\coloneqq \widetilde{\mathcal{L}}(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}) - \mathcal{S}_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})
\end{aligned}$$

with

$$\mathcal{S}_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}) \coloneqq \sum_{m=1}^{M} \int_{I_m} s_h^m(\boldsymbol{u_{kh}})(\boldsymbol{z_{kh}}) \, \mathrm{d}t.$$

Remark 4.3. The Lagrange multipliers $\boldsymbol{z} = (\boldsymbol{w}, q)^T$, $\boldsymbol{z_k} = (\boldsymbol{w_k}, q_k)^T$, and $\boldsymbol{z_{kh}} = (\boldsymbol{w_{kh}}, q_{kh})^T$ introduced in this context, are usually called *dual variables* in contrast to the *primal variables* $\boldsymbol{u} = (\boldsymbol{v}, p)^T$, $\boldsymbol{u_k} = (\boldsymbol{v_k}, p_k)^T$, and $\boldsymbol{u_{kh}} = (\boldsymbol{v_{kh}}, p_{kh})^T$.

Using the Lagrangians, we can express the functional values of the continuous, semi-discrete, and fully discrete solution in the case of a dG(r) time discretization as follows:

$$J(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{u}, \boldsymbol{\varphi}) \qquad \forall \boldsymbol{\varphi} \in X, \tag{4.5a}$$

$$J(\boldsymbol{u_k}) = \widetilde{\mathcal{L}}(\boldsymbol{u_k}, \boldsymbol{\varphi}) \qquad \forall \boldsymbol{\varphi} \in \widetilde{X}_k^r, \tag{4.5b}$$

$$J(\boldsymbol{u_{kh}}) = \widetilde{\mathcal{L}}_h(\boldsymbol{u_{kh}}, \boldsymbol{\varphi}) \quad \forall \boldsymbol{\varphi} \in \widetilde{X}_{kh}^{r,s}. \tag{4.5c}$$

In the case of a cG(r) semi-discretization in time, we have to replace (4.5b) and (4.5c) by

$$J(\boldsymbol{u_k}) = \mathcal{L}(\boldsymbol{u_k}, \boldsymbol{\varphi}) \qquad \forall \boldsymbol{\varphi} \in \widetilde{X}_k^{r-1} \quad \text{and}$$
$$J(\boldsymbol{u_{kh}}) = \mathcal{L}_h(\boldsymbol{u_{kh}}, \boldsymbol{\varphi}) \quad \forall \boldsymbol{\varphi} \in \widetilde{X}_{kh}^{r-1,s}.$$

Since we want to separate the influences of the temporal and spatial discretization, we split the total discretization error as

$$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}}) = (J(\boldsymbol{u}) - J(\boldsymbol{u_k})) + (J(\boldsymbol{u_k}) - J(\boldsymbol{u_{kh}})),$$

where $\boldsymbol{u}$ denotes the continuous solution, $\boldsymbol{u_k}$ the semi-discrete solution of the dG(r) discretization in time, and $\boldsymbol{u_{kh}}$ the fully discrete solution of the cG(s)dG(r) discretization. Note that these solutions are given as the first component of stationary points of the corresponding Lagrangians, since

$$\mathcal{L}'_{\boldsymbol{z}}(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{\varphi}) = 0 \quad \forall \boldsymbol{\varphi} \in X,$$
$$\widetilde{\mathcal{L}}'_{\boldsymbol{z}}(\boldsymbol{u_k}, \boldsymbol{z_k})(\boldsymbol{\varphi}) = 0 \quad \forall \boldsymbol{\varphi} \in \widetilde{X}_k^r,$$
$$\widetilde{\mathcal{L}}'_{h,\boldsymbol{z}}(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\boldsymbol{\varphi}) = 0 \quad \forall \boldsymbol{\varphi} \in \widetilde{X}_{kh}^{r,s}$$

are just the equations for the continuous, semi-discrete and fully discrete problem.

We are now able to state the following

Theorem 4.2. *Let $(\boldsymbol{u}, \boldsymbol{z})^T$, $(\boldsymbol{u_k}, \boldsymbol{z_k})^T$, and $(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})^T$ denote stationary points of $\mathcal{L}$, $\widetilde{\mathcal{L}}$, and $\widetilde{\mathcal{L}}_h$ on different discretization levels, i. e.,*

$$\mathcal{L}'(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{\delta u}, \boldsymbol{\delta z}) = \widetilde{\mathcal{L}}'(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{\delta u}, \boldsymbol{\delta z}) = 0 \quad \forall (\boldsymbol{\delta u}, \boldsymbol{\delta z})^T \in X \times X,$$
$$\widetilde{\mathcal{L}}'(\boldsymbol{u_k}, \boldsymbol{z_k})(\boldsymbol{\delta u_k}, \boldsymbol{\delta z_k}) = 0 \quad \forall (\boldsymbol{\delta u_k}, \boldsymbol{\delta z_k})^T \in \widetilde{X}_k^r \times \widetilde{X}_k^r,$$
$$\widetilde{\mathcal{L}}'_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\boldsymbol{\delta u_{kh}}, \boldsymbol{\delta z_{kh}}) = 0 \quad \forall (\boldsymbol{\delta u_{kh}}, \boldsymbol{\delta z_{kh}})^T \in \widetilde{X}_{kh}^{r,s} \times \widetilde{X}_{kh}^{r,s}.$$

Then, there hold the following error representation formulas for the discretization errors in time and space:

$$J(\boldsymbol{u}) - J(\boldsymbol{u_k}) = \frac{1}{2}\widetilde{\mathcal{L}}'(\boldsymbol{u_k}, \boldsymbol{z_k})(\boldsymbol{u} - \tilde{\boldsymbol{u}}_k, \boldsymbol{z} - \tilde{\boldsymbol{z}}_k) + \mathcal{R}_k,$$
$$J(\boldsymbol{u_k}) - J(\boldsymbol{u_{kh}}) = \frac{1}{2}\widetilde{\mathcal{L}}'(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\boldsymbol{u_k} - \tilde{\boldsymbol{u}}_{kh}, \boldsymbol{z_k} - \tilde{\boldsymbol{z}}_{kh}) + \frac{1}{2}\mathcal{S}'_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\tilde{\boldsymbol{u}}_{kh} - \boldsymbol{u_{kh}}, \tilde{\boldsymbol{z}}_{kh} - \boldsymbol{z_{kh}}) + \mathcal{S}_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}) + \mathcal{R}_h.$$

Here, $(\tilde{\boldsymbol{u}}_k, \tilde{\boldsymbol{z}}_k)^T \in \widetilde{X}_k^r \times \widetilde{X}_k^r$ and $(\tilde{\boldsymbol{u}}_{kh}, \tilde{\boldsymbol{z}}_{kh})^T \in \widetilde{X}_{kh}^{r,s} \times \widetilde{X}_{kh}^{r,s}$ can be chosen arbitrarily and the remainder terms $\mathcal{R}_k$ and $\mathcal{R}_h$ have the same structure as in Lemma 4.1.

Proof. Due to (4.5), we may especially write

$$J(\boldsymbol{u}) - J(\boldsymbol{u_k}) = \mathcal{L}(\boldsymbol{u}, \boldsymbol{z}) - \widetilde{\mathcal{L}}(\boldsymbol{u_k}, \boldsymbol{z_k}) = \widetilde{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{z}) - \widetilde{\mathcal{L}}(\boldsymbol{u_k}, \boldsymbol{z_k}), \tag{4.6a}$$

$$J(\boldsymbol{u_k}) - J(\boldsymbol{u_{kh}}) = \widetilde{\mathcal{L}}(\boldsymbol{u_k}, \boldsymbol{z_k}) - \widetilde{\mathcal{L}}_h(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}}). \tag{4.6b}$$

Here, we have used the fact that

$$J(\boldsymbol{u}) = \mathcal{L}(\boldsymbol{u}, \boldsymbol{z}) = \widetilde{\mathcal{L}}(\boldsymbol{u}, \boldsymbol{z}),$$

since the first component $\boldsymbol{v}$ of $\boldsymbol{u} \in X$ is continuous and hence the additional jump terms in $\widetilde{\mathcal{L}}$ compared to $\mathcal{L}$ vanish. Next, we apply Lemma 4.1 with

$$\begin{aligned} &L = \widetilde{\mathcal{L}}, \quad \widetilde{L} = \widetilde{\mathcal{L}}, \quad Y_1 = X \times X, \qquad Y_2 = \widetilde{X}_k^r \times \widetilde{X}_k^r \qquad \text{for (4.6a)}, \\ &L = \widetilde{\mathcal{L}}, \quad \widetilde{L} = \widetilde{\mathcal{L}}_h, \quad Y_1 = \widetilde{X}_k^r \times \widetilde{X}_k^r, \quad Y_2 = \widetilde{X}_{kh}^{r,s} \times \widetilde{X}_{kh}^{r,s} \quad \text{for (4.6b)}. \end{aligned}$$

In the second case, we have $Y_2 \subseteq Y_1$ since $\widetilde{X}_{kh}^{r,s} \subseteq \widetilde{X}_k^r$. Hence, we can take $Y \coloneqq Y_1$ and condition (4.3) is fulfilled automatically.

For the first case, he have to choose $Y \coloneqq Y_1 + Y_2$ since $\widetilde{X}_k^r \not\subseteq X$. Thus, we must check condition (4.3) which reads

$$\widetilde{\mathcal{L}}'(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{u_k}, \boldsymbol{z_k}) = 0$$

or equivalently

$$\widetilde{\mathcal{L}}'_{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{u_k}) = 0 \quad \text{and} \quad \widetilde{\mathcal{L}}'_{\boldsymbol{z}}(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{z_k}) = 0.$$

We only show the proof of the second condition

$$\widetilde{\mathcal{L}}'_{\boldsymbol{z}}(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{z_k}) = 0. \tag{4.7}$$

The first one can be handled analogously. Due to the continuity of the first component of the continuous solution $\boldsymbol{u}$ with respect to time, the jump terms and the initial condition in $\widetilde{\mathcal{L}}$ vanish on $\boldsymbol{u} \in X$. Hence, equation (4.7) may be rewritten as

$$\sum_{m=1}^{M} \int_{I_m} (\boldsymbol{f} - \partial_t \boldsymbol{v}, \boldsymbol{w_k}) \, \mathrm{d}t - a(\boldsymbol{u})(\boldsymbol{z_k}) = 0.$$

By construction, the continuous solution $\boldsymbol{u}$ fulfills

$$\int_I (\partial_t \boldsymbol{v}, \boldsymbol{\psi}) + a(\boldsymbol{u})(\boldsymbol{\varphi}) = \int_I (\boldsymbol{f}, \boldsymbol{\psi}) \, \mathrm{d}t \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in X. \tag{4.8}$$

Since X is dense in $L^2(I, H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R})$ with respect to the norm of $L^2(I, H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R})$ and since there are no time derivatives on $\boldsymbol{\psi}$ in (4.8), this equation also holds true for all $\boldsymbol{\varphi} \in L^2(I, H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R})$. The inclusion $\boldsymbol{z_k} \in \widetilde{X}_k^r \subseteq L^2(I, H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R})$ then implies that condition (4.7) is fulfilled.

Finally, the assertion of the theorem is a direct consequence of Lemma 4.1 applied to the separated errors (4.6). □

Introducing the primal and dual residual

$$\begin{aligned}\rho(\boldsymbol{u})(\boldsymbol{\varphi}) &\coloneqq \widetilde{\mathcal{L}}'_{\boldsymbol{z}}(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}),\\ \rho^*(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}) &\coloneqq \widetilde{\mathcal{L}}'_{\boldsymbol{u}}(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}),\end{aligned}$$

the result of Theorem 4.2 may be rewritten as

$$J(\boldsymbol{u}) - J(\boldsymbol{u_k}) \approx \frac{1}{2}\Big\{\rho(\boldsymbol{u_k})(\boldsymbol{z}-\tilde{\boldsymbol{z}}_{\boldsymbol{k}}) + \rho^*(\boldsymbol{u_k},\boldsymbol{z_k})(\boldsymbol{u}-\tilde{\boldsymbol{u}}_{\boldsymbol{k}})\Big\}, \tag{4.9a}$$

$$J(\boldsymbol{u_k}) - J(\boldsymbol{u_{kh}}) \approx \frac{1}{2}\Big\{\rho(\boldsymbol{u_{kh}})(\boldsymbol{z_k}-\tilde{\boldsymbol{z}}_{\boldsymbol{kh}}) + \rho^*(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{u_k}-\tilde{\boldsymbol{u}}_{\boldsymbol{kh}})\Big\}, \tag{4.9b}$$

where we have neglected the remainder terms $\mathcal{R}_k$ and $\mathcal{R}_h$ as well as the additional terms due to stabilization which can be assumed to be small because they contain small stabilization parameters. At least, numerical results show that they are indeed negligible, see Section 4.7.

In the case of a cG(r) discretization in time, we obtain a similar result which is summarized in the following corollary:

Corollary 4.3. *Let $(\boldsymbol{u},\boldsymbol{z})^T$, $(\boldsymbol{u_k},\boldsymbol{z_k})^T$, and $(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})^T$ denote stationary points of $\mathcal{L}$ and $\mathcal{L}_h$ on different discretization levels, i. e.,*

$$\begin{aligned}\mathcal{L}'(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\delta u},\boldsymbol{\delta z}) &= 0 \quad \forall(\boldsymbol{\delta u},\boldsymbol{\delta z})^T \in X\times X,\\ \mathcal{L}'(\boldsymbol{u_k},\boldsymbol{z_k})(\boldsymbol{\delta u_k},\boldsymbol{\delta z_k}) &= 0 \quad \forall(\boldsymbol{\delta u_k},\boldsymbol{\delta z_k})^T \in X_k^r\times\widetilde{X}_k^{r-1},\\ \mathcal{L}'_h(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{\delta u_{kh}},\boldsymbol{\delta z_{kh}}) &= 0 \quad \forall(\boldsymbol{\delta u_{kh}},\boldsymbol{\delta z_{kh}})^T \in X_{kh}^{r,s}\times\widetilde{X}_{kh}^{r-1,s}.\end{aligned}$$

Then, there hold the following error representation formulas for the discretization errors in time and space:

$$\begin{aligned}J(\boldsymbol{u}) - J(\boldsymbol{u_k}) &= \frac{1}{2}\mathcal{L}'(\boldsymbol{u_k},\boldsymbol{z_k})(\boldsymbol{u}-\tilde{\boldsymbol{u}}_{\boldsymbol{k}},\boldsymbol{z}-\tilde{\boldsymbol{z}}_{\boldsymbol{k}}) + \mathcal{R}_k,\\ J(\boldsymbol{u_k}) - J(\boldsymbol{u_{kh}}) &= \frac{1}{2}\mathcal{L}'(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{u_k}-\tilde{\boldsymbol{u}}_{\boldsymbol{kh}},\boldsymbol{z_k}-\tilde{\boldsymbol{z}}_{\boldsymbol{kh}})\\ &\quad + \frac{1}{2}\mathcal{S}'_h(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\tilde{\boldsymbol{u}}_{\boldsymbol{kh}}-\boldsymbol{u_{kh}},\tilde{\boldsymbol{z}}_{\boldsymbol{kh}}-\boldsymbol{z_{kh}}) + \mathcal{S}_h(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}}) + \mathcal{R}_h.\end{aligned}$$

Here, $(\tilde{\boldsymbol{u}}_{\boldsymbol{k}},\tilde{\boldsymbol{z}}_{\boldsymbol{k}})^T \in X_k^r\times\widetilde{X}_k^{r-1}$ and $(\tilde{\boldsymbol{u}}_{\boldsymbol{kh}},\tilde{\boldsymbol{z}}_{\boldsymbol{kh}})^T \in X_{kh}^{r,s}\times\widetilde{X}_{kh}^{r-1,s}$ can be chosen arbitrarily and the remainder terms $\mathcal{R}_k$ and $\mathcal{R}_h$ have the same structure as in Lemma 4.1.

Of course, we also obtain a similar representation as in (4.9) for the case of a cG(r) discretization in time, this time involving the primal and dual residuals defined via the continuous Lagrangian:

$$\begin{aligned}\rho(\boldsymbol{u})(\boldsymbol{\varphi}) &\coloneqq \mathcal{L}'_{\boldsymbol{z}}(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}),\\ \rho^*(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}) &\coloneqq \mathcal{L}'_{\boldsymbol{u}}(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}).\end{aligned}$$

4.3 Evaluation of the error estimators

In this section, we give details on the numerical evaluation of the a posteriori error estimators developed in the previous section. The error estimators involve the continuous, semi-discrete, and fully discrete dual solutions $\boldsymbol{z} \in X$, $\boldsymbol{z_k} \in \widetilde{X}_k^r$ or $\widetilde{X}_k^{r-1}$ (depending on whether a dG(r) or cG(r) discretization in time is used), and $\boldsymbol{z_{kh}} \in \widetilde{X}_{kh}^{r,s}$ or $\widetilde{X}_{kh}^{r-1,s}$. In the case of a dG($r$) semi-discretization in time, they are given as solutions of

$$\begin{aligned}
\mathcal{L}'_{\boldsymbol{u}}(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}) &= 0 \quad \forall \boldsymbol{\varphi} \in X, \\
\widetilde{\mathcal{L}}'_{\boldsymbol{u}}(\boldsymbol{u_k},\boldsymbol{z_k})(\boldsymbol{\varphi}) &= 0 \quad \forall \boldsymbol{\varphi} \in \widetilde{X}_k^r, \\
\widetilde{\mathcal{L}}'_{h,\boldsymbol{u}}(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{\varphi}) &= 0 \quad \forall \boldsymbol{\varphi} \in \widetilde{X}_{kh}^{r,s},
\end{aligned}$$

while for cG(r) case the characterizing equations are

$$\begin{aligned}
\mathcal{L}'_{\boldsymbol{u}}(\boldsymbol{u},\boldsymbol{z})(\boldsymbol{\varphi}) &= 0 \quad \forall \boldsymbol{\varphi} \in X, \\
\mathcal{L}'_{\boldsymbol{u}}(\boldsymbol{u_k},\boldsymbol{z_k})(\boldsymbol{\varphi}) &= 0 \quad \forall \boldsymbol{\varphi} \in X_k^r, \\
\mathcal{L}'_{h,\boldsymbol{u}}(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{\varphi}) &= 0 \quad \forall \boldsymbol{\varphi} \in X_{kh}^{r,s}.
\end{aligned}$$

We want to show the precise form of these derivatives, i. e., the equations the dual solutions have to fulfill. The continuous dual solution $\boldsymbol{z} = (\boldsymbol{w}, q)^T \in X$ is the solution of

$$\begin{aligned}
&\int_I (\boldsymbol{\psi}, -\partial_t \boldsymbol{w}) \,\mathrm{d}t + a'(\boldsymbol{u})(\boldsymbol{\varphi},\boldsymbol{z}) + (\boldsymbol{\psi}(T), \boldsymbol{w}(T)) \\
&\qquad = \int_I J_1'(\boldsymbol{u})(\boldsymbol{\varphi}) \,\mathrm{d}t + J_2'(\boldsymbol{u}(T))(\boldsymbol{\varphi}(T)) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in X, \quad (4.10)
\end{aligned}$$

where we have integrated by parts which is admissible for functions in X, see, for instance, Wloka [108]. Let us next consider the case of a dG(r) time discretization. Then the semi-discrete dual solution $\boldsymbol{z_k} = (\boldsymbol{w_k}, q_k)^T \in \widetilde{X}_k^r$ and the fully discrete dual solution $\boldsymbol{z_{kh}} = (\boldsymbol{w_{kh}}, q_{kh})^T \in \widetilde{X}_{kh}^{r,s}$ fulfill

$$\begin{aligned}
&\sum_{m=1}^{M} \int_{I_m} (\boldsymbol{\psi}, -\partial_t \boldsymbol{w_k}) \,\mathrm{d}t + a'(\boldsymbol{u_k})(\boldsymbol{\varphi},\boldsymbol{z_k}) - \sum_{m=0}^{M-1} (\boldsymbol{\psi}_m^-, [\boldsymbol{w_k}]_m) + (\boldsymbol{\psi}_M^-, \boldsymbol{w}_{\boldsymbol{k},M}^-) \\
&\qquad = \int_I J_1'(\boldsymbol{u_k})(\boldsymbol{\varphi}) \,\mathrm{d}t + J_2'(\boldsymbol{u}_{\boldsymbol{k},M}^-)(\boldsymbol{\varphi}_M^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_k^r \quad (4.11)
\end{aligned}$$

and

$$\begin{aligned}
&\sum_{m=1}^{M} \int_{I_m} (\boldsymbol{\psi}, -\partial_t \boldsymbol{w_{kh}}) \,\mathrm{d}t + a_h'(\boldsymbol{u_{kh}})(\boldsymbol{\varphi},\boldsymbol{z_{kh}}) - \sum_{m=0}^{M-1} (\boldsymbol{\psi}_m^-, [\boldsymbol{w_{kh}}]_m) + (\boldsymbol{\psi}_M^-, \boldsymbol{w}_{\boldsymbol{kh},M}^-) \\
&\qquad = \int_I J_1'(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) \,\mathrm{d}t + J_2'(\boldsymbol{u}_{\boldsymbol{kh},M}^-)(\boldsymbol{\varphi}_M^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in \widetilde{X}_{kh}^{r,s}, \quad (4.12)
\end{aligned}$$

respectively. In the context of a cG(r) semi-discretization in time, the semi-discrete dual solution $\boldsymbol{z_k} = (\boldsymbol{w_k}, q_k)^T \in \widetilde{X}_k^{r-1}$ and the fully discrete dual solution $\boldsymbol{z_{kh}} = (\boldsymbol{w_{kh}}, q_{kh})^T \in \widetilde{X}_{kh}^{r-1,s}$ instead satisfy

$$\sum_{m=1}^{M} \int_{I_m} (\boldsymbol{\psi}, -\partial_t \boldsymbol{w_k}) \, \mathrm{d}t + a'(\boldsymbol{u_k})(\boldsymbol{\varphi}, \boldsymbol{z_k}) - \sum_{m=0}^{M-1} (\boldsymbol{\psi}(t_m), [\boldsymbol{w_k}]_m) + (\boldsymbol{\psi}(t_M), \boldsymbol{w}_{\boldsymbol{k},M}^-)$$
$$= \int_I J_1'(\boldsymbol{u_k})(\boldsymbol{\varphi}) \, \mathrm{d}t + J_2'(\boldsymbol{u}_{\boldsymbol{k},M}^-)(\boldsymbol{\varphi}_M^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in X_k^r \quad (4.13)$$

and

$$\sum_{m=1}^{M} \int_{I_m} (\boldsymbol{\psi}, -\partial_t \boldsymbol{w_{kh}}) \, \mathrm{d}t + a_h'(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}, \boldsymbol{z_{kh}}) - \sum_{m=0}^{M-1} (\boldsymbol{\psi}(t_m), [\boldsymbol{w_{kh}}]_m) + (\boldsymbol{\psi}(t_M), \boldsymbol{w}_{\boldsymbol{kh},M}^-)$$
$$= \int_I J_1'(\boldsymbol{u_{kh}})(\boldsymbol{\varphi}) \, \mathrm{d}t + J_2'(\boldsymbol{u}_{\boldsymbol{kh},M}^-)(\boldsymbol{\varphi}_M^-) \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in X_{kh}^{r,s}. \quad (4.14)$$

The precise time-stepping formulation of these dual problems in the considered cases cG(s)dG(0), cG(s)dG(1), and cG(s)cG(1) with $s \in \{ 1, 2 \}$ can be obtained in a similar way as for the primal problem, see Section 3.4.

Note that for solving the dual problems (4.12) or (4.14), the primal solution $\boldsymbol{u_{kh}}$ is needed on the whole time interval $\bar{I}$ due to the nonlinear structure of the primal problem. A common way to deal with this difficulty is to apply checkpointing techniques which reduce the required amount of memory because the primal solution is only stored on so-called *checkpoints*. The drawback is that we have to solve the (nonlinear) primal problem more often to recover the primal solution between two checkpoints. More information on checkpointing can be found, for instance, in Griewank [59], Berggren, Glowinski, and Lions [16] or Walther and Griewank [104]. However, since in the last years the capacities of main memory and hard disk drives have been growing rapidly, we propose to store the primal solution over the whole time interval. For two-dimensional simulations, this can often be done by only using the main memory, while in three spatial dimensions we suggest storing the data on hard disk. Even though the access of reading and writing from and to hard disk is much slower than the access to main memory, this can be assumed to be still much faster than solving several time steps of the nonlinear primal problem more than once. For a discussion of this topic, we also refer to Meidner [78].

Let us now consider the numerical evaluation of the error estimator developed in the previous section. Since the quantities $\tilde{\boldsymbol{u}}_{\boldsymbol{k}}$, $\tilde{\boldsymbol{z}}_{\boldsymbol{k}}$, $\tilde{\boldsymbol{u}}_{\boldsymbol{kh}}$, and $\tilde{\boldsymbol{z}}_{\boldsymbol{kh}}$ can be chosen arbitrarily in the corresponding spaces, the so-called *weights*, that is $\boldsymbol{u} - \tilde{\boldsymbol{u}}_{\boldsymbol{k}}$, $\boldsymbol{z} - \tilde{\boldsymbol{z}}_{\boldsymbol{k}}$, and so on, are mainly interpolation errors. We approximate these interpolation errors by higher order reconstructions of the discrete solutions. This approach relies on the "super-closeness" of the derivatives of these higher order interpolations to those of the continuous solution, see Becker and Rannacher [13] for more details on this topic and alternative approaches.

We introduce the following linear operators for approximating the weights in the error estimator:

$$\begin{aligned} \boldsymbol{v} - \tilde{\boldsymbol{v}}_{\boldsymbol{k}} &\approx \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{v})} \boldsymbol{v}_{\boldsymbol{k}}, & \boldsymbol{v}_{\boldsymbol{k}} - \tilde{\boldsymbol{v}}_{\boldsymbol{kh}} &\approx \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{v})} \boldsymbol{v}_{\boldsymbol{kh}}, \\ p - \tilde{p}_k &\approx \Pi_k^{(p)} p_k, & p_k - \tilde{p}_{kh} &\approx \Pi_h^{(p)} p_{kh}, \\ \boldsymbol{w} - \tilde{\boldsymbol{w}}_{\boldsymbol{k}} &\approx \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{w})} \boldsymbol{w}_{\boldsymbol{k}}, & \boldsymbol{w}_{\boldsymbol{k}} - \tilde{\boldsymbol{w}}_{\boldsymbol{kh}} &\approx \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{w})} \boldsymbol{w}_{\boldsymbol{kh}}, \\ q - \tilde{q}_k &\approx \Pi_k^{(q)} q_k, & q_k - \tilde{q}_{kh} &\approx \Pi_h^{(q)} q_{kh}. \end{aligned}$$

The operators $\boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{v})}$, $\Pi_k^{(p)}$, $\boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{w})}$, $\Pi_k^{(q)}$ as well as $\boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{v})}$, $\Pi_h^{(p)}$, $\boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{w})}$, and $\Pi_h^{(q)}$ are chosen as

cG(s)dG(0):

$$\begin{aligned} \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{v})} &\coloneqq \boldsymbol{I}_{\boldsymbol{k}}^{(\boldsymbol{1})} - \mathbf{id}, & \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{v})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2h}}^{(\boldsymbol{2s})} - \mathbf{id}, \\ \Pi_k^{(p)} &\coloneqq I_k^{(1)} - \mathrm{id}, & \Pi_h^{(p)} &\coloneqq I_{2h}^{(2s)} - \mathrm{id}, \\ \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{w})} &\coloneqq \boldsymbol{I}_{\boldsymbol{k}}^{(\boldsymbol{1})} - \mathbf{id}, & \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{w})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2h}}^{(\boldsymbol{2s})} - \mathbf{id}, \\ \Pi_k^{(q)} &\coloneqq I_k^{(1)} - \mathrm{id}, & \Pi_h^{(q)} &\coloneqq I_{2h}^{(2s)} - \mathrm{id}, \end{aligned}$$

where $I_k^{(1)}$ is given as in Figure 4.1(a) and $\boldsymbol{I}_{\boldsymbol{k}}^{(\boldsymbol{1})}$ acts component-wise as $I_k^{(1)}$.

cG(s)dG(1):

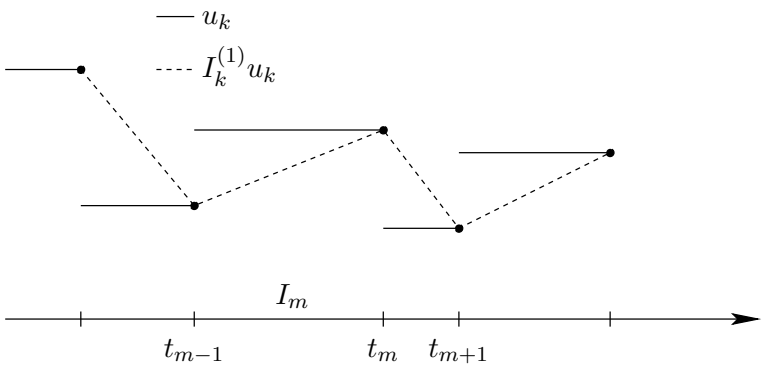
$$\begin{aligned} \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{v})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2k}}^{(\boldsymbol{2})} - \mathbf{id}, & \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{v})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2h}}^{(\boldsymbol{2s})} - \mathbf{id}, \\ \Pi_k^{(p)} &\coloneqq I_{2k}^{(2)} - \mathrm{id}, & \Pi_h^{(p)} &\coloneqq I_{2h}^{(2s)} - \mathrm{id}, \\ \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{w})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2k}}^{(\boldsymbol{2})} - \mathbf{id}, & \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{w})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2h}}^{(\boldsymbol{2s})} - \mathbf{id}, \\ \Pi_k^{(q)} &\coloneqq I_{2k}^{(2)} - \mathrm{id}, & \Pi_h^{(q)} &\coloneqq I_{2h}^{(2s)} - \mathrm{id}, \end{aligned}$$

where $I_{2k}^{(2)}$ is given as in Figure 4.1(b) and $\boldsymbol{I}_{\boldsymbol{2k}}^{(\boldsymbol{2})}$ acts component-wise as $I_{2k}^{(2)}$.

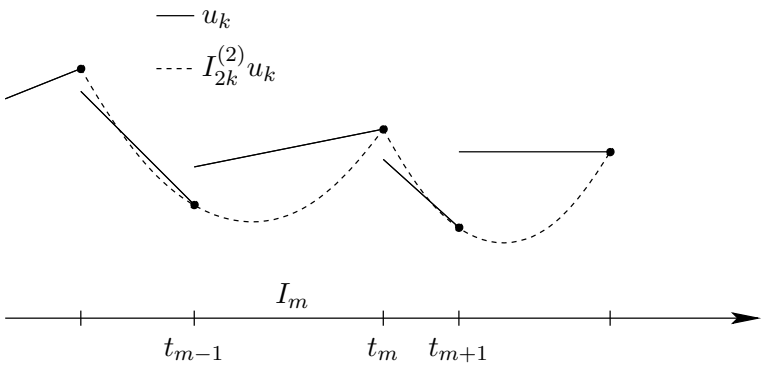
cG(s)cG(1):

$$\begin{aligned} \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{v})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2k}}^{(\boldsymbol{2})} - \mathbf{id}, & \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{v})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2h}}^{(\boldsymbol{2s})} - \mathbf{id}, \\ \Pi_k^{(p)} &\coloneqq I_k^{(1)} - \mathrm{id}, & \Pi_h^{(p)} &\coloneqq I_{2h}^{(2s)} - \mathrm{id}, \\ \boldsymbol{\Pi}_{\boldsymbol{k}}^{(\boldsymbol{w})} &\coloneqq \boldsymbol{I}_{\boldsymbol{k}}^{(\boldsymbol{1})} - \mathbf{id}, & \boldsymbol{\Pi}_{\boldsymbol{h}}^{(\boldsymbol{w})} &\coloneqq \boldsymbol{I}_{\boldsymbol{2h}}^{(\boldsymbol{2s})} - \mathbf{id}, \\ \Pi_k^{(q)} &\coloneqq I_k^{(1)} - \mathrm{id}, & \Pi_h^{(q)} &\coloneqq I_{2h}^{(2s)} - \mathrm{id}, \end{aligned}$$

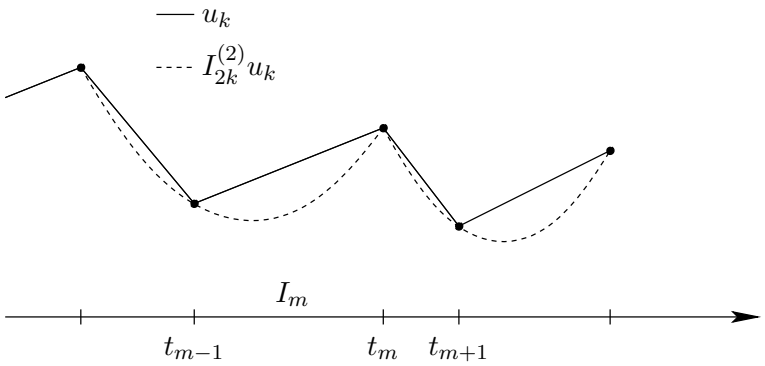
where $I_{2k}^{(2)}$ is given as in Figure 4.1(c) and $\boldsymbol{I}_{\boldsymbol{2k}}^{(\boldsymbol{2})}$ acts component-wise as $I_{2k}^{(2)}$.



(a) Continuous piecewise linear interpolation of a discontinuous piecewise constant function



(b) Continuous piecewise quadratic interpolation of a discontinuous piecewise linear function



(c) Continuous piecewise quadratic interpolation of a continuous piecewise linear function

Figure 4.1. Interpolation operators $I_k^{(1)}$ and $I_{2k}^{(2)}$

The spatial interpolation operators $I_{2h}^{(2)}\colon V_h^1 \to V_{2h}^2$ (for $s=1$) into the space of bi- or tri-quadratic trial functions and $I_{2h}^{(4)}\colon V_h^2 \to V_{2h}^4$ (for $s=2$) into the space of bi- or tri-quartic trial functions can easily be computed if the underlying mesh possesses a patch structure, see Figure 3.3. These spatial interpolation operators are extended into time point-wise by

$$(I_{2h}^{(2s)} u_{kh})(t) := I_{2h}^{(2s)}(u_{kh}(t)).$$

The last step in making the derived a posteriori error estimators computable is to replace all unknown solutions appearing either in the weights or in the residuals by the fully discrete versions, i. e., we replace

$$\rho(\boldsymbol{u_k})(\boldsymbol{z}-\tilde{\boldsymbol{z}}_{\boldsymbol{k}}) \text{ by } \rho(\boldsymbol{u_{kh}})(\boldsymbol{\Pi_k z_{kh}}) \quad \text{and} \quad \rho^*(\boldsymbol{u_k},\boldsymbol{z_k})(\boldsymbol{u}-\tilde{\boldsymbol{u}}_{\boldsymbol{k}}) \text{ by } \rho^*(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{\Pi_k u_{kh}})$$

with $\boldsymbol{\Pi_k z_{kh}} := (\boldsymbol{\Pi_k^{(w)}} \boldsymbol{w_{kh}}, \Pi_k^{(q)} q_{kh})^T$ and $\boldsymbol{\Pi_k u_{kh}} := (\boldsymbol{\Pi_k^{(v)}} \boldsymbol{v_{kh}}, \Pi_k^{(p)} p_{kh})^T$. The replacement in the weights is well accepted while the replacement in the residuals, i. e., the replacement of the linearization point, might seem critical. One could think of replacing the unknown solutions also by higher order interpolations as we did in the weights. However, in our numerical examples we see that this additional effort is not necessary to obtain quantitatively good results.

Remark 4.4. This observation is also substantiated by the fact that the replacement of the linearization point introduces an additional error which usually is of higher order. This can be seen as follows: The introduced error can be expressed as

$$\widetilde{\mathcal{L}}'(\boldsymbol{\zeta_k})(\boldsymbol{\zeta}-\tilde{\boldsymbol{\zeta}}_{\boldsymbol{k}}) - \widetilde{\mathcal{L}}'(\boldsymbol{\zeta_{kh}})(\boldsymbol{\zeta}-\tilde{\boldsymbol{\zeta}}_{\boldsymbol{k}}) = \int_0^1 \widetilde{\mathcal{L}}''(\boldsymbol{\zeta_{kh}} + s(\boldsymbol{\zeta_k}-\boldsymbol{\zeta_{kh}}))(\boldsymbol{\zeta_k}-\boldsymbol{\zeta_{kh}}, \boldsymbol{\zeta}-\tilde{\boldsymbol{\zeta}}_{\boldsymbol{k}})\, \mathrm{d}s$$

with $\boldsymbol{\zeta} = (\boldsymbol{u},\boldsymbol{z})^T$, $\boldsymbol{\zeta_k} = (\boldsymbol{u_k},\boldsymbol{z_k})^T$, and $\boldsymbol{\zeta_{kh}} = (\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})^T$. By choosing an appropriate interpolant for $\tilde{\boldsymbol{\zeta}}_{\boldsymbol{k}}$, this identity shows that the discussed replacement introduces an error of the order $O(h^2 k)$ whereas the total discretization error usually is not better than $O(h^2+k)$ in the case of a cG(1)dG(0) discretization. For more details, we refer to Meidner [78] or Schmich and Vexler [96].

Proceeding as proposed, we obtain the following a posteriori error estimator

$$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}}) \approx \eta_k + \eta_h$$

with

$$\eta_k := \frac{1}{2}\Big\{\rho(\boldsymbol{u_{kh}})(\boldsymbol{\Pi_k z_{kh}}) + \rho^*(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{\Pi_k u_{kh}})\Big\},$$
$$\eta_h := \frac{1}{2}\Big\{\rho(\boldsymbol{u_{kh}})(\boldsymbol{\Pi_h z_{kh}}) + \rho^*(\boldsymbol{u_{kh}},\boldsymbol{z_{kh}})(\boldsymbol{\Pi_h u_{kh}})\Big\}.$$

To get an impression of what terms have to be evaluated, we present exemplarily for the backward Euler variant of the cG(s)dG(0) discretization their precise form. To this end,

temporal integrals involving $\boldsymbol{u_{kh}}$ and $\boldsymbol{z_{kh}}$ are approximated by the box rule whereas those involving $\boldsymbol{I_k^{(1)}u_{kh}}$ and $\boldsymbol{I_k^{(1)}u_{kh}}$ are evaluated using the trapezoidal rule. This leads to the following representation:

$$\begin{aligned}
\rho(\boldsymbol{u_{kh}})(\boldsymbol{\Pi_k z_{kh}}) &= \sum_{m=1}^{M} \Big\{ (\boldsymbol{V_m} - \boldsymbol{V_{m-1}}, \boldsymbol{W_m} - \boldsymbol{W_{m-1}}) + \frac{k_m}{2}\bar{a}(\boldsymbol{U_m})(\boldsymbol{Z_m} - \boldsymbol{Z_{m-1}}) \\
&\quad + \frac{k_m}{2}(\boldsymbol{f}(t_{m-1}), \boldsymbol{W_{m-1}}) - \frac{k_m}{2}(\boldsymbol{f}(t_m), \boldsymbol{W_m}) \Big\}, \\
\rho^*(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\boldsymbol{\Pi_k u_{kh}}) &= \sum_{m=1}^{M} \Big\{ \frac{k_m}{2}\bar{a}'(\boldsymbol{U_m})(\boldsymbol{U_m} - \boldsymbol{U_{m-1}}, \boldsymbol{Z_m}) \\
&\quad - \frac{k_m}{2} J_1'(\boldsymbol{U_m})(\boldsymbol{U_m} - \boldsymbol{U_{m-1}}) \Big\}, \\
\rho(\boldsymbol{u_{kh}})(\boldsymbol{\Pi_h z_{kh}}) &= \sum_{m=1}^{M} \Big\{ k_m(\boldsymbol{f}(t_m), \boldsymbol{I_{2h}^{(2s)}W_m} - \boldsymbol{W_m}) - k_m\bar{a}(\boldsymbol{U_m})(\boldsymbol{I_{2h}^{(2s)}Z_m} - \boldsymbol{Z_m}) \\
&\quad - (\boldsymbol{V_m} - \boldsymbol{V_{m-1}}, \boldsymbol{I_{2h}^{(2s)}W_m} - \boldsymbol{W_m}) \Big\} - (\boldsymbol{V_0} - \boldsymbol{v^0}, \boldsymbol{I_{2h}^{(2s)}W_0} - \boldsymbol{W_0}), \\
\rho^*(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\boldsymbol{\Pi_h u_{kh}}) &= \sum_{m=1}^{M} \Big\{ k_m J_1'(\boldsymbol{U_m})(\boldsymbol{I_{2h}^{(2s)}U_m} - \boldsymbol{U_m}) \\
&\quad - k_m\bar{a}'(\boldsymbol{U_m})(\boldsymbol{I_{2h}^{(2s)}U_m} - \boldsymbol{U_m}, \boldsymbol{Z_m}) \\
&\quad + (\boldsymbol{I_{2h}^{(2s)}V_{m-1}} - \boldsymbol{V_{m-1}}, \boldsymbol{W_m} - \boldsymbol{W_{m-1}}) \Big\} \\
&\quad + J_2'(\boldsymbol{U_M})(\boldsymbol{I_{2h}^{(2s)}U_M} - \boldsymbol{U_M}) - (\boldsymbol{I_{2h}^{(2s)}V_M} - \boldsymbol{V_M}, \boldsymbol{W_M}).
\end{aligned}$$

Similar expressions are obtained for the cG(s)dG(1) and cG(s)cG(1) discretizations. Of course, quadrature rules of higher order have to be applied in order to exactly evaluate the temporal integrals.

4.4 Localization of the error estimators

The a posteriori error estimators developed in Section 4.2 serve for two purposes: Firstly, the quantitative assessment of the discretization error and secondly the adaptive refinement of the underlying discretizations in order to efficiently improve the accuracy. For the second aim, the information of the error estimators has to be localized to cell-wise or node-wise contributions. These quantities are then called *local error indicators*.

To this end, we split the overall error estimators into their contributions on each subinterval I_m by

$$\eta_k = \sum_{m=1}^{M} \eta_k^m \quad \text{and} \quad \eta_h = \sum_{m=0}^{M} \eta_h^m,$$

where the interval-wise error estimators are defined analogously to the global error estimators by

$$\eta_k^m := \frac{1}{2}\Big\{\rho_m(\boldsymbol{u_k})(\boldsymbol{z} - \tilde{\boldsymbol{z}}_{\boldsymbol{k}}) + \rho_m^*(\boldsymbol{u_k}, \boldsymbol{z_k})(\boldsymbol{u} - \tilde{\boldsymbol{u}}_{\boldsymbol{k}})\Big\},$$
$$\eta_h^m := \frac{1}{2}\Big\{\rho_m(\boldsymbol{u_{kh}})(\boldsymbol{z_k} - \tilde{\boldsymbol{z}}_{\boldsymbol{kh}}) + \rho_m^*(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\boldsymbol{u_k} - \tilde{\boldsymbol{u}}_{\boldsymbol{kh}})\Big\},$$

but involve only those parts ρ_m and ρ_m^* of the global residuals ρ and ρ^* belonging to the subinterval I_m or to the initial time $t = 0$ for $m = 0$.

While the hereby obtained local contributions η_k^m for the temporal discretization error can directly be used for an adaptive refinement of the temporal discretization, the spatial contributions have to be localized further. However, a simple splitting into cell-wise contributions leads to a large overestimation of the actual error due to oscillatory behavior of the residuals (see Carstensen and Verfürth [28] for details on this). A commonly used way to overcome this difficulty is to apply cell-wise integration by parts in space (see, for example, Becker and Rannacher [12, 13]). The resulting local error indicators involve the strong residual of the equation as well as jumps of the discrete solution over faces of cells.

In Braack and Ern [20], a new way of overcoming this problem was presented without having to evaluate strong residuals and jumps over faces of cells and still obtaining the right local order of convergence. For simplicity, we restrict the description of this method to the case of a cG(s)dG(0) discretization. To this end, let us introduce the Lagrange nodal bases

$$\left\{ \varphi_i^m \;\middle|\; i = 1, \ldots, \dim V_h^{s,m} \right\}$$

of $V_h^{s,m}$ for $m = 1, \ldots, M$. By application of the operator $I_{2h}^{(2s)}$, these bases define the following set of bi- or tri-quadratic ($s = 1$) or bi- or tri-quartic ($s = 2$) nodal functions

$$\left\{ \psi_i^m := I_{2h}^{(2s)} \varphi_i^m \;\middle|\; i = 1, \ldots, \dim V_h^{s,m} \right\} \subseteq V_{2h}^{2s,m}.$$

Let $\boldsymbol{\Psi_m}$ and $\boldsymbol{\Psi_m^*}$ denote the difference of the primal and dual residual between the basis $\left\{ \boldsymbol{\varphi_i^m} \right\}$ of degree s and the basis $\left\{ \boldsymbol{\psi_i^m} \right\}$ of degree $2s$ of the corresponding product spaces $(V_h^{s,m})^d \times V_h^{s,m}$ and $(V_{2h}^{2s,m})^d \times V_{2h}^{2s,m}$, respectively. Thus, we have

$$\Psi_{m,i} = \rho_m(\boldsymbol{u_{kh}})(\boldsymbol{\psi_i^m} - \boldsymbol{\varphi_i^m}) \quad \text{and} \quad \Psi_{m,i}^* = \rho_m^*(\boldsymbol{u_{kh}}, \boldsymbol{z_{kh}})(\boldsymbol{\psi_i^m} - \boldsymbol{\varphi_i^m}).$$

Since for the considered discretization $\boldsymbol{u_{kh}}$ and $\boldsymbol{z_{kh}}$ are constant on each subinterval, we have

$$\boldsymbol{u_{kh}}\Big|_{I_m} = \sum_{i=1}^{N_m} U_i^m \boldsymbol{\varphi_i^m}, \qquad \boldsymbol{z_{kh}}\Big|_{I_m} = \sum_{i=1}^{N_m} Z_i^m \boldsymbol{\varphi_i^m},$$
$$\boldsymbol{I_{2h}^{(2s)}} \boldsymbol{u_{kh}}\Big|_{I_m} = \sum_{i=1}^{N_m} U_i^m \boldsymbol{\psi_i^m}, \quad \boldsymbol{I_{2h}^{(2s)}} \boldsymbol{z_{kh}}\Big|_{I_m} = \sum_{i=1}^{N_m} Z_i^m \boldsymbol{\psi_i^m},$$

where $N_m = \dim(V_h^{s,m})^d \times V_h^{s,m} = \dim(V_{2h}^{2s,m})^d \times V_{2h}^{2s,m}$ and $\boldsymbol{U^m}, \boldsymbol{Z^m} \in \mathbb{R}^{N_m}$ represent the nodal coefficient vectors of $\boldsymbol{u_{kh}}\Big|_{I_m}$ and $\boldsymbol{z_{kh}}\Big|_{I_m}$, respectively. By means of these definitions, we can express the error indicators in space as

$$\eta_h^m = \frac{1}{2}\left\{\langle \boldsymbol{\Psi_m}, \boldsymbol{Z^m}\rangle + \langle \boldsymbol{\Psi_m^*}, \boldsymbol{U^m}\rangle\right\}$$

with $\langle\cdot,\cdot\rangle$ being the Euclidean inner product on $\mathbb{R}^{N_m}$.

Let us further introduce a *filtering operator* $\boldsymbol{\pi}$ by

$$\boldsymbol{\pi} := \mathbf{id} - \boldsymbol{I_{2h}^{(s)}} \quad \text{with} \quad \boldsymbol{I_{2h}^{(s)}} : \widetilde{X}_{kh}^{0,s} \to \widetilde{X}_{k,2h}^{0,s},$$

see also the description of the local projection stabilization in Section 3.3.2. This construction again uses the patch-wise structure of the mesh. This structure also implies $V_{2h}^{s,m} \subseteq V_h^{s,m}$. Hence, for the filtered solutions $\boldsymbol{\pi u_{kh}}\Big|_{I_m}$ and $\boldsymbol{\pi z_{kh}}\Big|_{I_m}$, we also have an expression with respect to the nodal basis $\left\{\, \boldsymbol{\varphi_i^m} \,\right\}$, denoted by $\boldsymbol{U^{\pi,m}}$ and $\boldsymbol{Z^{\pi,m}}$, respectively:

$$\boldsymbol{\pi u_{kh}}\Big|_{I_m} = \sum_{i=1}^{N_m} U_i^{\pi,m}\boldsymbol{\varphi_i^m} \quad \text{and} \quad \boldsymbol{\pi z_{kh}}\Big|_{I_m} = \sum_{i=1}^{N_m} Z_i^{\pi,m}\boldsymbol{\varphi_i^m}.$$

Noting that $\boldsymbol{I_{2h}^{(2s)}}$ is the identity on $(V_{2h}^{s,m})^d \times V_{2h}^{s,m}$, we conclude

$$\begin{aligned}\boldsymbol{I_{2h}^{(2s)}}\boldsymbol{\pi}\boldsymbol{\varphi_i^m} - \boldsymbol{\pi}\boldsymbol{\varphi_i^m} &= \boldsymbol{I_{2h}^{(2s)}}\boldsymbol{\varphi_i^m} - \boldsymbol{I_{2h}^{(2s)}}\boldsymbol{I_{2h}^{(s)}}\boldsymbol{\varphi_i^m} - \boldsymbol{\varphi_i^m} + \boldsymbol{I_{2h}^{(s)}}\boldsymbol{\varphi_i^m} \\ &= \boldsymbol{I_{2h}^{(2s)}}\boldsymbol{\varphi_i^m} - \boldsymbol{\varphi_i^m} = \boldsymbol{\psi_i^m} - \boldsymbol{\varphi_i^m}.\end{aligned}$$

Then, using the linearity of the residuals in the second argument, we obtain by simple calculations (see Braack and Ern [20])

$$\eta_h^m = \frac{1}{2}\left\{\langle \boldsymbol{\Psi_m}, \boldsymbol{Z^m}\rangle + \langle \boldsymbol{\Psi_m^*}, \boldsymbol{U^m}\rangle\right\} = \frac{1}{2}\left\{\langle \boldsymbol{\Psi_m}, \boldsymbol{Z^{\pi,m}}\rangle + \langle \boldsymbol{\Psi_m^*}, \boldsymbol{U^{\pi,m}}\rangle\right\}.$$

This leads us to the definition of the local error indicators as

$$\eta_{h,i}^m := \frac{1}{2}\left\{\Psi_{m,i} Z_i^{\pi,m} + \Psi_{m,i}^* U_i^{\pi,m}\right\},$$

satisfying the following upper bound for the error estimator η_h^m:

$$|\eta_h^m| \le \sum_{i=1}^{N_m} \left|\eta_{h,i}^m\right|.$$

For setting up an efficient adaptive algorithm, it is essential that the temporal error estimator η_k is independent of the refinement of the spatial discretization and vice versa. We will see in Section 4.7, that this is (almost) the case for both η_k and η_h. However, the local contributions η_h^m depend linearly on the local size k_m of the subintervals I_m because

$$\eta_h = \sum_{m=0}^{M} \eta_h^m.$$

It is important to get rid of this dependence since otherwise the spatial error indicators would decrease, for instance, while keeping the spatial discretization fixed and only refining the temporal discretization. Therefore, we introduce spatial error indicators $\hat{\eta}_h^m$ independent of k_m and hence usable in a simultaneous mesh adaption algorithm as presented in Section 4.5 by rescaling:

$$\hat{\eta}_{h,i}^m \coloneqq \frac{T}{k_m} \eta_{h,i}^m, \quad i = 1, \dots, N_m, \quad m = 0, \dots, M.$$

This scaling has the following special property: If the rescaled spatial error indicators $\hat{\eta}_h^m$ fulfill

$$\hat{\eta}_h^m < \mathrm{TOL},$$

we then have for the whole spatial error estimator

$$\eta_h = \sum_{m=0}^{M} \eta_h^m = \sum_{m=0}^{M} \frac{k_m}{T} \hat{\eta}_h^m < \frac{\mathrm{TOL}}{T} \sum_{m=0}^{M} k_m = \mathrm{TOL}.$$

Remark 4.5. If the cells to be refined within an adaptive mesh refinement procedure are not chosen by a tolerance-based selection criterion, the simpler rescaling

$$\hat{\eta}_{h,i}^m \coloneqq \frac{1}{k_m} \eta_{h,i}^m$$

is sufficient.

Cell-wise error indicators $\hat{\eta}_{h,K}^m$ for cells $K \in \mathcal{T}_h^m$ usable in an adaptive mesh refinement procedure are obtained by reassembling the scaled node-wise contributions $\hat{\eta}_{h,i}^m$.

This supplies us with two sets of error indicators which will be used within the adaptive algorithm presented in the next section for an efficient automatic adaptation of the temporal and spatial discretizations. These sets are given by

$$\Sigma_k \coloneqq \{ \eta_k^m \mid m = 1, \dots, M \} \quad \text{and} \quad \Sigma_h \coloneqq \left\{ \hat{\eta}_{h,K}^m \;\middle|\; K \in \mathcal{T}_h^m,\ m = 0, \dots, M \right\}.$$

Remark 4.6. Note that for efficiency reasons it is necessary to treat the cell-wise error indicators of all spatial meshes simultaneously rather than for each mesh separately. If we used $M + 1$ different sets of cell-wise error indicators

$$\Sigma_h^m \coloneqq \left\{ \hat{\eta}_{h,K}^m \;\middle|\; K \in \mathcal{T}_h^m \right\}, \quad m = 0, \dots, M,$$

for deciding which cells should be refined, we would probably obtain a rather inefficient spatial discretization. This becomes clear if we assume, for example, that the error indicators on one subinterval are much smaller than those on another subinterval. Using, for instance, a fixed fraction strategy for selecting the cells to be refined leads to cells that are marked to be refined although the corresponding error indicators are smaller than the error indicators of cells in other meshes which might not be refined. The other way

around, we also observe inefficiency when marking cells for coarsening. Even if their error indicators might be small compared to other cells in the same mesh, their contribution to the spatial discretization error still might be large compared to cells in other meshes. However, applying a fixed fraction strategy to the full set of error indicators Σ_h does not produce such inefficient meshes because the error indicators are sorted "globally".

4.5 Adaptive algorithm

In this section, we present an adaptive refinement algorithm which uses the developed a posteriori error estimators of Section 4.2 to automatically adjust the temporal and spatial discretizations in order to efficiently increase the accuracy. To obtain efficient discretizations, it is essential to equilibrate the temporal and spatial discretization errors and keep them balanced under further refinement. This requires a precise quantitative assessment of both discretization errors as it is available with the derived error estimators (see Section 4.7 for numerical results).

If the functional value $J(\boldsymbol{u})$ is to be computed to a given accuracy TOL, this can be achieved by refining each discretization as long as the corresponding part of the error is greater than $\frac{\text{TOL}}{2}$. However, this might lead to an inefficient algorithm, especially in the case when the temporal and spatial discretization error are unbalanced in the beginning. Furthermore, the desired accuracy TOL might be too small to be achieved with the given computational resources. In the sequel, we present an adaptive algorithm which balances the initial temporal and spatial discretization errors and keeps them balanced during further refinement without having to prescribe a certain accuracy TOL. This leads to an algorithm which uses the given computational resources efficiently in order to achieve an accuracy as good as possible. The stopping criterion therefore is based on reaching a prescribed maximum number of degrees of freedom (determined by the given architecture) which must not be exceeded rather than on reaching the desired accuracy TOL. Of course, these stopping criterions can easily be exchanged by instead checking if

$$|\eta| = |\eta_k + \eta_h| \leq \text{TOL},$$

at least under consideration of the problem mentioned above.

As already mentioned, the goal of an efficient adaptive refinement algorithm has to be the equilibrated reduction of the temporal and spatial discretization error. To this end, we introduce an equilibration constant $\kappa \geq 1$ (usually $\kappa \approx 3$ in our numerical examples) and propose to proceed as in Algorithm 4.1.

Remark 4.7. The behavior of Algorithm 4.1 strongly depends on the choice of κ. Choosing κ too small, results in a slower reduction of the overall discretization error because only the temporal or spatial discretization is refined while the temporal and spatial discretization error actually are of the same size. On the other hand, choosing κ too large, makes the algorithm inefficient because both discretizations are refined although the total discretization error is dominated by only the temporal or the spatial discretization error. Numerical tests show $\kappa \approx 3$ to be a good choice.

Algorithm 4.1. Adaptive refinement algorithm

1: Choose an initial temporal and spatial discretization $\mathcal{T}_{k_0,h_0}$.
2: Set $n = 0$.
3: **loop**
4: Compute the primal and dual solution $\boldsymbol{u_{k_n h_n}}$ and $\boldsymbol{z_{k_n h_n}}$.
5: Evaluate the a posteriori error estimators η_{k_n} and η_{h_n}.
6: **if** the maximum number of degrees of freedom is reached **then**
7: **return**
8: **if** $|\eta_{k_n}| > \kappa\,|\eta_{h_n}|$ **then**
9: Adapt the temporal discretization.
10: **else if** $|\eta_{h_n}| > \kappa\,|\eta_{k_n}|$ **then**
11: Adapt the spatial discretization.
12: **else**
13: Adapt the temporal and spatial discretization.
14: Increase n.

When refining a discretization, the cells (or time intervals) which are to be refined are chosen using sets Σ_k or Σ_h of error indicators like the ones shown at the end of the previous section. Thus, we have to select subsets $\Sigma_k^R \subseteq \Sigma_k$ or $\Sigma_h^R \subseteq \Sigma_h$ indicating which cells (or time intervals) should be refined. As already noted in Remark 4.6, the selection of the spatial cells to be refined is done simultaneously on all meshes $\mathcal{T}_h^m$, $m = 0, \dots, M$.

For the selection of the subsets Σ_k^R or Σ_h^R, several standard approaches are available like error balancing or fixed fraction strategies. However, for the computations in this thesis, a quite different approach was used which is described in Richter [92, 93], for example. In the remaining part of this section, we will shortly present its main ideas based on the set of error indicators $\Sigma = \{\, \eta_1, \dots, \eta_N \,\}$ which is thought of as the localization of an error estimator η. In a first step, we compute a permutation $(i_1, \dots, i_N)$ of $(1, \dots, N)$ such that the local error indicators are sorted in descending order according to their absolute value:

$$|\eta_{i_1}| \geq \cdots \geq |\eta_{i_N}|\,.$$

Then, the subset $\Sigma^R = \{\, \eta_{i_1}, \dots, \eta_{i_r} \,\} \subseteq \Sigma$ is chosen as coherent queue where the index r is given by

$$r := \operatorname*{arg\,min}_{1 \leq r \leq N} \mathcal{E}(r)\mathcal{N}(r)^\beta. \tag{4.15}$$

Here, $\mathcal{E}(r)$ is a prediction of the error on the refined discretization which is given by

$$\mathcal{E}(r) = \sum_{i=1}^{N} |\eta_i| - \sum_{i=1}^{r} (1 - 2^{-\alpha})\,|\eta_i|\,.$$

The parameter α denotes the expected order of convergence, i. e., we assume

$$\sum_{K' \subseteq K} |\eta_{K'}| = 2^{-\alpha}\,|\eta_K|$$

after refining a cell K into 2^d cells K'. $\mathcal{N}(r)$ is the number of degrees of freedom of the refined discretization and the parameter β is given as the quotient of the order of the finite element space and the dimension d of the discretized domain. The optimal value of r is determined by successively testing (4.15) with $r = 1, \dots, N$.

More details can be found in Richter [93]. Braack [17] gives an analytical justification of this approach under certain regularity assumptions.

4.6 Heuristic error indicators

This section is dedicated to the derivation of heuristic error indicators. Such error indicators which can be evaluated without solving the additional dual problem are only smoothness based, of course. They are not able to quantitatively assess the discretization error. In addition, they cannot detect where large errors have large influence on the discretization error measured in a functional value and especially where this is not the case. Let us briefly present their derivation before we compare them in the next section to the quantitative error estimators developed in Section 4.2.

Let us for simplicity consider the stationary Poisson problem

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned}$$

An a posteriori error estimator assessing the error in the energy norm $\|\nabla(u - u_h)\|$ is given by

$$\|\nabla(u - u_h)\| \leq C \left(\sum_{K \in \mathcal{T}_h} h_K^2 \{\rho_K(u_h)^2 + \rho_{\partial K}(u_h)^2\} \right)^{\frac{1}{2}} \tag{4.16}$$

with the cell residuals $\rho_K(u_h)$ and jump residuals $\rho_{\partial K}(u_h)$ defined as

$$\rho_K(u_h) \coloneqq \|f + \Delta u_h\|_{L^2(K)} \quad \text{and} \quad \rho_{\partial K}(u_h) \coloneqq \frac{1}{2} h_K^{-\frac{1}{2}} \left\| [\partial_{\boldsymbol{n}} u_h] \right\|_{L^2(\partial K \setminus \partial\Omega)},$$

see, for example, Verfürth [101]. The value of the constant C appearing in (4.16) is in general unknown since it involves an interpolation constant C_I and a stability constant C_S. While the interpolation constant C_I can be approximated quite well, bounding the stability constant C_S would require bounds on higher order derivatives of the dual solution.

For the case of bi- or tri-linear finite elements it is known (Carstensen and Verfürth [28]) that the contributions of the cell residuals $\rho_K(u_h)$ can be neglected compared to the jump residuals $\rho_{\partial K}(u_h)$. Furthermore, the jumps of the normal derivatives $[\partial_{\boldsymbol{n}} u_h]$ over faces can be estimated by a recovery of the second derivatives on the cells. One possible way to do this is the application of the patch-wise interpolation operator $I_{2h}^{(2)}$ (see Section 4.3):

$$\rho_{\partial K}(u_h) \approx \|\nabla^2 I_{2h}^{(2)} u_h\|_{L^2(K)}.$$

This leads to the following error estimator:

$$\|\nabla(u-u_h)\| \lesssim C\left(\sum_{K\in\mathcal{T}_h}\tilde{\eta}_{h,K}^2\right)^{\frac{1}{2}}$$

with the error indicators

$$\tilde{\eta}_{h,K} \coloneqq h_K\|\nabla^2 I_{2h}^{(2)}u_h\|_{L^2(K)}.$$

Similar recovery techniques can be found, for instance, in Zienkiewicz and Zhu [110, 111].

Even though the derivation of this heuristic error indicator was based on a stationary problem, we use this error indicator for the spatial refinement in time-dependent problems to compare the results to those obtained by the application of the quantitative error estimator derived in Section 4.2.

In a similar fashion, one can derive an heuristic error indicator $\tilde{\eta}_k$ for the temporal refinement which for first-order methods like the dG(0) discretization in time is given by

$$\tilde{\eta}_{k,m} = k_m\left\|u_{kh}^m - u_{kh}^{m-1}\right\|.$$

4.7 Numerical results

In this section, we present some numerical results achieved by applying the proposed adaptive algorithm in combination with different temporal and spatial discretizations to the incompressible Navier-Stokes equations. After showing the very good quantitative assessment of both the temporal and spatial discretization errors with respect to a given functional in Section 4.7.1, we compare these results in Section 4.7.2 with those obtained by adaptive refinement controlled by the heuristic error indicators developed in Section 4.6.

To this end, let us consider the following model problem on the two-dimensional unit square $\Omega = (0,1)^2$ and final time $T = 1$: Find $(\boldsymbol{v}, p)^T$ such that

$$\begin{aligned}
\partial_t\boldsymbol{v} - \Delta\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p &= \boldsymbol{f} && \text{in } (0,1)\times\Omega,\\
\nabla\cdot\boldsymbol{v} &= 0 && \text{in } (0,1)\times\Omega,\\
\boldsymbol{v} &= 0 && \text{in } \{\,0\,\}\times\Omega,\\
\boldsymbol{v} &= 0 && \text{on } (0,1)\times\partial\Omega.
\end{aligned} \tag{4.17}$$

Let the force $\boldsymbol{f}$ be given in such a way that the exact solution $(\boldsymbol{v}, p)^T$ is given by

$$\begin{aligned}
\boldsymbol{v}(t,\boldsymbol{x}) &= \begin{pmatrix} \sin(t)\sin^2(\pi x_1)\sin(\pi x_2)\cos(\pi x_2)\\ -\sin(t)\sin(\pi x_1)\cos(\pi x_1)\sin^2(\pi x_2)\end{pmatrix},\\
p(t,\boldsymbol{x}) &= \sin(t)\sin(\pi x_1)\cos(\pi x_1)\sin(\pi x_2)\cos(\pi x_2).
\end{aligned}$$

We aim at computing the functional value

$$J(\boldsymbol{u}) = \frac{1}{2}\int_\Omega |\boldsymbol{v}(1,\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}$$

at final time $T = 1$. The exact value can be computed to be

$$J(\boldsymbol{u}) = \frac{3}{64}\sin^2(1) \approx 0.03319094148157365.$$

In our computations, the constants in the stabilization parameters α and δ are chosen as $\alpha_0 = \delta_0 = 0.2$.

4.7.1 Numerical results employing the quantitative error estimator

First, we present the numerical justification for the splitting of the total discretization error into a temporal and spatial contribution. In Tables 4.1–4.3, the independence of the temporal error estimator on the refinement of the spatial discretization and vice versa can be seen. This is an important feature in equilibrating both discretization errors during the adaptive algorithm presented in Section 4.5. Here and in the rest of this thesis, N denotes the number of degrees of freedom of one spatial mesh while M denotes the number of subintervals. Also note the very good agreement of the spatial error estimators between all three temporal discretizations (columns three and four in Tables 4.1–4.3) as well as the agreement of the temporal error estimators (columns five and six of these tables) when using either the cG(1) or the cG(2) discretization in space.

Table 4.1. Independence of one part of the error estimator on the refinement of the other discretization: dG(0) discretization in time, cG(1) or cG(2) discretization in space

N	M	η_h		η_k	
		cG(1)	cG(2)	cG(1)	cG(2)
243	40			$3.8136 \cdot 10^{-04}$	$4.2374 \cdot 10^{-04}$
867	40			$4.1703 \cdot 10^{-04}$	$4.2887 \cdot 10^{-04}$
3267	40	—	—	$4.2620 \cdot 10^{-04}$	$4.2922 \cdot 10^{-04}$
12675	40			$4.2848 \cdot 10^{-04}$	$4.2924 \cdot 10^{-04}$
49923	40			$4.2905 \cdot 10^{-04}$	$4.2924 \cdot 10^{-04}$
3267	10	$1.9876 \cdot 10^{-04}$	$1.5554 \cdot 10^{-06}$		
3267	20	$2.0636 \cdot 10^{-04}$	$1.6150 \cdot 10^{-06}$		
3267	40	$2.1009 \cdot 10^{-04}$	$1.6442 \cdot 10^{-06}$	—	—
3267	80	$2.1194 \cdot 10^{-04}$	$1.6586 \cdot 10^{-06}$		
3267	160	$2.1286 \cdot 10^{-04}$	$1.6656 \cdot 10^{-06}$		

In Tables 4.4–4.6, we present the development of the total discretization error $J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})$ as well as the spatial and temporal error estimators η_h and η_k during an adaptive run with local refinement of the temporal and spatial discretization using dynamic meshes for the cG(1)dG(0), cG(2)dG(1), and cG(2)cG(1) discretization, respectively. Here, $N_{\max}$ denotes the number of degrees of freedom of the finest spatial mesh used whereas N_{tot} denotes the total number of degrees of freedom of the space-time discretization. The last column shows

Table 4.2. Independence of one part of the error estimator on the refinement of the other discretization: dG(1) discretization in time, cG(1) or cG(2) discretization in space

N	M	η_h		η_k	
		cG(1)	cG(2)	cG(1)	cG(2)
243	40			$-5.4041 \cdot 10^{-07}$	$-5.7303 \cdot 10^{-07}$
867	40			$-5.6914 \cdot 10^{-07}$	$-5.7818 \cdot 10^{-07}$
3267	40	—	—	$-5.7621 \cdot 10^{-07}$	$-5.7851 \cdot 10^{-07}$
12675	40			$-5.7795 \cdot 10^{-07}$	$-5.7853 \cdot 10^{-07}$
49923	40			$-5.7839 \cdot 10^{-07}$	$-5.7853 \cdot 10^{-07}$
3267	10	$2.1389 \cdot 10^{-04}$	$1.6741 \cdot 10^{-06}$		
3267	20	$2.1379 \cdot 10^{-04}$	$1.6732 \cdot 10^{-06}$		
3267	40	$2.1377 \cdot 10^{-04}$	$1.6731 \cdot 10^{-06}$	—	—
3267	80	$2.1377 \cdot 10^{-04}$	$1.6730 \cdot 10^{-06}$		
3267	160	$2.1377 \cdot 10^{-04}$	$1.6728 \cdot 10^{-06}$		

Table 4.3. Independence of one part of the error estimator on the refinement of the other discretization: cG(1) discretization in time, cG(1) or cG(2) discretization in space

N	M	η_h		η_k	
		cG(1)	cG(2)	cG(1)	cG(2)
243	40			$-2.3392 \cdot 10^{-06}$	$-2.5911 \cdot 10^{-06}$
867	40			$-2.6453 \cdot 10^{-06}$	$-2.7200 \cdot 10^{-06}$
3267	40	—	—	$-2.7104 \cdot 10^{-06}$	$-2.7295 \cdot 10^{-06}$
12675	40			$-2.7254 \cdot 10^{-06}$	$-2.7302 \cdot 10^{-06}$
49923	40			$-2.7290 \cdot 10^{-06}$	$-2.7302 \cdot 10^{-06}$
3267	10	$2.1415 \cdot 10^{-04}$	$1.6755 \cdot 10^{-06}$		
3267	20	$2.1387 \cdot 10^{-04}$	$1.6731 \cdot 10^{-06}$		
3267	40	$2.1380 \cdot 10^{-04}$	$1.6724 \cdot 10^{-06}$	—	—
3267	80	$2.1378 \cdot 10^{-04}$	$1.6721 \cdot 10^{-06}$		
3267	160	$2.1377 \cdot 10^{-04}$	$1.6719 \cdot 10^{-06}$		

the effectivity index I_{eff} which is given by

$$I_{\text{eff}} \coloneqq \frac{J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})}{\eta_k + \eta_h}.$$

Looking at Tables 4.4 and 4.5, we observe for finer discretizations $I_{\text{eff}} \approx 1$ which shows the very good quantitative estimation of the discretization error. We also note the equilibration of the temporal and spatial discretization error achieved during refinement.

On the other hand, Table 4.6 shows very bad results. The error is even increasing under refinement of the discretizations. This is due to oscillations occurring in the solution under change of the mesh. This instability for rough data is well known for the Crank-Nicolson scheme, see, for example, Luskin and Rannacher [77], Rannacher [91], or Heywood and Rannacher [66]. It is mainly caused by the missing strong A-stability of the Crank-Nicolson scheme. This can be seen by looking at the results for a computation on fixed spatial meshes, shown in Table 4.7. Due to this instability, we are going to use the cG(s)cG(1) discretization only on fixed spatial meshes.

Table 4.4. Adaptive refinement on dynamic meshes with equilibration for the cG(1)dG(0) discretization

N_{tot}	$N_{\max}$	M	η_h	η_k	$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})$	I_{eff}
2673	243	10	$2.82 \cdot 10^{-03}$	$1.39 \cdot 10^{-03}$	$5.42 \cdot 10^{-03}$	1.29
5655	867	12	$8.26 \cdot 10^{-04}$	$8.33 \cdot 10^{-04}$	$1.96 \cdot 10^{-03}$	1.18
18621	3267	14	$2.25 \cdot 10^{-04}$	$5.04 \cdot 10^{-04}$	$8.05 \cdot 10^{-04}$	1.10
91113	12435	18	$6.32 \cdot 10^{-05}$	$2.71 \cdot 10^{-04}$	$3.57 \cdot 10^{-04}$	1.07
162657	12435	26	$6.07 \cdot 10^{-05}$	$1.41 \cdot 10^{-04}$	$2.08 \cdot 10^{-04}$	1.03
767913	47859	34	$1.94 \cdot 10^{-05}$	$8.51 \cdot 10^{-05}$	$1.03 \cdot 10^{-04}$	0.98
1402389	47859	54	$1.87 \cdot 10^{-05}$	$4.48 \cdot 10^{-05}$	$6.23 \cdot 10^{-05}$	0.98
7419177	177627	82	$6.36 \cdot 10^{-06}$	$2.63 \cdot 10^{-05}$	$3.18 \cdot 10^{-05}$	0.97

Table 4.5. Adaptive refinement on dynamic meshes with equilibration for the cG(2)dG(1) discretization

N_{tot}	$N_{\max}$	M	η_h	η_k	$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})$	I_{eff}
5103	243	10	$4.49 \cdot 10^{-04}$	$-1.71 \cdot 10^{-05}$	$3.84 \cdot 10^{-04}$	0.89
6975	867	10	$3.03 \cdot 10^{-05}$	$-2.89 \cdot 10^{-05}$	$4.12 \cdot 10^{-06}$	2.81
27243	3267	12	$1.68 \cdot 10^{-06}$	$-3.47 \cdot 10^{-06}$	$-2.43 \cdot 10^{-06}$	1.35
106167	12675	14	$1.12 \cdot 10^{-07}$	$-6.20 \cdot 10^{-07}$	$-5.12 \cdot 10^{-07}$	1.01
233067	12675	20	$1.21 \cdot 10^{-07}$	$-8.82 \cdot 10^{-08}$	$3.76 \cdot 10^{-08}$	1.15
1028019	49923	24	$1.01 \cdot 10^{-08}$	$-1.77 \cdot 10^{-08}$	$-8.78 \cdot 10^{-09}$	1.15
7651347	197571	40	$7.57 \cdot 10^{-10}$	$-2.56 \cdot 10^{-09}$	$-1.92 \cdot 10^{-09}$	1.07

A comparison of different refinement strategies for the cG(1)dG(0), cG(2)dG(1), and cG(2)cG(1) discretization is depicted in Figures 4.2–4.4. We use the following labeling:

Table 4.6. Adaptive refinement on dynamic meshes with equilibration for the cG(2)cG(1) discretization

N_{tot}	$N_{\max}$	M	η_h	η_k	$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})$	I_{eff}
9537	867	10	$2.93 \cdot 10^{-05}$	$-3.23 \cdot 10^{-05}$	$-2.37 \cdot 10^{-05}$	7.76
41805	3267	14	$1.76 \cdot 10^{-06}$	$-9.67 \cdot 10^{-06}$	$-1.17 \cdot 10^{-05}$	1.48
48339	3267	16	$1.76 \cdot 10^{-06}$	$-2.74 \cdot 10^{-06}$	$-1.44 \cdot 10^{-06}$	1.48
156201	12675	18	$-7.22 \cdot 10^{-07}$	$-1.02 \cdot 10^{-05}$	$-4.67 \cdot 10^{-06}$	0.43
274119	12675	28	$-1.58 \cdot 10^{-06}$	$-8.23 \cdot 10^{-06}$	$3.09 \cdot 10^{-07}$	-0.03
546075	12675	56	$-8.51 \cdot 10^{-06}$	$-2.24 \cdot 10^{-05}$	$7.13 \cdot 10^{-07}$	-0.02
2773779	49923	112	$2.43 \cdot 10^{-05}$	$-2.51 \cdot 10^{-04}$	$1.63 \cdot 10^{-04}$	-0.72

Table 4.7. Adaptive refinement on fixed spatial meshes with equilibration for the cG(2)cG(1) discretization

N	M	η_h	η_k	$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})$	I_{eff}
867	10	$2.93 \cdot 10^{-05}$	$-3.23 \cdot 10^{-05}$	$-2.37 \cdot 10^{-05}$	7.76
3267	14	$1.68 \cdot 10^{-06}$	$-9.66 \cdot 10^{-06}$	$-1.17 \cdot 10^{-05}$	1.46
3267	16	$1.67 \cdot 10^{-06}$	$-2.72 \cdot 10^{-06}$	$-1.45 \cdot 10^{-06}$	1.37
12675	18	$1.01 \cdot 10^{-07}$	$-8.57 \cdot 10^{-07}$	$-9.19 \cdot 10^{-07}$	1.21
12675	26	$1.03 \cdot 10^{-07}$	$-2.43 \cdot 10^{-07}$	$-1.43 \cdot 10^{-07}$	1.02
49923	40	$6.41 \cdot 10^{-09}$	$-6.47 \cdot 10^{-08}$	$-6.07 \cdot 10^{-08}$	1.04
49923	62	$6.46 \cdot 10^{-09}$	$-1.95 \cdot 10^{-08}$	$-1.27 \cdot 10^{-08}$	0.97

- "uniform": We apply uniform refinement to the temporal and spatial discretization in each iteration.
- "adaptive": We apply adaptive refinement to the temporal and spatial discretization together with the proposed equilibration strategy. The spatial mesh is fixed on the whole time interval.
- "dynamic": We apply adaptive refinement to the temporal and spatial discretization together with the proposed equilibration strategy. The spatial meshes may vary from subinterval to subinterval.

Even for this example with smooth solution, we achieve a reduction factor of about 50–100 in the degrees of freedom needed for reaching a certain accuracy when using adaptive refinement in time and space compared to uniform meshes. In situations where the dynamics in space are larger than here, one can also achieve a greater reduction factor between the adaptive refinement on a fixed spatial mesh and dynamic meshes.

The size of the time steps obtained in the last iteration for the cG(1)dG(0) discretization is depicted in Figure 4.5. The other discretizations lead to similar adaptive refinement at the end of the time interval. This is not very surprising since the functional J only acts at final time $T = 1$ and the solution is mainly driven by the force $\boldsymbol{f}$ and not by problem inherent dynamics.

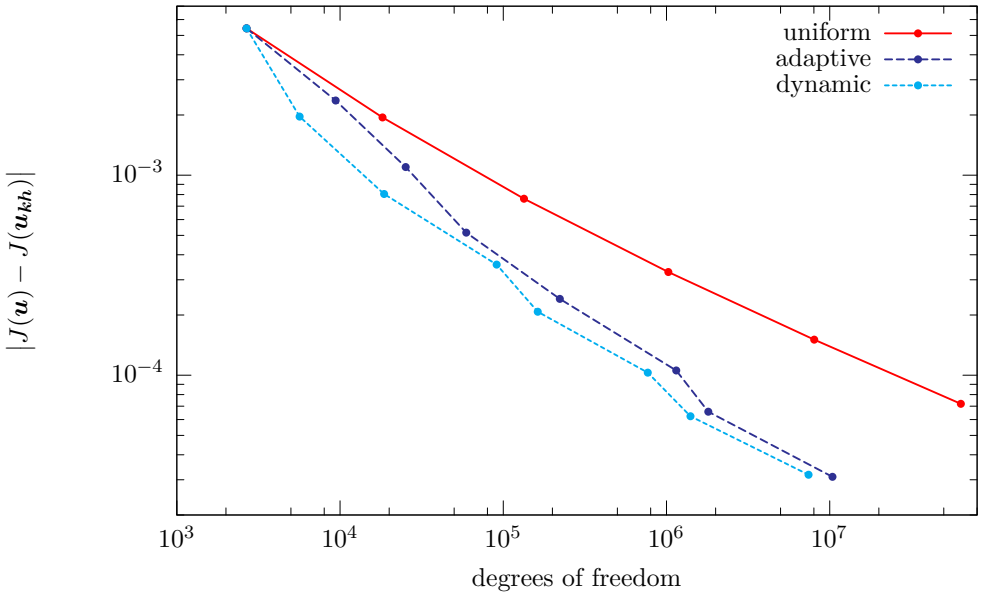


Figure 4.2. Comparison of the error $|J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})|$ for different refinement strategies with the cG(1)dG(0) discretization

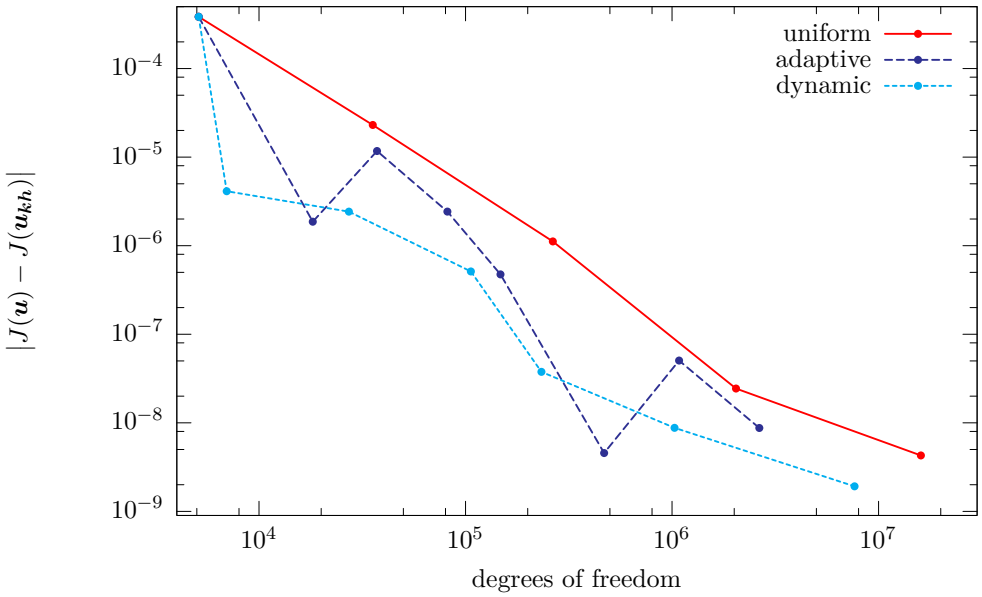


Figure 4.3. Comparison of the error $|J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})|$ for different refinement strategies with the cG(2)dG(1) discretization

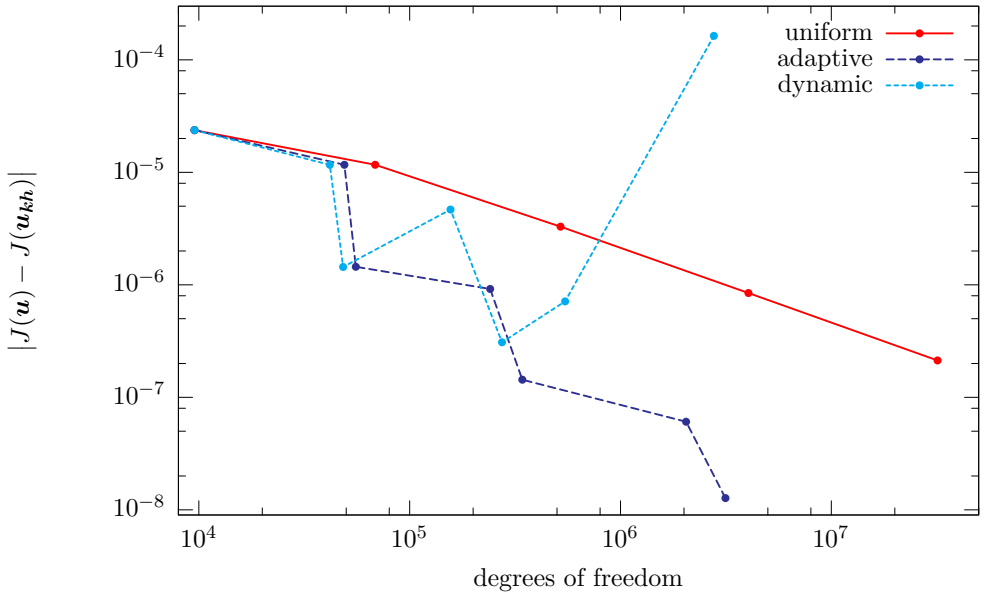


Figure 4.4. Comparison of the error $|J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})|$ for different refinement strategies with the cG(2)cG(1) discretization

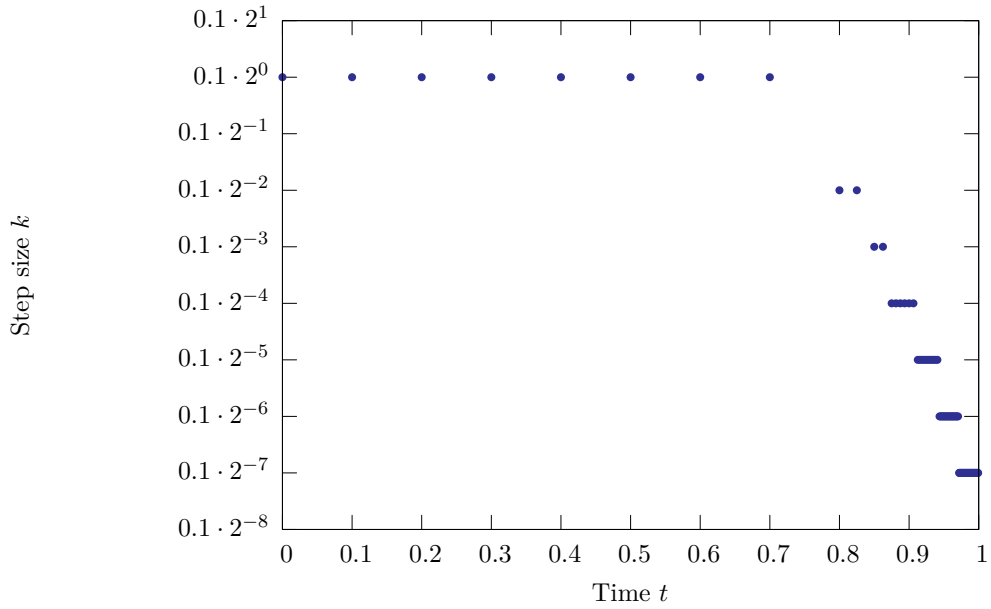


Figure 4.5. Adaptively determined time step size k with the cG(1)dG(0) discretization

Finally, we show in Figure 4.6 exemplarily for the cG(1)dG(0) discretization a sequence of adaptively refined meshes obtained in the last iteration using dynamic meshes. Note that the mesh is much more refined to the end of the time interval. Actually, the mesh is kept coarse and constant for $t \in [0, 0.6]$.

4.7.2 Comparison with heuristic error indicators

In this section, we compare the results of the quantitative error estimator of Section 4.2 with those of the heuristic error indicators of Section 4.6. The benefit of the heuristic error indicators is that we do not need to solve the dual problem additionally, but can evaluate the indicators by only using information of the primal solution. To compare the quantitative error estimator and the heuristic error indicator, we reconsider model problem (4.17). The results presented in this section are obtained using the cG(1)dG(0) discretization.

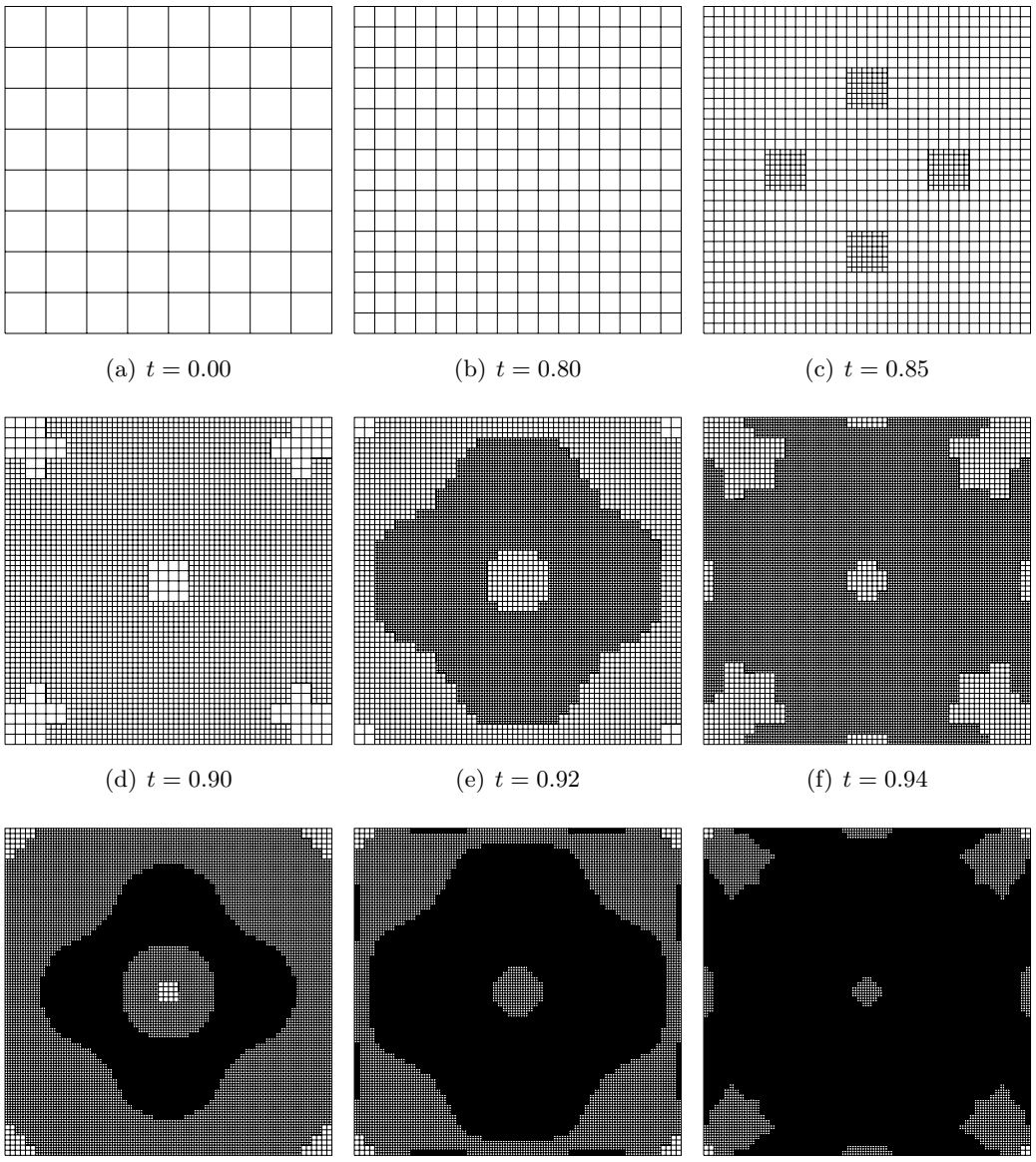
First, we choose a rather fine temporal discretization ($M = 640$) which yields a temporal discretization error of approximately $3 \cdot 10^{-5}$ and apply both the quantitative error estimator η_h and the heuristic error indicator $\tilde{\eta}_h$ to adaptively refine the spatial meshes. We allow dynamic change of the meshes. The results are depicted in Figure 4.7.

We observe that the meshes created on the basis of the quantitative error estimator η_h are much more efficient in reducing the discretization error than those created by the heuristic error indicator. Figure 4.7 even shows a slightly better order of convergence on the meshes built by the quantitative error estimator. Furthermore, the reduction of the discretization error is "smoother" when applying the quantitative error estimator.

We get a hint for the slower reduction of the discretization error when using the heuristic error indicator as refinement criterion if we look at the meshes created by it. Figure 4.8 shows a sequence of meshes created by the quantitative error estimator η_h and the heuristic error indicator $\tilde{\eta}_h$ after five refinement cycles which both lead to a space-time discretization with approximately $2 \cdot 10^6$ degrees of freedom.

The heuristic error indicator leads to a stronger refinement already at the beginning of the time interval, whereas the quantitative error estimator here leads to rather coarse meshes. In contrast, the situation near final time $T = 1$ is completely contrary. Here, the heuristic error indicator leads to coarser meshes than the quantitative error estimator. Since the functional J only evaluates for $T = 1$, this is not an efficient discretization. This behavior although is clear since the heuristic error indicator only measures the smoothness of the solution and cannot incorporate sensitivity information from the dual solution.

Now, we choose a fixed spatial mesh with 12675 degrees of freedom leading to a spatial discretization error of about $5 \cdot 10^{-5}$ and apply adaptive refinement in time by means of either the quantitative error estimator η_k or the heuristic error indicator $\tilde{\eta}_k$. The results are shown in Figure 4.9. Again, we note a better order of convergence using the quantitative error estimator to trigger the refinement of the temporal discretization compared to the refinement based on the heuristic error indicator. These results become clear when noting



(a) $t = 0.00$ (b) $t = 0.80$ (c) $t = 0.85$

(d) $t = 0.90$ (e) $t = 0.92$ (f) $t = 0.94$

(g) $t = 0.96$ (h) $t = 0.98$ (i) $t = 1.00$

Figure 4.6. Spatial meshes at different time points obtained with the cG(1)dG(0) discretization

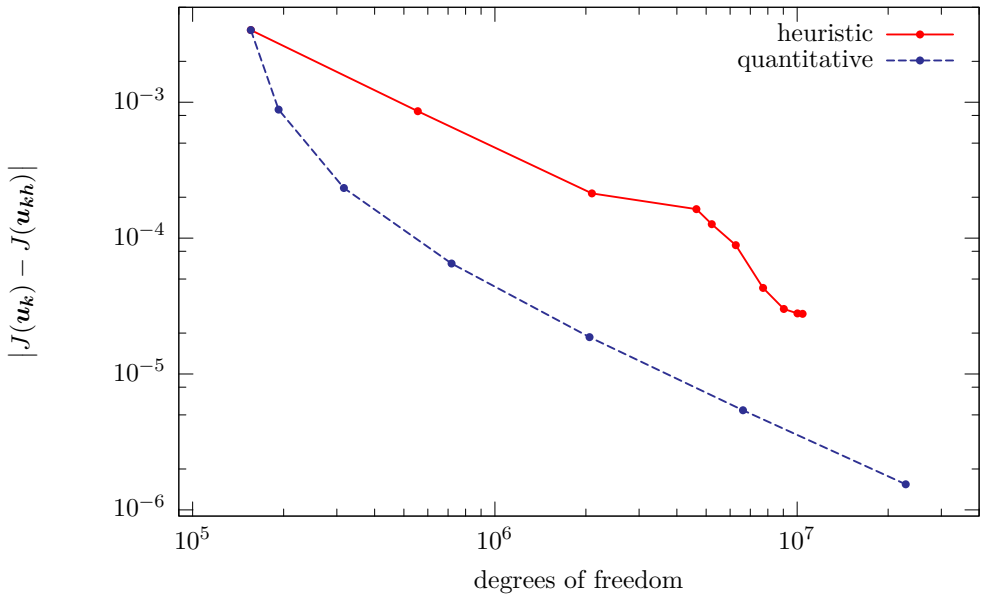
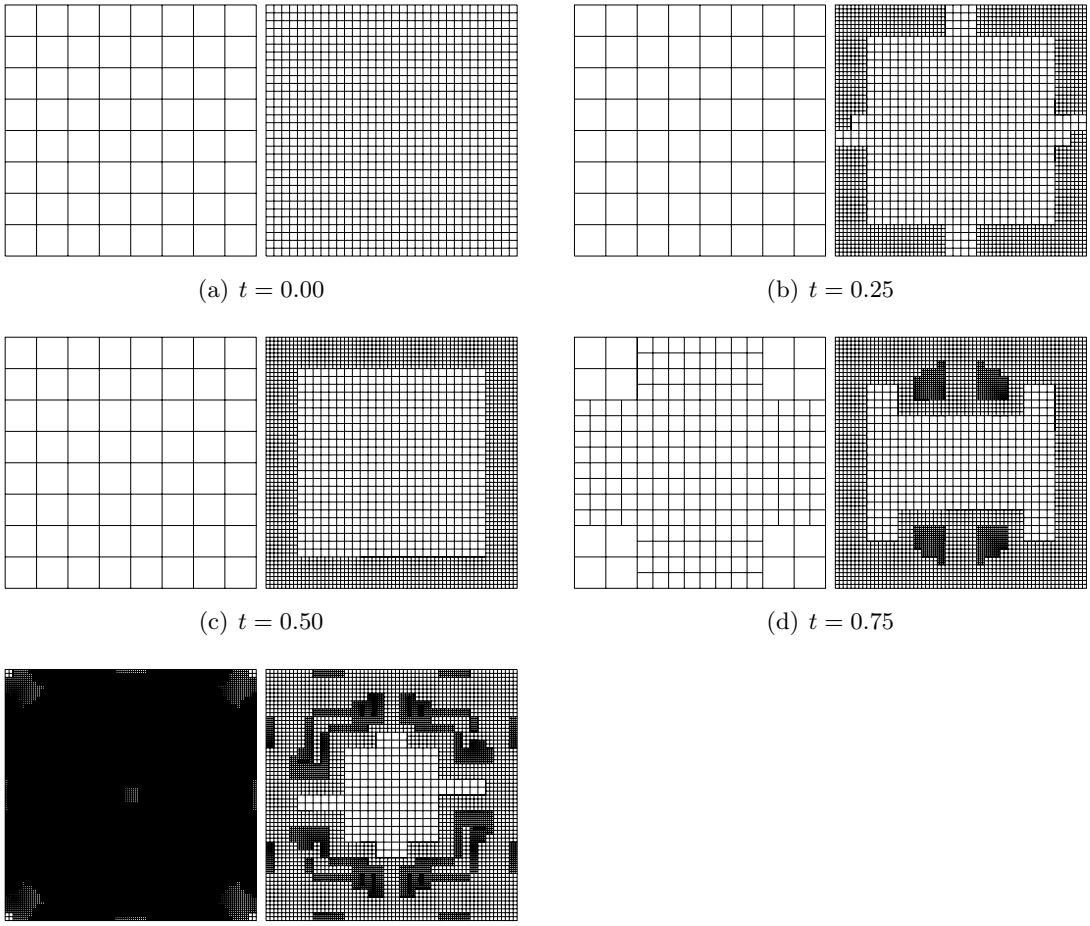


Figure 4.7. Comparison of the discretization errors $|J(\boldsymbol{u_k}) - J(\boldsymbol{u_{kh}})|$ obtained by adaptive spatial refinement using the quantitative error estimator η_h or the heuristic error indicator $\tilde{\eta}_h$

that the quantitative error estimator leads to adaptive refinement of the time interval only near final time $T = 1$ whereas the heuristic error indicator in fact leads to uniform refinement of the time interval. This is more or less obvious since the temporal smoothness of the exact solution is quite similar on the whole time interval $I = (0, 1)$. Only the quantitative error estimator η_k detects by using sensitivity information from the dual solution that a large discretization error near $t = 0$ (nearly) does not influence the accuracy of the functional value $J(\boldsymbol{u_{kh}})$ which only acts on final time $T = 1$. Hence, the quantitative error estimator leads to an incomparably more efficient discretization.

The most severe drawback of the heuristic error indicators is that they are not able to quantitatively assess the discretization error. This is due to the fact that the constant C in their development usually is unknown. Hence, it is not possible to use $\tilde{\eta}_h$ or $\tilde{\eta}_k$ in an adaptive algorithm like Algorithm 4.1 to equilibrate the temporal and spatial contribution of the total discretization error. Thus, using only heuristic error indicators may be suitable for stationary problems, but in the context of time-dependent problems they hardly lead to efficient adaptive discretizations. To obtain efficient discretizations for nonstationary problems, the adaptive refinement of the temporal and spatial discretization has to be based on quantitative error estimators like the ones developed in Section 4.2.

We have seen that the use of the proposed quantitative error estimators leads to more efficient discretizations. This quickly compensates the additional computational costs due to the additional dual problem that has to be solved and finally saves time when doing simulations in computational fluid dynamics.



(a) $t = 0.00$ (b) $t = 0.25$

(c) $t = 0.50$ (d) $t = 0.75$

(e) $t = 1.00$

Figure 4.8. Comparison of spatial meshes created by the quantitative error estimator η_h (left) and the heuristic error indicator $\tilde{\eta}_h$ (right) after five refinement cycles using dynamic meshes with the cG(1)dG(0) discretization

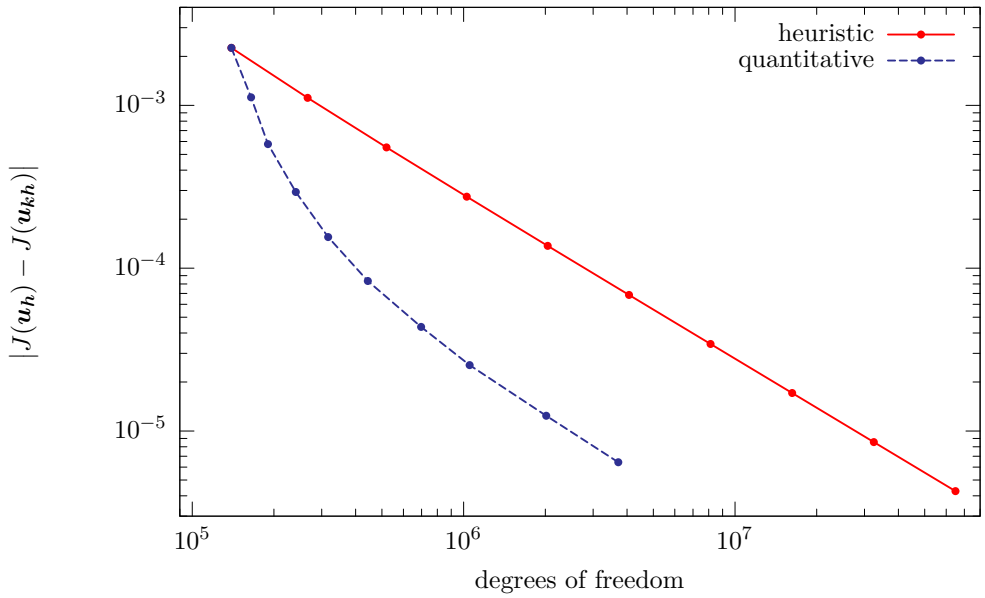


Figure 4.9. Comparison of the discretization errors $|J(\boldsymbol{u}_h) - J(\boldsymbol{u}_{kh})|$ obtained by adaptive temporal refinement using the quantitative error estimator η_k or the heuristic error indicator $\tilde{\eta}_k$

5 Issues on Dynamic Meshes

In this chapter, we discuss a particular problem which arises when using dynamic meshes in the approximation of solutions of the incompressible Navier-Stokes equations. The outline of this chapter is as follows: In the first Section 5.1, we present a simulation of the benchmark problem "Laminar Flow Around a Cylinder" (see Schäfer and Turek [95] and Section 6.1) using the cG(1)dG(1) discretization on dynamic meshes. In Section 5.2, we show that the effects shown in Section 5.1 are not specifically related to the Navier-Stokes equations, but can already be seen when solving the linear Stokes equations on dynamic meshes. We will see that while the approximations of the velocities are quite satisfactory, the approximation of the pressure on dynamic meshes deteriorates. Section 5.3 focuses on the precise numerical analysis of the error that occurs after changing the spatial mesh. For this, we apply the cG(s)dG(0) and the cG(s)dG(1) discretization to two model problems. We especially investigate the behavior of the error under systematic refinement of the temporal and the spatial discretization. Furthermore, we present some results using the inf-sup-stable $\mathcal{Q}_2/\mathcal{Q}_1$-Taylor-Hood element (see, for instance, Hood and Taylor [69]) for the spatial discretization to show that the effects are not induced by stabilization. In addition to the already presented Galerkin finite element discretizations in time, we also study the fractional-step-θ scheme (see, for example, Müller-Urbaniak [81]) in combination with dynamic meshes which is a widely used finite difference scheme for the temporal discretization in computational fluid dynamics. The following Section 5.4 then presents some attempts to solve the problem discussed in this chapter. In the final Section 5.5, we analytically investigate the phenomenon discussed in this chapter.

5.1 Description of the problem

In this section, we aim at computing the mean lift-coefficient in the two-dimensional benchmark problem "Laminar Flow Around a Cylinder", see Schäfer and Turek [95] and Section 6.1.1 for a detailed description of the setting as well as the precise formula for the lift-coefficient.

The time-dependent inflow condition is given by

$$v_1(t,\boldsymbol{x}) = \frac{6\sin(\frac{\pi t}{8\,\mathrm{s}})}{(0.41\,\mathrm{m})^2} x_2(0.41\,\mathrm{m} - x_2)\,\mathrm{m\,s^{-1}}, \quad v_2(t,\boldsymbol{x}) = 0\,\mathrm{m\,s^{-1}}$$

which yields a time-dependent Reynolds number of $0 \leq \mathrm{Re}(t) \leq 100$ for $t \in I = (0\,\mathrm{s}, 8\,\mathrm{s})$.

In this example, we apply the cG(1)dG(1) discretization. After five iterations of adaptive temporal and spatial refinement using dynamic meshes, the temporal evolution of the lift-coefficient looks as depicted in Figure 5.1.

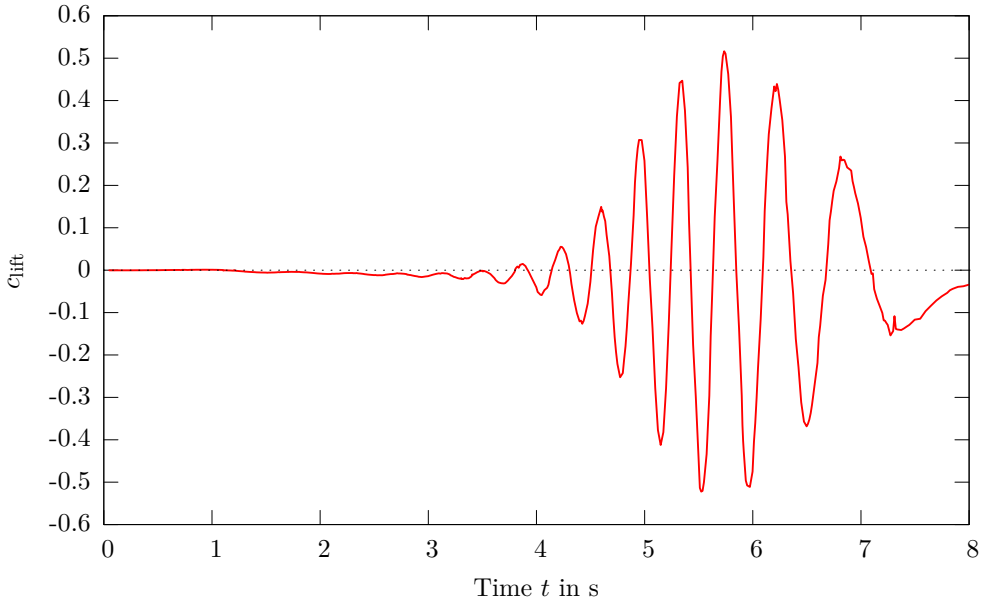


Figure 5.1. Lift-coefficient c_{lift} after five adaptation cycles

We note slight oscillations in the lift-coefficient, for example, near $t = 7.25\,\text{s}$. Further investigations show that such oscillations especially occur when switching from one spatial mesh to another. In the following sections, we are going to numerically analyze these oscillations.

5.2 Reduction to model problem

In this section, we will show that the oscillations in the lift-coefficient shown in the last section are caused by errors solely located in the discrete pressure when switching from one spatial mesh to another. This effect can already be observed when solving the linear Stokes equations instead of the nonlinear Navier-Stokes equations. Furthermore, these errors also arise when applying uniform refinement of a mesh. We state two model problems: one on a domain with curved boundary and one on a polygonally bounded domain.

Let us first show that the arising problems are not related to the time-dependent inflow boundary condition. To this end, we remove the oscillatory sine-term from the inflow condition and reduce the inflow velocity to

$$v_1(t, \boldsymbol{x}) = \frac{1.2}{(0.41\,\text{m})^2} x_2(0.41\,\text{m} - x_2)\,\text{m}\,\text{s}^{-1}, \quad v_2(t, \boldsymbol{x}) = 0\,\text{m}\,\text{s}^{-1}.$$

This yields a constant Reynolds number of $\mathrm{Re} = 20$ with a stationary solution. Since the effects we want to study can already be seen when working with the simplest temporal discretization, we apply the cG(1)dG(0) discretization which here coincides with the backward Euler scheme since there are no forces $\boldsymbol{f}$ depending on time. We now focus on computing the drag-coefficient whose reference value in this configuration is given as $c_{\mathrm{drag}}^{(\mathrm{ref})} = 5.57953523384$, see, for example, Nabh [82].

We use the time interval $I = (0\,\mathrm{s}, 8\,\mathrm{s})$ with different spatial meshes. Denoting the meshes on $(1\,\mathrm{s}, 2\,\mathrm{s}] \cup (3\,\mathrm{s}, 4\,\mathrm{s}] \cup (5\,\mathrm{s}, 6\,\mathrm{s}] \cup (7\,\mathrm{s}, 8\,\mathrm{s}]$ with $\mathcal{T}_h$, we use the mesh $\mathcal{T}_{2h}$ on $[0\,\mathrm{s}, 1\,\mathrm{s}] \cup (2\,\mathrm{s}, 3\,\mathrm{s}] \cup (4\,\mathrm{s}, 5\,\mathrm{s}] \cup (6\,\mathrm{s}, 7\,\mathrm{s}]$. That is, we perform uniform refinement of the spatial mesh at $t = 1\,\mathrm{s}$, $t = 3\,\mathrm{s}$, $t = 5\,\mathrm{s}$, and $t = 7\,\mathrm{s}$, whereas at $t = 2\,\mathrm{s}$, $t = 4\,\mathrm{s}$, and $t = 6\,\mathrm{s}$ uniform coarsening is applied. The evolution of the drag-coefficient for a uniform time step size of $k = 6.25 \cdot 10^{-3}\,\mathrm{s}$ on the time interval $[0.5\,\mathrm{s}, 4.5\,\mathrm{s}]$ is shown in Figure 5.2. We neglect the beginning of the time interval where a singularity in the pressure evolves for $t \to 0$ due to compatibility conditions that are not fulfilled with the initial condition $\boldsymbol{v}^{\mathbf{0}} = \mathbf{0}$, see, for instance, Heywood and Rannacher [65]. The evolution for $t > 4.5\,\mathrm{s}$ is exactly the same as shown in Figure 5.2, since the discrete solution always reaches the steady state on $\mathcal{T}_h$ or $\mathcal{T}_{2h}$ before the spatial mesh is changed again. The spatial meshes $\mathcal{T}_{2h}$ and $\mathcal{T}_h$ used in these computations are depicted in Figure 5.3.

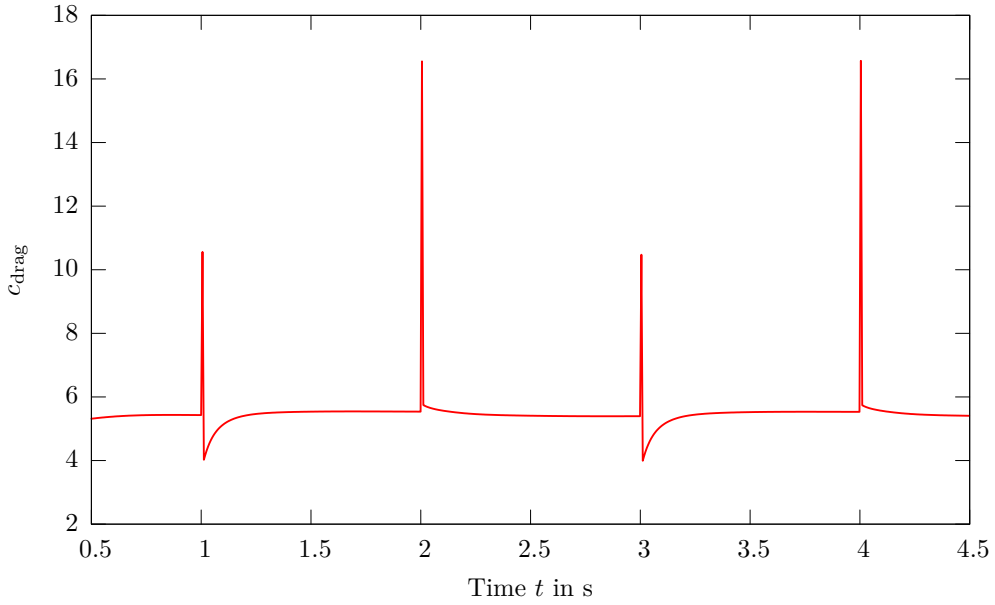
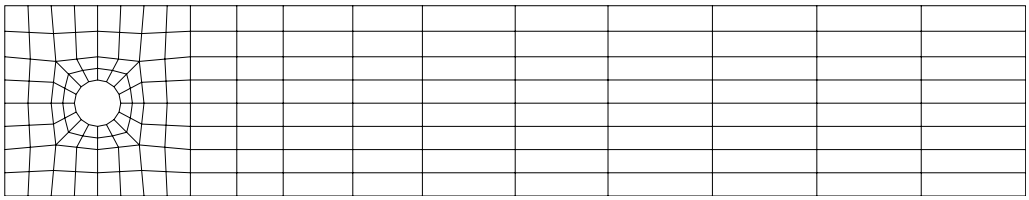


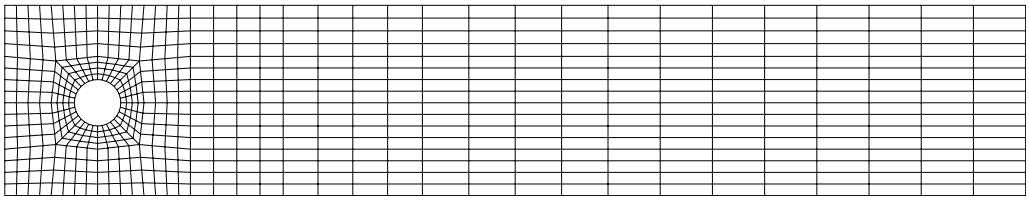
Figure 5.2. Temporal evolution of the drag-coefficient

We observe that precisely in the first time step on the new mesh the drag-coefficient deteriorates. We also note that these errors are even larger than in the example presented in the previous section.

In the remaining part of this section, we will show that these errors occurring when switching the spatial mesh are solely located in the discrete pressure. Furthermore, we will



(a) Mesh $\mathcal{T}_{2h}$



(b) Mesh $\mathcal{T}_h$

Figure 5.3. Spatial meshes used for the computation of the drag-coefficient

show that these effects are not specifically related to the nonlinearity of the Navier-Stokes equations or to higher Reynolds numbers, but also arise when solving the linear Stokes equations with $\nu = 1$. Therefore, we introduce two linear model problems with known analytical (stationary) solution: one on a domain with curved boundary and one on a polygonally bounded domain. These simple configurations allow us to study the presented phenomenon. For sake of simplicity, we use the cG(1)dG(0) discretization here, too.

Hence, we want to find $(\boldsymbol{v}, p)^T$ such that

$$\begin{aligned} \partial_t \boldsymbol{v} - \Delta \boldsymbol{v} + \nabla p &= \boldsymbol{f} && \text{in } I \times \Omega, \\ \nabla \cdot \boldsymbol{v} &= 0 && \text{in } I \times \Omega, \\ \boldsymbol{v} &= \boldsymbol{0} && \text{in } \{\, 0 \,\} \times \Omega, \\ \boldsymbol{v} &= \boldsymbol{0} && \text{on } I \times \partial\Omega. \end{aligned} \tag{5.1}$$

The two configurations mentioned above are the following ones:

Configuration I: Let $I = (0, 9)$ and $\Omega = (-1, 1)^2$. Let $\boldsymbol{f}$ be given in such a way that (5.1) possesses the stationary solution

$$\begin{aligned} \boldsymbol{v}(\boldsymbol{x}) &= \begin{pmatrix} \cos^2(\frac{\pi x_1}{2}) \cos(\frac{\pi x_2}{2}) \sin(\frac{\pi x_2}{2}) \\ -\cos(\frac{\pi x_1}{2}) \sin(\frac{\pi x_1}{2}) \cos^2(\frac{\pi x_2}{2}) \end{pmatrix}, \\ p(\boldsymbol{x}) &= \cos(\tfrac{\pi x_1}{2}) \sin(\tfrac{\pi x_1}{2}) \cos(\tfrac{\pi x_2}{2}) \sin(\tfrac{\pi x_2}{2}). \end{aligned}$$

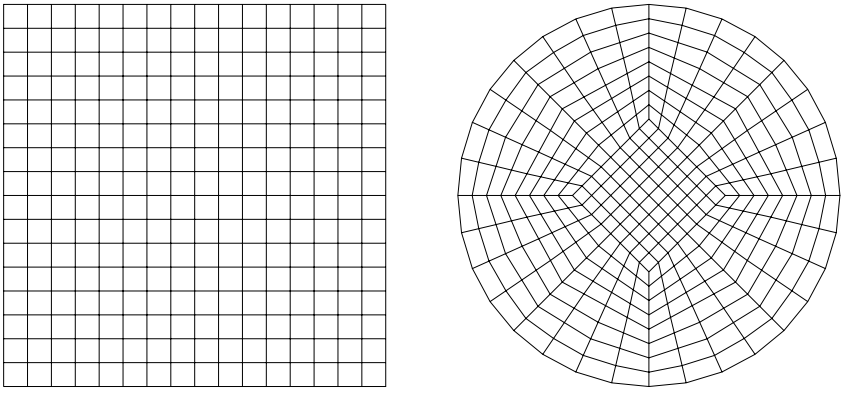
Configuration II: Let $I = (0, 9)$ and $\Omega = B_1(\mathbf{0}) \subseteq \mathbb{R}^2$. Let $\boldsymbol{f}$ be given in such a way that (5.1) possesses the stationary solution

$$\boldsymbol{v}(\boldsymbol{x}) = \begin{pmatrix} -x_2(1 - |\boldsymbol{x}|^2) \\ x_1(1 - |\boldsymbol{x}|^2) \end{pmatrix},$$
$$p(\boldsymbol{x}) = \cos(\pi\,|\boldsymbol{x}|^2).$$

We subdivide $\bar{I} = I^{(1)} \cup I^{(2)} \cup I^{(3)}$ with

$$I^{(1)} = [0, 3], \quad I^{(2)} = (3, 6], \quad I^{(3)} = (6, 9].$$

On $I^{(1)}$ and $I^{(3)}$ we use a uniform spatial mesh of cell size $2h$, whereas on the subinterval $I^{(2)}$ a uniform spatial mesh of cell size h is used. That is, we switch the spatial mesh uniformly from $2h$ to h at $t = 3$ and from h to $2h$ at $t = 6$. The subintervals are chosen long enough for the discrete solution to reach the stationary limit on each mesh. The spatial meshes $\mathcal{T}_h$ used on the subinterval $I^{(2)}$ are depicted in Figure 5.4.

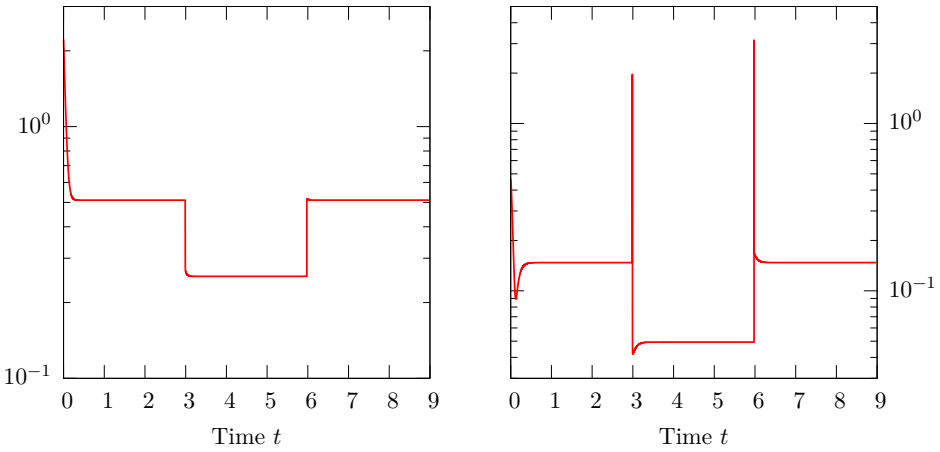


(a) Configuration I (b) Configuration II

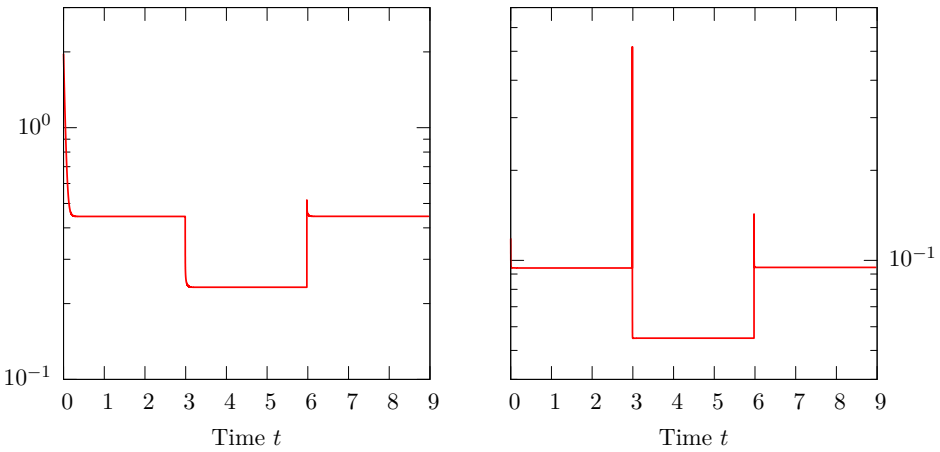
Figure 5.4. Spatial meshes $\mathcal{T}_h$ used on $I^{(2)}$

The errors $\|\nabla(\boldsymbol{v} - \boldsymbol{v_{kh}})\|$ and $\|p - p_{kh}\|$ for a uniform step size of $k \approx 2 \cdot 10^{-4}$ are shown in Figure 5.5. As we can see, both the velocity and the pressure approximation show a transient phenomenon when switching the spatial mesh. However, while the approximation of the velocity component is quite satisfactory, the transient phenomenon in the pressure component is superposed by an additional error which causes the approximation of the pressure to deteriorate under a change of the spatial mesh, see Figure 5.6. The larger errors near $t = 0$ stem from the fact that we do not start the simulation with the stationary solution, but rather with $\boldsymbol{v^0} = \mathbf{0}$. Hence the error compared to the stationary limit is large. We can conclude that the effect seen here does not originate from the approximation of the curved boundary, since we observe the same behavior in Configuration I and Configuration II.

In the next section, we will further analyze how these errors behave under systematic refinement of the temporal and spatial discretization.

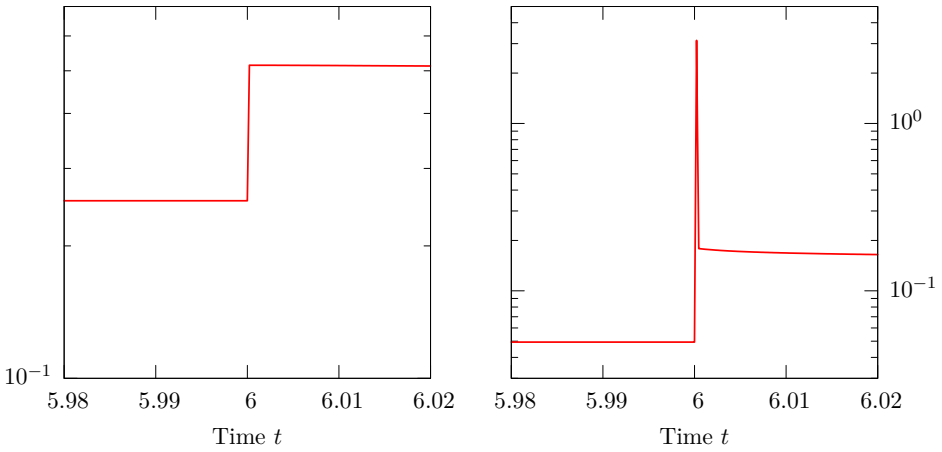


(a) Configuration I

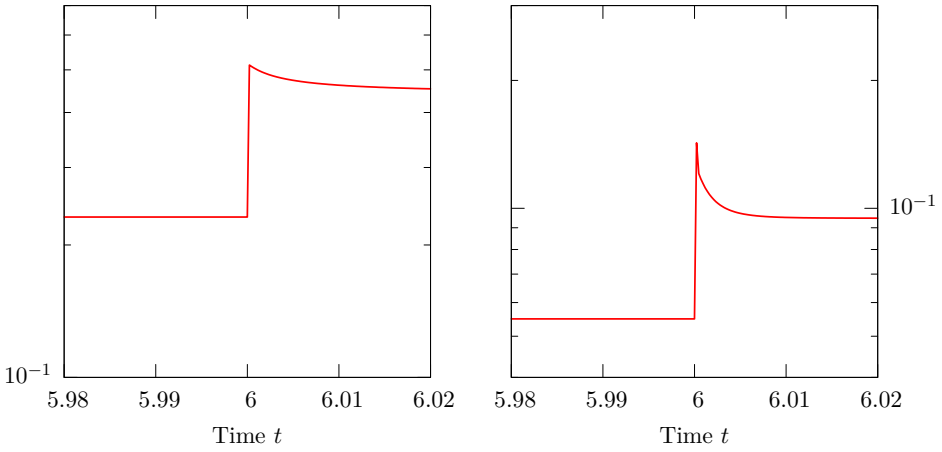


(b) Configuration II

Figure 5.5. Errors $\|\nabla(\boldsymbol{v} - \boldsymbol{v_{kh}})\|$ (left) and $\|p - p_{kh}\|$ (right)



(a) Configuration I



(b) Configuration II

Figure 5.6. Errors $\|\nabla(\boldsymbol{v} - \boldsymbol{v_{kh}})\|$ (left) and $\|p - p_{kh}\|$ (right) near $t = 6$

5.3 Behavior of the error under temporal and spatial refinement

In this section, we numerically analyze the behavior of the error described in the last section. We especially consider systematic uniform refinement of the temporal and spatial discretization. The analysis will be done by means of the Configurations I and II presented in the previous section. Thereby, we are going to consider the equal-order cG(1) and cG(2) discretizations in space together with the local projection stabilization as well as the inf-sup-stable $\mathcal{Q}_2/\mathcal{Q}_1$-Taylor-Hood element. For the temporal discretization we will apply the dG(0) and dG(1) method as well as the fractional-step-θ scheme which for convenience of the reader is briefly presented below.

Let the parameters

$$\theta = 1 - \frac{1}{2}\sqrt{2}, \quad \theta' = 1 - 2\theta, \quad \alpha = \frac{\theta'}{1-\theta}, \quad \beta = 1 - \alpha$$

be given. Then one fractional-step-θ time step $(\boldsymbol{V_{m-1}}, P_{m-1})^T \rightarrow (\boldsymbol{V_m}, P_m)^T$ for the incompressible Navier-Stokes equations consists of the following three sub-steps $t_{m-1} \rightarrow t_{m-1+\theta} \rightarrow t_{m-\theta} \rightarrow t_m$ (where the equations should be fulfilled for all $\boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in (H_h^m)^d \times L_h^m$):

$\boldsymbol{t_{m-1} \rightarrow t_{m-1+\theta}}$:

$$\begin{aligned}(\boldsymbol{V_{m-1+\theta}}, \boldsymbol{\psi}) + \alpha\theta k_m \Big\{\nu(\nabla \boldsymbol{V_{m-1+\theta}}, \nabla\boldsymbol{\psi}) + ((\boldsymbol{V_{m-1+\theta}} \cdot \nabla)\boldsymbol{V_{m-1+\theta}}, \boldsymbol{\psi})\Big\} \\ + \theta k_m (P_{m-1+\theta}, \nabla \cdot \boldsymbol{\psi}) = (\boldsymbol{V_{m-1}}, \boldsymbol{\psi}) - \beta\theta k_m \Big\{\nu(\nabla \boldsymbol{V_{m-1}}, \nabla\boldsymbol{\psi}) \\ + ((\boldsymbol{V_{m-1}} \cdot \nabla)\boldsymbol{V_{m-1}}, \boldsymbol{\psi})\Big\} + \theta k_m (\boldsymbol{f_{m-1}}, \boldsymbol{\psi}) \\ (\nabla \cdot \boldsymbol{V_{m-1+\theta}}, \chi) = 0\end{aligned}$$

$\boldsymbol{t_{m-1+\theta} \rightarrow t_{m-\theta}}$:

$$\begin{aligned}(\boldsymbol{V_{m-\theta}}, \boldsymbol{\psi}) + \beta\theta' k_m \Big\{\nu(\nabla \boldsymbol{V_{m-\theta}}, \nabla\boldsymbol{\psi}) + ((\boldsymbol{V_{m-\theta}} \cdot \nabla)\boldsymbol{V_{m-\theta}}, \boldsymbol{\psi})\Big\} \\ + \theta' k_m (P_{m-\theta}, \nabla \cdot \boldsymbol{\psi}) = (\boldsymbol{V_{m-1+\theta}}, \boldsymbol{\psi}) - \alpha\theta' k_m \Big\{\nu(\nabla \boldsymbol{V_{m-1+\theta}}, \nabla\boldsymbol{\psi}) \\ + ((\boldsymbol{V_{m-1+\theta}} \cdot \nabla)\boldsymbol{V_{m-1+\theta}}, \boldsymbol{\psi})\Big\} + \theta' k_m (\boldsymbol{f_{m-\theta}}, \boldsymbol{\psi}) \\ (\nabla \cdot \boldsymbol{V_{m-\theta}}, \chi) = 0\end{aligned}$$

$\boldsymbol{t_{m-\theta} \rightarrow t_m}$:

$$\begin{aligned}(\boldsymbol{V_m}, \boldsymbol{\psi}) + \alpha\theta k_m \Big\{\nu(\nabla \boldsymbol{V_m}, \nabla\boldsymbol{\psi}) + ((\boldsymbol{V_m} \cdot \nabla)\boldsymbol{V_m}, \boldsymbol{\psi})\Big\} \\ + \theta k_m (P_m, \nabla \cdot \boldsymbol{\psi}) = (\boldsymbol{V_{m-\theta}}, \boldsymbol{\psi}) - \beta\theta k_m \Big\{\nu(\nabla \boldsymbol{V_{m-\theta}}, \nabla\boldsymbol{\psi}) \\ + ((\boldsymbol{V_{m-\theta}} \cdot \nabla)\boldsymbol{V_{m-\theta}}, \boldsymbol{\psi})\Big\} + \theta k_m (\boldsymbol{f_{m-\theta}}, \boldsymbol{\psi}) \\ (\nabla \cdot \boldsymbol{V_m}, \chi) = 0\end{aligned}$$

For simplicity, we have omitted terms arising from stabilization.

In the following two sections, the behavior of the error in the pressure under systematic uniform refinement of the spatial or the temporal discretization is studied for different spatial and temporal discretizations. Since the error is concentrated to the first time step on a new mesh, we especially focus on its development there.

5.3.1 Spatial refinement

This section is dedicated to the numerical analysis of the error in the pressure under uniform refinement of the spatial discretization. To this end, we fix the temporal discretization which is either the dG(0), dG(1), or fractional-step-θ method. We always use a uniform time step size of $k = 3 \cdot 10^{-2}$. As in the previous section, we use the mesh $\mathcal{T}_{2h}$ on $I^{(1)}$ and $I^{(3)}$, whereas on $I^{(2)}$ the mesh $\mathcal{T}_h$ is used. We study the development of the error in the pressure component for $h \to 0$.

Let us first consider Configuration I. When using the dG(0) discretization in time, we obtain the results shown in Table 5.1. The corresponding results for the dG(1) time discretization are given in Table 5.2. Using the fractional-step-θ scheme for the temporal discretization leads to the results which are presented in Table 5.3. The orders of convergence given in the last lines are numerically computed from the values on the two finest discretizations.

Table 5.1. Configuration I: $\|p - p_{kh}\|$ under spatial refinement for the dG(0) time discretization with different spatial discretizations

Refinement	cG(1)		cG(2)		$\mathcal{Q}_2/\mathcal{Q}_1$	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$2.88 \cdot 10^{-01}$	$4.94 \cdot 10^{-01}$	$2.44 \cdot 10^{-02}$	$5.47 \cdot 10^{-02}$	$2.95 \cdot 10^{-02}$	$5.19 \cdot 10^{-02}$
2	$5.67 \cdot 10^{-02}$	$1.82 \cdot 10^{-01}$	$3.41 \cdot 10^{-03}$	$1.21 \cdot 10^{-02}$	$2.81 \cdot 10^{-03}$	$9.15 \cdot 10^{-03}$
3	$1.89 \cdot 10^{-02}$	$5.74 \cdot 10^{-02}$	$7.45 \cdot 10^{-04}$	$2.95 \cdot 10^{-03}$	$5.24 \cdot 10^{-04}$	$2.08 \cdot 10^{-03}$
4	$6.00 \cdot 10^{-03}$	$1.77 \cdot 10^{-02}$	$1.84 \cdot 10^{-04}$	$7.36 \cdot 10^{-04}$	$1.27 \cdot 10^{-04}$	$5.10 \cdot 10^{-04}$
order	1.66	1.70	2.02	2.00	2.04	2.03

Table 5.2. Configuration I: $\|p - p_{kh}\|$ under spatial refinement for the dG(1) time discretization with different spatial discretizations

Refinement	cG(1)		cG(2)		$\mathcal{Q}_2/\mathcal{Q}_1$	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$4.51 \cdot 10^{-01}$	$4.78 \cdot 10^{-01}$	$4.34 \cdot 10^{-02}$	$4.53 \cdot 10^{-02}$	$5.41 \cdot 10^{-02}$	$3.85 \cdot 10^{-02}$
2	$3.96 \cdot 10^{-02}$	$1.31 \cdot 10^{-01}$	$4.53 \cdot 10^{-03}$	$1.19 \cdot 10^{-02}$	$4.28 \cdot 10^{-03}$	$8.74 \cdot 10^{-03}$
3	$1.51 \cdot 10^{-02}$	$4.87 \cdot 10^{-02}$	$7.75 \cdot 10^{-04}$	$2.95 \cdot 10^{-03}$	$5.68 \cdot 10^{-04}$	$2.07 \cdot 10^{-03}$
4	$5.81 \cdot 10^{-03}$	$1.73 \cdot 10^{-02}$	$1.85 \cdot 10^{-04}$	$7.36 \cdot 10^{-04}$	$1.28 \cdot 10^{-04}$	$5.10 \cdot 10^{-04}$
order	1.38	1.49	2.07	2.00	2.15	2.02

Table 5.3. Configuration I: $\|p - p_{kh}\|$ under spatial refinement for the fractional-step-θ time discretization with different spatial discretizations

Refinement	cG(1)		cG(2)		$\mathcal{Q}_2/\mathcal{Q}_1$	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$8.00 \cdot 10^{-02}$	$3.45 \cdot 10^{-01}$	$1.04 \cdot 10^{-02}$	$4.43 \cdot 10^{-02}$	$9.27 \cdot 10^{-03}$	$4.46 \cdot 10^{-02}$
2	$3.27 \cdot 10^{-02}$	$1.40 \cdot 10^{-01}$	$2.51 \cdot 10^{-03}$	$9.76 \cdot 10^{-03}$	$2.11 \cdot 10^{-03}$	$8.72 \cdot 10^{-03}$
3	$1.34 \cdot 10^{-02}$	$4.24 \cdot 10^{-02}$	$6.25 \cdot 10^{-04}$	$2.39 \cdot 10^{-03}$	$5.33 \cdot 10^{-04}$	$2.07 \cdot 10^{-03}$
4	$5.15 \cdot 10^{-03}$	$1.40 \cdot 10^{-02}$	$1.50 \cdot 10^{-04}$	$5.96 \cdot 10^{-04}$	$1.28 \cdot 10^{-04}$	$5.10 \cdot 10^{-04}$
order	1.38	1.60	2.06	2.00	2.06	2.02

If we perform the same tests with Configuration II, we obtain for the dG(0), dG(1), and fractional-step-θ time discretization the results shown in Tables 5.4–5.6, respectively.

We can conclude that the error in the pressure component in the first time step on a new spatial mesh converges for $h \to 0$ (at least) with the same order as the overall spatial discretization error.

Table 5.4. Configuration II: $\|p - p_{kh}\|$ under spatial refinement for the dG(0) time discretization with different spatial discretizations

Refinement	cG(1)		cG(2)		$\mathcal{Q}_2/\mathcal{Q}_1$	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$1.48 \cdot 10^{-01}$	$2.81 \cdot 10^{-01}$	$4.72 \cdot 10^{-02}$	$1.04 \cdot 10^{-01}$	$4.59 \cdot 10^{-02}$	$1.12 \cdot 10^{-01}$
2	$5.53 \cdot 10^{-02}$	$9.85 \cdot 10^{-02}$	$1.17 \cdot 10^{-02}$	$4.35 \cdot 10^{-02}$	$1.16 \cdot 10^{-02}$	$4.48 \cdot 10^{-02}$
3	$1.56 \cdot 10^{-02}$	$3.97 \cdot 10^{-02}$	$2.89 \cdot 10^{-03}$	$1.16 \cdot 10^{-02}$	$2.89 \cdot 10^{-03}$	$1.15 \cdot 10^{-02}$
4	$4.03 \cdot 10^{-03}$	$1.14 \cdot 10^{-02}$	$7.21 \cdot 10^{-04}$	$2.90 \cdot 10^{-03}$	$7.20 \cdot 10^{-04}$	$2.88 \cdot 10^{-03}$
order	1.95	1.80	2.00	2.00	2.01	2.00

Table 5.5. Configuration II: $\|p - p_{kh}\|$ under spatial refinement for the dG(1) time discretization with different spatial discretizations

Refinement	cG(1)		cG(2)		$\mathcal{Q}_2/\mathcal{Q}_1$	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$1.53 \cdot 10^{-01}$	$2.73 \cdot 10^{-01}$	$4.87 \cdot 10^{-02}$	$1.00 \cdot 10^{-01}$	$4.79 \cdot 10^{-02}$	$1.48 \cdot 10^{-01}$
2	$5.56 \cdot 10^{-02}$	$9.57 \cdot 10^{-02}$	$1.18 \cdot 10^{-02}$	$4.31 \cdot 10^{-02}$	$1.16 \cdot 10^{-02}$	$4.42 \cdot 10^{-02}$
3	$1.57 \cdot 10^{-02}$	$3.92 \cdot 10^{-02}$	$2.89 \cdot 10^{-03}$	$1.16 \cdot 10^{-02}$	$2.87 \cdot 10^{-03}$	$1.13 \cdot 10^{-02}$
4	$4.04 \cdot 10^{-03}$	$1.14 \cdot 10^{-02}$	$7.21 \cdot 10^{-04}$	$2.90 \cdot 10^{-03}$	$7.19 \cdot 10^{-04}$	$2.86 \cdot 10^{-03}$
order	1.96	1.78	2.00	2.00	2.00	1.98

Table 5.6. Configuration II: $\|p - p_{kh}\|$ under spatial refinement for the fractional-step-θ time discretization with different spatial discretizations

Refinement	cG(1)		cG(2)		$\mathcal{Q}_2/\mathcal{Q}_1$	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$1.47 \cdot 10^{-01}$	$2.63 \cdot 10^{-01}$	$4.70 \cdot 10^{-02}$	$1.05 \cdot 10^{-01}$	$4.46 \cdot 10^{-02}$	$1.10 \cdot 10^{-01}$
2	$5.60 \cdot 10^{-02}$	$9.83 \cdot 10^{-02}$	$1.19 \cdot 10^{-02}$	$4.46 \cdot 10^{-02}$	$1.15 \cdot 10^{-02}$	$4.46 \cdot 10^{-02}$
3	$1.57 \cdot 10^{-02}$	$4.15 \cdot 10^{-02}$	$2.92 \cdot 10^{-03}$	$1.16 \cdot 10^{-02}$	$2.89 \cdot 10^{-03}$	$1.15 \cdot 10^{-02}$
4	$4.10 \cdot 10^{-03}$	$1.15 \cdot 10^{-02}$	$7.23 \cdot 10^{-04}$	$2.90 \cdot 10^{-03}$	$7.20 \cdot 10^{-04}$	$2.88 \cdot 10^{-03}$
order	1.94	1.85	2.01	2.00	2.00	2.00

5.3.2 Temporal refinement

In this section, the development of the pressure error under systematic uniform refinement of the temporal discretization is considered, that is we consider the case $k \to 0$. To this end, we fix the spatial discretization which is either the equal-order cG(1) or cG(2) method together with local projection stabilization or the inf-sup-stable $\mathcal{Q}_2/\mathcal{Q}_1$-Taylor-Hood element. The spatial meshes $\mathcal{T}_h$ used on the subintervals $I^{(2)}$ are the ones depicted in Figure 5.4. On the subintervals $I^{(1)}$ and $I^{(3)}$ the corresponding meshes $\mathcal{T}_{2h}$ are employed.

Firstly, we consider again Configuration I. For the cG(1) discretization in space, we obtain the results of Table 5.7. The corresponding results for the cG(2) discretization are given in Table 5.8, whereas Table 5.9 shows the results obtained with the inf-sup-stable $\mathcal{Q}_2/\mathcal{Q}_1$-Taylor-Hood element. As in the previous section, the order of convergence is numerically computed from the values of the two finest discretizations.

Table 5.7. Configuration I: $\|p - p_{kh}\|$ under temporal refinement for the cG(1) discretization with different temporal discretizations

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$3.12 \cdot 10^{-01}$	$5.16 \cdot 10^{-01}$	$4.75 \cdot 10^{-01}$	$6.39 \cdot 10^{-01}$	$1.04 \cdot 10^{-01}$	$1.48 \cdot 10^{-01}$
2	$5.52 \cdot 10^{-01}$	$8.86 \cdot 10^{-01}$	$9.83 \cdot 10^{-01}$	$1.39 \cdot 10^{+00}$	$1.75 \cdot 10^{-01}$	$1.49 \cdot 10^{-01}$
3	$1.02 \cdot 10^{+00}$	$1.63 \cdot 10^{+00}$	$1.97 \cdot 10^{+00}$	$2.89 \cdot 10^{+00}$	$2.33 \cdot 10^{-01}$	$1.49 \cdot 10^{-01}$
4	$1.96 \cdot 10^{+00}$	$3.14 \cdot 10^{+00}$	$3.90 \cdot 10^{+00}$	$5.90 \cdot 10^{+00}$	$2.71 \cdot 10^{-01}$	$1.49 \cdot 10^{-01}$
order	-0.94	-0.95	-0.99	-1.03	-0.22	0.00

Performing the same tests with Configuration II, leads to similar results which are presented in Tables 5.10–5.12 for the cG(1), cG(2), and $\mathcal{Q}_2/\mathcal{Q}_1$ discretization in space, respectively.

We observe that under uniform refinement of the temporal discretization the error in the pressure component when uniformly refining the spatial mesh increases like $O(k^{-1})$ for the dG(0) and the dG(1) discretization whereas for the fractional-step-θ scheme this error is almost independent of k. For uniform coarsening of the spatial mesh we observe the same

Table 5.8. Configuration I: $\|p - p_{kh}\|$ under temporal refinement for the cG(2) discretization with different temporal discretizations

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$2.56 \cdot 10^{-02}$	$2.38 \cdot 10^{-02}$	$4.31 \cdot 10^{-02}$	$3.15 \cdot 10^{-02}$	$5.41 \cdot 10^{-03}$	$1.07 \cdot 10^{-02}$
2	$4.87 \cdot 10^{-02}$	$4.00 \cdot 10^{-02}$	$8.66 \cdot 10^{-02}$	$6.12 \cdot 10^{-02}$	$8.44 \cdot 10^{-03}$	$1.16 \cdot 10^{-02}$
3	$9.38 \cdot 10^{-02}$	$7.28 \cdot 10^{-02}$	$1.75 \cdot 10^{-01}$	$1.24 \cdot 10^{-01}$	$1.05 \cdot 10^{-02}$	$1.24 \cdot 10^{-02}$
4	$1.83 \cdot 10^{-01}$	$1.38 \cdot 10^{-01}$	$3.54 \cdot 10^{-01}$	$2.54 \cdot 10^{-01}$	$1.16 \cdot 10^{-02}$	$1.29 \cdot 10^{-02}$
order	-0.96	-0.92	-1.02	-1.03	-0.14	-0.06

Table 5.9. Configuration I: $\|p - p_{kh}\|$ under temporal refinement for the $\mathcal{Q}_2/\mathcal{Q}_1$ discretization with different temporal discretizations

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$2.80 \cdot 10^{-02}$	$2.35 \cdot 10^{-02}$	$4.70 \cdot 10^{-02}$	$2.97 \cdot 10^{-02}$	$5.26 \cdot 10^{-03}$	$1.00 \cdot 10^{-02}$
2	$5.35 \cdot 10^{-02}$	$4.05 \cdot 10^{-02}$	$9.42 \cdot 10^{-02}$	$5.94 \cdot 10^{-02}$	$8.33 \cdot 10^{-03}$	$1.12 \cdot 10^{-02}$
3	$1.03 \cdot 10^{-01}$	$7.38 \cdot 10^{-02}$	$1.91 \cdot 10^{-01}$	$1.23 \cdot 10^{-01}$	$1.04 \cdot 10^{-02}$	$1.22 \cdot 10^{-02}$
4	$2.01 \cdot 10^{-01}$	$1.40 \cdot 10^{-01}$	$3.85 \cdot 10^{-01}$	$2.51 \cdot 10^{-01}$	$1.15 \cdot 10^{-02}$	$1.28 \cdot 10^{-02}$
order	-0.96	-0.92	-1.01	-1.03	-0.15	-0.07

Table 5.10. Configuration II: $\|p - p_{kh}\|$ under temporal refinement for the cG(1) discretization with different temporal discretizations

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$9.43 \cdot 10^{-02}$	$1.11 \cdot 10^{-01}$	$1.36 \cdot 10^{-01}$	$1.06 \cdot 10^{-01}$	$6.19 \cdot 10^{-02}$	$1.09 \cdot 10^{-01}$
2	$1.52 \cdot 10^{-01}$	$1.18 \cdot 10^{-01}$	$2.51 \cdot 10^{-01}$	$1.17 \cdot 10^{-01}$	$7.04 \cdot 10^{-02}$	$1.16 \cdot 10^{-01}$
3	$2.73 \cdot 10^{-01}$	$1.26 \cdot 10^{-01}$	$4.90 \cdot 10^{-01}$	$1.37 \cdot 10^{-01}$	$8.16 \cdot 10^{-02}$	$1.20 \cdot 10^{-01}$
4	$5.17 \cdot 10^{-01}$	$1.43 \cdot 10^{-01}$	$9.75 \cdot 10^{-01}$	$1.86 \cdot 10^{-01}$	$9.20 \cdot 10^{-02}$	$1.23 \cdot 10^{-01}$
order	-0.93	-0.18	-0.99	-0.44	-0.17	-0.04

Table 5.11. Configuration II: $\|p - p_{kh}\|$ under temporal refinement for the cG(2) discretization with different temporal discretizations

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$2.15 \cdot 10^{-02}$	$4.55 \cdot 10^{-02}$	$3.29 \cdot 10^{-02}$	$4.41 \cdot 10^{-01}$	$1.22 \cdot 10^{-02}$	$4.55 \cdot 10^{-02}$
2	$3.63 \cdot 10^{-02}$	$4.68 \cdot 10^{-02}$	$6.07 \cdot 10^{-02}$	$4.61 \cdot 10^{-02}$	$1.34 \cdot 10^{-02}$	$4.68 \cdot 10^{-02}$
3	$6.68 \cdot 10^{-02}$	$4.86 \cdot 10^{-02}$	$1.17 \cdot 10^{-01}$	$4.96 \cdot 10^{-02}$	$1.59 \cdot 10^{-02}$	$4.85 \cdot 10^{-02}$
4	$1.27 \cdot 10^{-01}$	$5.08 \cdot 10^{-02}$	$2.32 \cdot 10^{-01}$	$5.61 \cdot 10^{-02}$	$1.91 \cdot 10^{-02}$	$4.99 \cdot 10^{-02}$
order	-0.93	-0.06	-0.99	-0.18	-0.26	-0.04

Table 5.12. Configuration II: $\|p - p_{kh}\|$ under temporal refinement for the $\mathcal{Q}_2/\mathcal{Q}_1$ discretization with different temporal discretizations

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$2.29 \cdot 10^{-02}$	$4.52 \cdot 10^{-02}$	$3.72 \cdot 10^{-02}$	$4.47 \cdot 10^{-02}$	$1.18 \cdot 10^{-02}$	$4.47 \cdot 10^{-02}$
2	$4.02 \cdot 10^{-02}$	$4.64 \cdot 10^{-02}$	$7.02 \cdot 10^{-02}$	$4.60 \cdot 10^{-02}$	$1.22 \cdot 10^{-02}$	$4.50 \cdot 10^{-02}$
3	$7.62 \cdot 10^{-02}$	$5.02 \cdot 10^{-02}$	$1.38 \cdot 10^{-01}$	$5.13 \cdot 10^{-02}$	$1.29 \cdot 10^{-02}$	$4.59 \cdot 10^{-02}$
4	$1.48 \cdot 10^{-01}$	$6.06 \cdot 10^{-02}$	$2.76 \cdot 10^{-01}$	$7.01 \cdot 10^{-02}$	$1.42 \cdot 10^{-02}$	$4.69 \cdot 10^{-02}$
order	-0.96	-0.27	-1.00	-0.45	-0.14	-0.03

behavior in Configuration I, whereas in Configuration II the occurring error seems to be bounded.

Since the support of this error is exactly one time step, this shows the behavior of a Dirac approximation and hence the error, for example, in mean functional values involving the pressure does not vanish for $k \to 0$.

5.4 Attempts to solve this problem

We have seen in the previous sections that the error in the pressure occurring when switching the spatial mesh decreases with (at least) the same order as the spatial discretization error for $h \to 0$, but increases like $O(k^{-1})$ for $k \to 0$. We showed that this effect does not originate from the stabilization since the inf-sup-stable Taylor-Hood element also produces qualitatively the same error. In this section, we discuss some attempts to overcome this problem and obtain pressure approximations which remain bounded for $k \to 0$.

Since the approximation of the velocity component is quite satisfactory and only the pressure approximation deteriorates and since the error is solely located in the first time step on the new spatial mesh, one is lead to the assumption that the problem arises from the fact that the velocity field from the last time step is not divergence-free with respect to the test functions of the new finite element space. We present three "ideas" that might be able to deal with this phenomenon:

Repeating one time step: After computing $(\boldsymbol{V_{m-1}}, P_{m-1})^T \in (H_h^{m-1})^d \times L_h^{m-1}$ repeat the current time step to determine approximations $(\widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}, \widetilde{P}_{m-1})^T \in (H_h^m)^d \times L_h^m$ for $t = t_{m-1}$, but already in the finite element spaces corresponding to $t = t_m$. One can hope that then only $\widetilde{P}_{m-1}$ contains this error and since $\widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}$ is divergence-free with respect to test functions in $(H_h^m)^d$ this error probably does not occur again when computing $(\boldsymbol{V_m}, P_m)^T \in (H_h^m)^d \times L_h^m$ using the initial values $\widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}$.

Divergence-free $\boldsymbol{L^2}$-projection ($\boldsymbol{H}$-projection): After computing $(\boldsymbol{V_{m-1}}, P_{m-1})^T \in (H_h^{m-1})^d \times L_h^{m-1}$ first compute a projection $\widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}$ of $\boldsymbol{V_{m-1}}$ into $(H_h^m)^d$ which is

divergence-free with respect to test functions in $(H_h^m)^d$ and use this projection as initial values for the next time step. The projection is determined by

$$\begin{aligned} (\widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}, \boldsymbol{\psi}) - (\widetilde{P}, \nabla \cdot \boldsymbol{\psi}) &= (\boldsymbol{V}_{\boldsymbol{m-1}}, \boldsymbol{\psi}) \qquad \forall \boldsymbol{\psi} \in (H_h^m)^d, \\ (\nabla \cdot \widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}, \chi) &= 0 \qquad \forall \chi \in L_h^m. \end{aligned} \tag{5.2}$$

Divergence-free $\boldsymbol{H_0^1}$-projection ($\boldsymbol{V}$-projection): Same procedure as for "Divergence-free L^2-projection", but this time the projection is determined by

$$\begin{aligned} (\nabla\widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}, \nabla\boldsymbol{\psi}) - (\widetilde{P}, \nabla \cdot \boldsymbol{\psi}) &= (\nabla\boldsymbol{V}_{\boldsymbol{m-1}}, \nabla\boldsymbol{\psi}) \qquad \forall \boldsymbol{\psi} \in (H_h^m)^d, \\ (\nabla \cdot \widetilde{\boldsymbol{V}}_{\boldsymbol{m-1}}, \chi) &= 0 \qquad \forall \chi \in L_h^m. \end{aligned} \tag{5.3}$$

In the sequel, we denote the divergence-free L^2-projection by H-projection and the divergence-free H_0^1-projection by V-projection. For the definition of the spaces H and V see (2.1). For the equal-order spatial discretizations cG(1) and cG(2) the variational formulations given above have to be stabilized, of course. This is also done by means of the local projection stabilization.

Since the behavior of the pressure error already is of the right order for $h \to 0$, we discuss in this section only the influence of the presented "ideas" on the development of the error under uniform temporal refinement. To this end, we consider only Configuration I because the results for Configuration II are similar to those of Configuration I and perform the same numerical analysis as in Section 5.3.2. The spatial meshes used are again the ones depicted in Figure 5.4.

5.4.1 Repeating one time step

In this section, we present the development of the pressure error when repeating the last time step of the old spatial mesh on the new one to determine the initial values for the first real time step on the new mesh.

The results under uniform temporal refinement are listed in Tables 5.13–5.15 for the cG(1), cG(2), and $\mathcal{Q}_2/\mathcal{Q}_1$ spatial discretization, respectively.

We observe for all spatial discretizations that repeating one time step leads to a slower increase of the error in the pressure when switching the spatial mesh from $\mathcal{T}_{2h}$ to $\mathcal{T}_h$ and to an almost constant error when switching from $\mathcal{T}_h$ to $\mathcal{T}_{2h}$.

5.4.2 H-projection

In this section, we present the development of the pressure error when using the H-projection of the old velocity field into the new finite element space as initial values when switching the spatial mesh.

Table 5.13. Configuration I: $\|p - p_{kh}\|$ for the cG(1) discretization with different temporal discretizations and repeating one time step

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$8.57 \cdot 10^{-02}$	$7.47 \cdot 10^{-01}$	$4.66 \cdot 10^{-02}$	$7.48 \cdot 10^{-01}$	$4.64 \cdot 10^{-02}$	$1.47 \cdot 10^{-01}$
2	$1.52 \cdot 10^{-01}$	$7.50 \cdot 10^{-01}$	$1.01 \cdot 10^{-01}$	$7.51 \cdot 10^{-01}$	$9.90 \cdot 10^{-02}$	$1.48 \cdot 10^{-01}$
3	$2.45 \cdot 10^{-01}$	$7.52 \cdot 10^{-01}$	$2.11 \cdot 10^{-01}$	$7.52 \cdot 10^{-01}$	$1.71 \cdot 10^{-01}$	$1.49 \cdot 10^{-01}$
4	$3.36 \cdot 10^{-01}$	$7.52 \cdot 10^{-01}$	$3.21 \cdot 10^{-01}$	$7.52 \cdot 10^{-01}$	$2.31 \cdot 10^{-01}$	$1.49 \cdot 10^{-01}$
order	-0.46	0.00	-0.61	0.00	-0.43	0.00

Table 5.14. Configuration I: $\|p - p_{kh}\|$ for the cG(2) discretization with different temporal discretizations and repeating one time step

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$4.14 \cdot 10^{-03}$	$4.41 \cdot 10^{-01}$	$3.26 \cdot 10^{-03}$	$4.46 \cdot 10^{-01}$	$2.95 \cdot 10^{-03}$	$1.01 \cdot 10^{-02}$
2	$6.06 \cdot 10^{-03}$	$4.36 \cdot 10^{-01}$	$5.09 \cdot 10^{-03}$	$4.39 \cdot 10^{-01}$	$5.14 \cdot 10^{-03}$	$1.05 \cdot 10^{-02}$
3	$8.40 \cdot 10^{-03}$	$4.31 \cdot 10^{-01}$	$7.96 \cdot 10^{-03}$	$4.33 \cdot 10^{-01}$	$8.11 \cdot 10^{-03}$	$1.14 \cdot 10^{-02}$
4	$1.04 \cdot 10^{-02}$	$4.27 \cdot 10^{-01}$	$1.03 \cdot 10^{-02}$	$4.28 \cdot 10^{-01}$	$1.03 \cdot 10^{-02}$	$1.22 \cdot 10^{-02}$
order	-0.31	0.01	-0.37	0.02	-0.34	-0.10

Table 5.15. Configuration I: $\|p - p_{kh}\|$ for the $\mathcal{Q}_2/\mathcal{Q}_1$ discretization with different temporal discretizations and repeating one time step

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$3.61 \cdot 10^{-03}$	$9.33 \cdot 10^{-03}$	$2.53 \cdot 10^{-03}$	$9.01 \cdot 10^{-03}$	$4.10 \cdot 10^{-03}$	$9.60 \cdot 10^{-03}$
2	$5.74 \cdot 10^{-03}$	$1.00 \cdot 10^{-02}$	$4.70 \cdot 10^{-03}$	$9.66 \cdot 10^{-03}$	$7.12 \cdot 10^{-03}$	$1.07 \cdot 10^{-02}$
3	$8.16 \cdot 10^{-03}$	$1.10 \cdot 10^{-02}$	$7.72 \cdot 10^{-03}$	$1.08 \cdot 10^{-02}$	$9.66 \cdot 10^{-03}$	$1.18 \cdot 10^{-02}$
4	$1.01 \cdot 10^{-02}$	$1.20 \cdot 10^{-02}$	$1.00 \cdot 10^{-02}$	$1.19 \cdot 10^{-02}$	$1.11 \cdot 10^{-02}$	$1.26 \cdot 10^{-02}$
order	-0.31	-0.13	-0.37	-0.14	-0.20	-0.09

Table 5.16. Configuration I: $\|p - p_{kh}\|$ for the cG(1) discretization with different temporal discretizations and H-projection

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$3.47 \cdot 10^{-02}$	$1.27 \cdot 10^{-01}$	$3.77 \cdot 10^{-02}$	$1.28 \cdot 10^{-01}$	$3.08 \cdot 10^{-02}$	$1.25 \cdot 10^{-01}$
2	$3.42 \cdot 10^{-02}$	$1.25 \cdot 10^{-01}$	$3.51 \cdot 10^{-02}$	$1.25 \cdot 10^{-01}$	$3.12 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$
3	$3.48 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$	$3.49 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$	$3.28 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$
4	$3.57 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$	$3.57 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$	$3.42 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$
order	-0.04	0.00	-0.03	0.00	-0.06	0.00

Table 5.17. Configuration I: $\|p - p_{kh}\|$ for the cG(2) discretization with different temporal discretizations and H-projection

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$5.73 \cdot 10^{-03}$	$1.48 \cdot 10^{-02}$	$5.54 \cdot 10^{-03}$	$1.30 \cdot 10^{-02}$	$5.29 \cdot 10^{-03}$	$1.27 \cdot 10^{-02}$
2	$6.97 \cdot 10^{-03}$	$1.73 \cdot 10^{-02}$	$7.82 \cdot 10^{-03}$	$1.59 \cdot 10^{-02}$	$7.81 \cdot 10^{-03}$	$1.64 \cdot 10^{-02}$
3	$7.67 \cdot 10^{-03}$	$1.99 \cdot 10^{-02}$	$8.72 \cdot 10^{-03}$	$1.93 \cdot 10^{-02}$	$8.64 \cdot 10^{-03}$	$1.98 \cdot 10^{-02}$
4	$7.69 \cdot 10^{-03}$	$2.19 \cdot 10^{-02}$	$8.36 \cdot 10^{-03}$	$2.17 \cdot 10^{-02}$	$8.24 \cdot 10^{-03}$	$2.20 \cdot 10^{-02}$
order	0.00	−0.14	0.06	−0.17	0.07	−0.15

Table 5.18. Configuration I: $\|p - p_{kh}\|$ for the $\mathcal{Q}_2/\mathcal{Q}_1$ discretization with different temporal discretizations and H-projection

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$6.44 \cdot 10^{-03}$	$1.03 \cdot 10^{-02}$	$4.74 \cdot 10^{-03}$	$9.63 \cdot 10^{-03}$	$5.23 \cdot 10^{-03}$	$9.83 \cdot 10^{-03}$
2	$8.48 \cdot 10^{-03}$	$1.12 \cdot 10^{-02}$	$7.79 \cdot 10^{-03}$	$1.08 \cdot 10^{-02}$	$8.27 \cdot 10^{-03}$	$1.10 \cdot 10^{-02}$
3	$1.02 \cdot 10^{-02}$	$1.21 \cdot 10^{-02}$	$1.01 \cdot 10^{-02}$	$1.19 \cdot 10^{-02}$	$1.04 \cdot 10^{-02}$	$1.20 \cdot 10^{-02}$
4	$1.13 \cdot 10^{-02}$	$1.27 \cdot 10^{-02}$	$1.13 \cdot 10^{-02}$	$1.26 \cdot 10^{-02}$	$1.14 \cdot 10^{-02}$	$1.27 \cdot 10^{-02}$
order	−0.15	−0.07	−0.16	−0.08	−0.13	−0.08

The results under uniform temporal refinement are listed in Tables 5.16–5.18 for the cG(1), cG(2), and $\mathcal{Q}_2/\mathcal{Q}_1$ spatial discretization, respectively.

We can conclude that using the H-projection of the velocity of the last time step into the new finite element space as initial values leads to pressure errors which are bounded for $k \to 0$. Actually, the pressure error becomes almost independent of k.

5.4.3 V-projection

In this section, we present the development of the pressure error when using the V-projection of the old velocity field into the new finite element space as initial values when switching the spatial mesh.

The results under uniform temporal refinement are listed in Tables 5.19–5.21 for the cG(1), cG(2), and $\mathcal{Q}_2/\mathcal{Q}_1$ spatial discretization, respectively.

Using the V-projection of the old velocity into the new finite element space also leads to pressure errors which remain bounded for $k \to 0$.

We have seen that all three "ideas" are able to (almost) remove the $O(k^{-1})$ increase in the pressure error while the H-projection performs best compared to the other strategies. This also substantiates the fact that the fractional-step-θ scheme leads to bounded pressure

Table 5.19. Configuration I: $\|p - p_{kh}\|$ for the cG(1) discretization with different temporal discretizations and V-projection

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$1.18 \cdot 10^{-01}$	$1.47 \cdot 10^{-01}$	$7.23 \cdot 10^{-02}$	$1.47 \cdot 10^{-01}$	$8.35 \cdot 10^{-02}$	$1.24 \cdot 10^{-01}$
2	$1.59 \cdot 10^{-01}$	$1.47 \cdot 10^{-01}$	$1.32 \cdot 10^{-01}$	$1.47 \cdot 10^{-01}$	$1.33 \cdot 10^{-01}$	$1.24 \cdot 10^{-01}$
3	$1.99 \cdot 10^{-01}$	$1.47 \cdot 10^{-01}$	$1.87 \cdot 10^{-01}$	$1.47 \cdot 10^{-01}$	$1.73 \cdot 10^{-01}$	$1.24 \cdot 10^{-01}$
4	$2.28 \cdot 10^{-01}$	$1.47 \cdot 10^{-01}$	$2.24 \cdot 10^{-01}$	$1.47 \cdot 10^{-01}$	$1.99 \cdot 10^{-01}$	$1.24 \cdot 10^{-01}$
order	-0.20	0.00	-0.26	0.00	-0.20	0.00

Table 5.20. Configuration I: $\|p - p_{kh}\|$ for the cG(2) discretization with different temporal discretizations and V-projection

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$6.08 \cdot 10^{-03}$	$1.20 \cdot 10^{-02}$	$4.86 \cdot 10^{-03}$	$1.19 \cdot 10^{-02}$	$4.90 \cdot 10^{-03}$	$9.98 \cdot 10^{-03}$
2	$7.84 \cdot 10^{-03}$	$1.21 \cdot 10^{-02}$	$7.39 \cdot 10^{-03}$	$1.21 \cdot 10^{-02}$	$7.65 \cdot 10^{-03}$	$1.01 \cdot 10^{-02}$
3	$9.34 \cdot 10^{-03}$	$1.22 \cdot 10^{-02}$	$9.33 \cdot 10^{-03}$	$1.22 \cdot 10^{-02}$	$9.49 \cdot 10^{-03}$	$1.02 \cdot 10^{-02}$
4	$1.03 \cdot 10^{-02}$	$1.23 \cdot 10^{-02}$	$1.04 \cdot 10^{-02}$	$1.23 \cdot 10^{-02}$	$1.04 \cdot 10^{-02}$	$1.03 \cdot 10^{-02}$
order	-0.14	-0.01	-0.16	-0.01	-0.13	-0.01

Table 5.21. Configuration I: $\|p - p_{kh}\|$ for the $\mathcal{Q}_2/\mathcal{Q}_1$ discretization with different temporal discretizations and V-projection

Refinement	dG(0)		dG(1)		fractional-step-θ	
level	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$	$t = 3 + k$	$t = 6 + k$
1	$5.78 \cdot 10^{-03}$	$8.83 \cdot 10^{-03}$	$4.34 \cdot 10^{-03}$	$8.81 \cdot 10^{-03}$	$4.74 \cdot 10^{-03}$	$8.82 \cdot 10^{-03}$
2	$7.61 \cdot 10^{-03}$	$8.87 \cdot 10^{-03}$	$7.11 \cdot 10^{-03}$	$8.86 \cdot 10^{-03}$	$7.53 \cdot 10^{-03}$	$8.87 \cdot 10^{-03}$
3	$9.10 \cdot 10^{-03}$	$8.92 \cdot 10^{-03}$	$9.11 \cdot 10^{-03}$	$8.91 \cdot 10^{-03}$	$9.37 \cdot 10^{-03}$	$8.92 \cdot 10^{-03}$
4	$1.00 \cdot 10^{-02}$	$8.95 \cdot 10^{-03}$	$1.01 \cdot 10^{-02}$	$8.95 \cdot 10^{-03}$	$1.02 \cdot 10^{-02}$	$8.95 \cdot 10^{-03}$
order	-0.14	0.00	-0.15	-0.01	-0.12	0.00

errors for $k \to 0$ without any additional effort. Here, the pressure error is hidden in the first sub-step and hence cannot be seen.

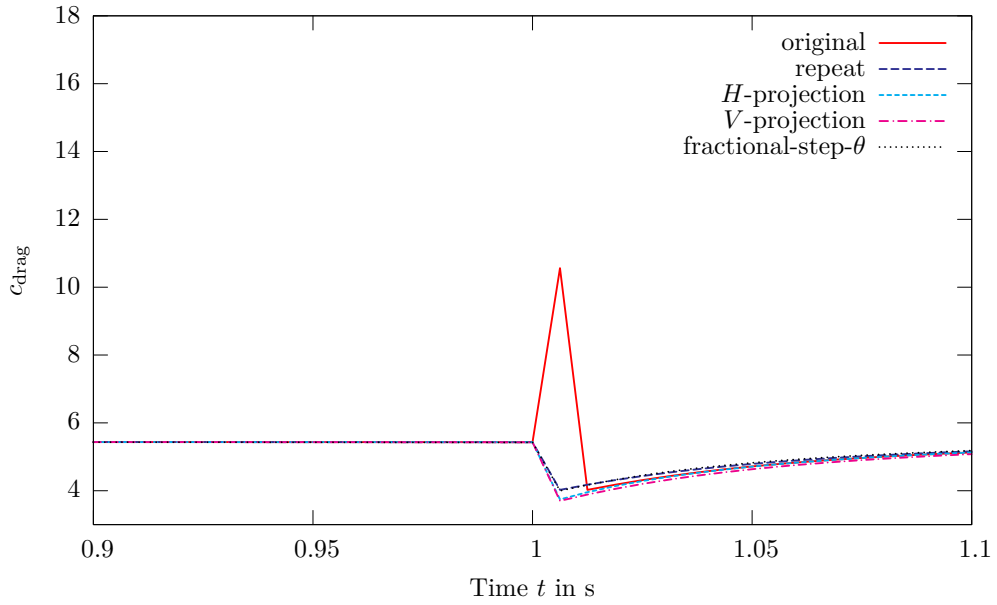
In the remaining part of this section, we want to compare all strategies. To this end, we return to the benchmark configuration "Laminar Flow Around a Cylinder" with constant inflow and Reynolds number Re = 20 which possesses a stationary solution. The discretization used here is again the cG(1)dG(0) method involving local projection stabilization. In Figure 5.7, the temporal evolution of the drag-coefficient is depicted for different choices of the initial value. For completeness, we also show the results of the fractional-step-θ scheme combined with a cG(1) discretization in space and local projection stabilization. The upper picture shows the development when switching the spatial mesh from $\mathcal{T}_{2h}$ to $\mathcal{T}_h$ which corresponds to a uniform refinement, while the lower picture shows the drag-coefficient when switching from $\mathcal{T}_h$ to $\mathcal{T}_{2h}$ which corresponds to a uniform coarsening. The labeling of the different curves is as follows:

- "original": No additional operations are performed when switching the spatial mesh.
- "repeat": When switching the spatial mesh, the last time step is repeated already on the new mesh to obtain initial values.
- "H-projection": When switching the spatial mesh, the H-projection of the old velocity into the new finite element space is used as initial values for the new time step.
- "V-projection": When switching the spatial mesh, the V-projection of the old velocity into the new finite element space is used as initial values for the new time step.

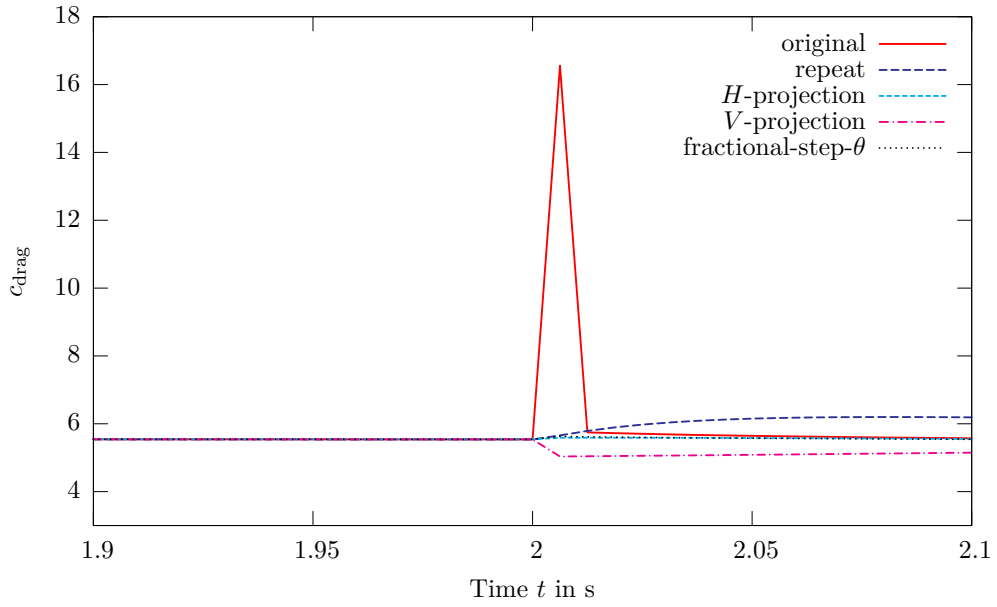
When looking at the upper picture of Figure 5.7, we note the large error in the drag-coefficient for the "original" method. Despite the first time step on the new mesh, the curves of the "H-projection" and "V-projection" are relatively close to each other while the "H-projection" is closest to the "original" curve. Repeating one time step leads to a slightly different temporal evolution of the drag-coefficient which is close to the values produced by the fractional-step-θ scheme.

If we consider the lower picture of Figure 5.7 which shows the temporal evolution of the drag-coefficient under a uniform coarsening of the spatial mesh, we observe quite large differences between the different strategies. While the "H-projection" mainly eliminates the large error of the "original" curve and stays very close to it elsewhere, the other "ideas" lead to completely different temporal evolutions of the drag-coefficient. Of course, for $t \to \infty$, those values converge to the same stationary limit as the other methods. The fractional-step-θ scheme mainly leads to the same evolution of the drag-coefficient as the "H-projection".

In the next section, we will theoretically investigate this error and see that using the H-projection of the old velocity into the new finite element space leads to exactly the same velocity in the first time step on the new spatial mesh as for the "original" method, but also to a bounded approximation of the pressure.



(a) Switching from $\mathcal{T}_{2h}$ to $\mathcal{T}_h$



(b) Switching from $\mathcal{T}_h$ to $\mathcal{T}_{2h}$

Figure 5.7. Temporal evolution of the drag-coefficient for different initial values

5.5 Theoretical investigation

This section presents a theoretical investigation of the behavior of the pressure approximation when switching the spatial mesh. To this end, we consider the inf-sup-stable $\mathcal{Q}_2/\mathcal{Q}_1$-Taylor-Hood element for the spatial discretization in combination with the backward Euler time-stepping scheme. As in the previous sections, we consider uniform refinement or coarsening of a uniformly refined mesh. Furthermore, we assume the domain $\Omega \subseteq \mathbb{R}^d$, $d \in \{ 2, 3 \}$ to be polygonally bounded and convex.

Let $(\boldsymbol{v}, p)^T \in H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R}$ be the unique solution of the stationary Stokes problem for $\nu = 1$:

$$\begin{aligned} (\nabla \boldsymbol{v}, \nabla \boldsymbol{\psi}) - (p, \nabla \cdot \boldsymbol{\psi}) &= (\boldsymbol{f}, \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_0^1(\Omega)^d, \\ (\nabla \cdot \boldsymbol{v}, \chi) &= 0 && \forall \chi \in L^2(\Omega)/\mathbb{R}. \end{aligned} \tag{5.4}$$

Then, this solution also satisfies $(\boldsymbol{v}, p)^T \in H^2(\Omega)^d \times H^1(\Omega)$ as well as the a priori estimate

$$\|\boldsymbol{v}\|_{H^2(\Omega)} + \|p\|_{H^1} \leq C \|\boldsymbol{f}\|, \tag{5.5}$$

see Kellogg and Osborn [76] and Dauge [34].

Let a uniform decomposition $\mathcal{T}_H$ of $\Omega \subseteq \mathbb{R}^d$ into cells be given. We define the following conforming finite element spaces for the Taylor-Hood element:

$$\begin{aligned} H_H &\coloneqq \left\{ v_H \in C(\overline{\Omega}) \;\middle|\; v_H\big|_K \in \mathcal{Q}_2(K) \; \forall K \in \mathcal{T}_H \right\} \cap H_0^1(\Omega), \\ L_H &\coloneqq \left\{ p_H \in C(\overline{\Omega}) \;\middle|\; p_H\big|_K \in \mathcal{Q}_1(K) \; \forall K \in \mathcal{T}_H \right\} \cap L^2(\Omega)/\mathbb{R}. \end{aligned}$$

Let $(\boldsymbol{v_H}, p_H)^T \in H_H^d \times L_H$ be the approximate solution on the mesh $\mathcal{T}_H$, that is

$$\begin{aligned} (\nabla \boldsymbol{v_H}, \nabla \boldsymbol{\psi}) - (p_H, \nabla \cdot \boldsymbol{\psi}) &= (\boldsymbol{f}, \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_H^d, \\ (\nabla \cdot \boldsymbol{v_H}, \chi) &= 0 && \forall \chi \in L_H. \end{aligned} \tag{5.6}$$

Uniformly refining or coarsening the mesh $\mathcal{T}_H$ yields a spatial mesh $\mathcal{T}_h$ and the corresponding finite element spaces

$$\begin{aligned} H_h &\coloneqq \left\{ v_h \in C(\overline{\Omega}) \;\middle|\; v_h\big|_K \in \mathcal{Q}_2(K) \; \forall K \in \mathcal{T}_h \right\} \cap H_0^1(\Omega), \\ L_h &\coloneqq \left\{ p_h \in C(\overline{\Omega}) \;\middle|\; p_h\big|_K \in \mathcal{Q}_1(K) \; \forall K \in \mathcal{T}_h \right\} \cap L^2(\Omega)/\mathbb{R}. \end{aligned}$$

Performing one backward Euler step with step size k, seeks the solution $(\boldsymbol{v_h^k}, p_h^k)^T \in H_h^d \times L_h$ of

$$\begin{aligned} \frac{1}{k}(\boldsymbol{v_h^k}, \boldsymbol{\psi}) + (\nabla \boldsymbol{v_h^k}, \nabla \boldsymbol{\psi}) - (p_h^k, \nabla \cdot \boldsymbol{\psi}) &= \frac{1}{k}(\boldsymbol{v_H}, \boldsymbol{\psi}) + (\boldsymbol{f}, \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot \boldsymbol{v_h^k}, \chi) &= 0 && \forall \chi \in L_h. \end{aligned} \tag{5.7}$$

Using the H-projection $\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}$ of $\boldsymbol{v}_{\boldsymbol{H}}$ into H_h^d as initial value in the backward Euler step, yields the solution $(\hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}, \hat{p}_h^k)^T \in H_h^d \times L_h$ of

$$\begin{aligned} \frac{1}{k}(\hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}, \boldsymbol{\psi}) + (\nabla \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}, \nabla \boldsymbol{\psi}) - (\hat{p}_h^k, \nabla \cdot \boldsymbol{\psi}) &= \frac{1}{k}(\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \boldsymbol{\psi}) + (\boldsymbol{f}, \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}, \chi) &= 0 && \forall \chi \in L_h. \end{aligned} \tag{5.8}$$

Hereby, $\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}$ is given as the first component of the solution $(\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \tilde{p}_h^H)^T \in H_h^d \times L_h$ of

$$\begin{aligned} (\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \boldsymbol{\psi}) - (\tilde{p}_h^H, \nabla \cdot \boldsymbol{\psi}) &= (\boldsymbol{v}_{\boldsymbol{H}}, \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \chi) &= 0 && \forall \chi \in L_h. \end{aligned} \tag{5.9}$$

If we subtract equation (5.8) from (5.7), we obtain

$$\begin{aligned} \frac{1}{k}(\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}, \boldsymbol{\psi}) + (\nabla(\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}), \nabla \boldsymbol{\psi}) - (p_h^k - \hat{p}_h^k, \nabla \cdot \boldsymbol{\psi}) &= -\frac{1}{k}(\tilde{p}_h^H, \nabla \cdot \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot (\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}), \chi) &= 0 && \forall \chi \in L_h, \end{aligned} \tag{5.10}$$

where we have applied (5.9) to obtain the right-hand side. Testing (5.10) with $\boldsymbol{\psi} = \boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}} \in H_h^d$ and $\chi = p_h^k - \hat{p}_h^k$ leads to

$$\frac{1}{k}\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}\right\|^2 + \left\|\nabla(\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}})\right\|^2 = 0,$$

where the other terms cancel out due to the second equation of (5.10). Hence, we have $\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} = \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}$.

When testing equation (5.7) with $\boldsymbol{\psi} = \boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}$ and $\chi = p_h^k$, we obtain

$$\frac{1}{k}\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}\right\|^2 + \left\|\nabla \boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}\right\|^2 = \frac{1}{k}(\boldsymbol{v}_{\boldsymbol{H}}, \boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}) + (\boldsymbol{f}, \boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}})$$

and hence

$$\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}\right\| \leq \|\boldsymbol{v}_{\boldsymbol{H}}\| + k\,\|\boldsymbol{f}\|\,.$$

Since $\boldsymbol{v}_{\boldsymbol{H}}$ and $\boldsymbol{f}$ do not depend on time, we conclude that $\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}\right\|$ remains bounded for $k \to 0$. Hence, there is at least one subsequence (which we also denote by $\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}$) and a function $\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{0}} \in H_h^d$ such that

$$\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{0}}\right\| \to 0 \quad (k \to 0).$$

Since H_h^d is finite dimensional, $\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}$ converges to $\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{0}}$ in every norm, even point-wise.

Now let $\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}}$ denote such an arbitrary subsequence and $\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{0}} \in H_h^d$ the corresponding limit. Equation (5.7) is equivalent to the algebraic system

$$\begin{pmatrix} \boldsymbol{M} + k\boldsymbol{A} & k\boldsymbol{B} \\ -\boldsymbol{B}^T & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x}^{\boldsymbol{k}} \\ \boldsymbol{y}^{\boldsymbol{k}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}^{\boldsymbol{k}} \\ 0 \end{pmatrix} \tag{5.11}$$

with

$$\boldsymbol{M} = \Big((\boldsymbol{\psi_j}, \boldsymbol{\psi_i})\Big)_{i,j=1,\dots,N_H},\ \boldsymbol{A} = \Big((\nabla\boldsymbol{\psi_j}, \nabla\boldsymbol{\psi_i})\Big)_{i,j=1,\dots,N_H},\ \boldsymbol{B} = \Big(-(\chi_j, \nabla\cdot\boldsymbol{\psi_i})\Big)_{\substack{i=1,\dots,N_H,\\ j=1,\dots,N_L}}$$

and right-hand side

$$\boldsymbol{b^k} = \Big((\boldsymbol{v_H}, \boldsymbol{\psi_i}) + k(\boldsymbol{f}, \boldsymbol{\psi_i})\Big)_{i=1,\dots,N_H}$$

where we use the representations

$$\boldsymbol{v_h^k} = \sum_{j=1}^{N_H} x_j^k \boldsymbol{\psi_j} \quad \text{and} \quad p_h^k = \sum_{j=1}^{N_L} y_j^k \chi_j.$$

Here, $\{\ \boldsymbol{\psi_j} \mid j = 1,\dots,N_H\ \}$ is a basis of H_h^d while $\{\ \chi_j \mid j = 1,\dots,N_L\ \}$ is a basis of L_h. This especially means

$$\boldsymbol{M}\boldsymbol{x^k} + k\boldsymbol{A}\boldsymbol{x^k} + k\boldsymbol{B}\boldsymbol{y^k} = \boldsymbol{b^k}. \tag{5.12}$$

Since $\boldsymbol{v_h^k}$ converges point-wise to $\boldsymbol{v_h^0}$, we have $\boldsymbol{x^k} \to \boldsymbol{x^0}$ with

$$\boldsymbol{v_h^0} = \sum_{j=1}^{N_H} x_j^0 \boldsymbol{\psi_j}.$$

For $k \to 0$, we have

$$\boldsymbol{b^k} \to \boldsymbol{b^0} = \Big((\boldsymbol{v_H}, \boldsymbol{\psi_i})\Big)_{i=1,\dots,N_H},$$

because $\boldsymbol{v_H}$ and $\boldsymbol{f}$ do not depend on time. Introducing the L^2-projection $\boldsymbol{P_h v_H} \in H_h^d$ of $\boldsymbol{v_H}$ onto H_h^d determined by

$$(\boldsymbol{P_h v_H}, \boldsymbol{\psi}) = (\boldsymbol{v_H}, \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_h^d,$$

we may write $\boldsymbol{b^0} = \boldsymbol{M}\bar{\boldsymbol{x}}$ where $\bar{\boldsymbol{x}} \in \mathbb{R}^{N_H}$ is given through the relation

$$\boldsymbol{P_h v_H} = \sum_{j=1}^{N_H} \bar{x}_j \boldsymbol{\psi_j}.$$

In virtue of (5.12), we conclude that $k\boldsymbol{y^k}$ also converges for $k \to 0$. By passing to the limit $k \to 0$ in (5.12), we obtain

$$\boldsymbol{M}\boldsymbol{x^0} + \boldsymbol{B}\boldsymbol{y^0} = \boldsymbol{M}\bar{\boldsymbol{x}}. \tag{5.13}$$

$\boldsymbol{y^0}$ hereby is the limit of $k\boldsymbol{y^k}$ for $k \to 0$. Equation (5.13) may equivalently be written as

$$(\boldsymbol{v_h^0}, \boldsymbol{\psi}) - (p_h^0, \nabla\cdot\boldsymbol{\psi}) = (\boldsymbol{P_h v_H}, \boldsymbol{\psi}) = (\boldsymbol{v_H}, \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_h^d,$$

which together with the second equation in (5.11) states that $\boldsymbol{v_h^0}$ is just the H-projection of $\boldsymbol{v_H}$ into H_h^d. The Lagrange multiplier $p_h^0 \in L_h$ herein is given as

$$p_h^0 = \sum_{j=1}^{N_L} y_j^0 \chi_j.$$

If $\boldsymbol{y^0} = \boldsymbol{0}$, we then have $\boldsymbol{x^0} = \bar{\boldsymbol{x}}$ or equivalently $\boldsymbol{v_h^0} = \boldsymbol{P_h v_H}$. Furthermore, this implies

$$(\nabla \cdot \boldsymbol{P_h v_H}, \chi) = (\nabla \cdot \boldsymbol{v_h^0}, \chi) = 0 \quad \forall \chi \in L_h.$$

This in general will not be true. Hence, we conclude $\boldsymbol{y^0} \neq \boldsymbol{0}$ in general. Thus, there is $j \in \{ 1, \ldots, N_L \}$ with $y_j^0 \neq 0$ and hence

$$k y_j^k \not\to 0.$$

Then, we obviously obtain

$$\left| y_j^k \right| \geq C \frac{1}{k} \quad \text{or} \quad \left\| p_h^k \right\| \geq C \frac{1}{k}.$$

Remark 5.1. If $\mathcal{T}_h$ is obtained from $\mathcal{T}_H$ by uniform refinement, then we obviously have $H_H \subseteq H_h$ as well as $L_H \subseteq L_h$ and thus the L^2-projection from H_H onto H_h is the identity mapping. As a consequence, we have $\boldsymbol{P_h v_H} = \boldsymbol{v_H}$. However, in general, we have

$$(\nabla \cdot \boldsymbol{v_H}, \chi) \neq 0$$

for $\chi \in L_h \setminus L_H$ also in this case.

In the remaining part of this section, we want to show that the pressure approximations $\hat{p}_h^k$ obtained through equation (5.8) remain bounded for $k \to 0$.

Subtracting $\frac{1}{k}$ times the first equation of (5.9) from the first equation of (5.7) leads to

$$\frac{1}{k}(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \boldsymbol{\psi}) + (\nabla \boldsymbol{v_h^k}, \nabla \boldsymbol{\psi}) - \frac{1}{k}(k p_h^k - \tilde{p}_h^H, \nabla \cdot \boldsymbol{\psi}) = (\boldsymbol{f}, \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_h^d$$

or equivalently

$$\begin{aligned}\frac{1}{k}(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \boldsymbol{\psi}) + (\nabla(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}), \nabla \boldsymbol{\psi}) - \frac{1}{k}(k p_h^k - \tilde{p}_h^H, \nabla \cdot \boldsymbol{\psi}) \\ = (\boldsymbol{f}, \boldsymbol{\psi}) - (\nabla \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \nabla \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_h^d.\end{aligned}$$

By testing with $\boldsymbol{\psi} = \boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H} \in (H_h^k)^d$, we obtain

$$\begin{aligned}\frac{1}{k} \left\| \boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H} \right\|^2 + \left\| \nabla(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}) \right\|^2 \\ = (\boldsymbol{f}, \boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}) - (\nabla \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \nabla(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H})),\end{aligned} \tag{5.14}$$

because the other terms cancel out due to the second equations of (5.7) and (5.9). For the further analysis we need to introduce some auxiliary quantities: Let the approximate solution $(\boldsymbol{v_h}, p_h)^T \in H_h^d \times L_h$ of the Stokes problem on the mesh $\mathcal{T}_h$ be given by

$$\begin{aligned}(\nabla \boldsymbol{v_h}, \nabla \boldsymbol{\psi}) - (p_h, \nabla \cdot \boldsymbol{\psi}) &= (\boldsymbol{f}, \boldsymbol{\psi}) \qquad \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot \boldsymbol{v_h}, \chi) &= 0 \qquad \forall \chi \in L_h.\end{aligned} \tag{5.15}$$

The H-projection $\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}$ of the continuous velocity $\boldsymbol{v}$ into H_h^d is given as the first component of the solution $(\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}, \tilde{p}_h)^T \in H_h^d \times L_h$ of

$$\begin{aligned} (\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}, \boldsymbol{\psi}) - (\tilde{p}_h, \nabla \cdot \boldsymbol{\psi}) &= (\boldsymbol{v}, \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}, \chi) &= 0 && \forall \chi \in L_h. \end{aligned} \tag{5.16}$$

Similarly, the V-projection $\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}$ of the continuous velocity $\boldsymbol{v}$ into H_h^d is given as the first component of the solution $(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}, \tilde{r}_h)^T \in H_h^d \times L_h$ of

$$\begin{aligned} (\nabla\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}, \nabla\boldsymbol{\psi}) - (\tilde{r}_h, \nabla \cdot \boldsymbol{\psi}) &= (\nabla\boldsymbol{v}, \nabla\boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}, \chi) &= 0 && \forall \chi \in L_h. \end{aligned} \tag{5.17}$$

We then have for arbitrary $\boldsymbol{\psi} \in H_h^d$

$$\begin{aligned} \left|(\nabla\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \nabla\boldsymbol{\psi})\right| &= \left|(\nabla(\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}), \nabla\boldsymbol{\psi}) + (\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}), \nabla\boldsymbol{\psi}) + (\nabla\boldsymbol{v}, \nabla\boldsymbol{\psi})\right| \\ &\leq \left\{\left\|\nabla(\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v})\right\| + \left\|\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})\right\|\right\} \|\nabla\boldsymbol{\psi}\| + |(\Delta\boldsymbol{v}, \boldsymbol{\psi})| \\ &\leq C\Big\{h^{-2}\left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\right\| + h^{-1}\left\|\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})\right\| + \|\Delta\boldsymbol{v}\|\Big\} \|\boldsymbol{\psi}\| \\ &\leq Ch^{-2}\Big\{\left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}\right\| + \left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}\right\| + \left\|\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\right\| \\ &\quad + h\left\|\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})\right\| + h^2\|\Delta\boldsymbol{v}\|\Big\} \|\boldsymbol{\psi}\|, \end{aligned} \tag{5.18}$$

where in the penultimate line inverse estimates have been used. We will now treat each term separately.

By subtracting equation (5.16) from (5.9), we obtain

$$\begin{aligned} (\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}, \boldsymbol{\psi}) - (\tilde{p}_h^H - \tilde{p}_h, \nabla \cdot \boldsymbol{\psi}) &= (\boldsymbol{v}_{\boldsymbol{H}} - \boldsymbol{v}, \boldsymbol{\psi}) \qquad && \forall \boldsymbol{\psi} \in H_h^d, \\ (\nabla \cdot (\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}), \chi) &= 0 && \forall \chi \in L_h. \end{aligned} \tag{5.19}$$

Testing with $\boldsymbol{\psi} \coloneqq \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}$ and $\chi \coloneqq \tilde{p}_h^H - \tilde{p}_h$ yields

$$\left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}\right\|^2 = (\boldsymbol{v}_{\boldsymbol{H}} - \boldsymbol{v}, \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v})$$

and hence due to the Cauchy-Schwarz inequality

$$\left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}\right\| \leq \|\boldsymbol{v}_{\boldsymbol{H}} - \boldsymbol{v}\|.$$

Using standard approximation results (see, for instance, Girault and Raviart [58] as well as Lemma 3.1 and Lemma 3.2) and the a priori estimate (5.5), we obtain

$$\left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}\right\| \leq \|\boldsymbol{v} - \boldsymbol{v}_{\boldsymbol{H}}\| \leq CH^2\Big\{\|\boldsymbol{v}\|_{H^2} + \|p\|_{H^1}\Big\} \leq CH^2\|\boldsymbol{f}\|. \tag{5.20}$$

In order to estimate the next term, we note that from (5.16) we have

$$(\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}, \boldsymbol{\psi}) = (\tilde{p}_h, \nabla \cdot \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_h^d.$$

Testing with $\boldsymbol{\psi} \coloneqq \boldsymbol{v_h} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}$ and recalling that

$$(\nabla \cdot \boldsymbol{v_h}, \chi) = (\nabla \cdot \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}, \chi) = 0 \quad \forall \chi \in L_h,$$

we conclude

$$(\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}, \boldsymbol{v_h} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}) = (\tilde{p}_h, \nabla \cdot (\boldsymbol{v_h} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v})) = 0.$$

Thus, we have

$$\left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}\right\|^2 = (\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}, \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}) = (\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}, \boldsymbol{v_h} - \boldsymbol{v}) \leq \left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}\right\| \|\boldsymbol{v_h} - \boldsymbol{v}\|$$

and therefore

$$\left\|\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}\right\| \leq \|\boldsymbol{v} - \boldsymbol{v_h}\| \leq Ch^2 \|\boldsymbol{f}\|. \tag{5.21}$$

Similarly, from (5.17) we see that

$$(\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}), \nabla\boldsymbol{\psi}) = (\tilde{r}_h, \nabla \cdot \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_h^d \tag{5.22}$$

and hence by testing with $\boldsymbol{\psi} \coloneqq \boldsymbol{v_h} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}$

$$(\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}), \nabla(\boldsymbol{v_h} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v})) = (\tilde{r}_h, \nabla \cdot (\boldsymbol{v_h} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v})) = 0.$$

This yields

$$\begin{aligned}\left\|\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})\right\|^2 &= (\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}), \nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})) = (\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}), \nabla(\boldsymbol{v_h} - \boldsymbol{v})) \\ &\leq \left\|\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})\right\| \|\nabla(\boldsymbol{v_h} - \boldsymbol{v})\|\end{aligned}$$

and thus

$$\left\|\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})\right\| \leq \|\nabla(\boldsymbol{v} - \boldsymbol{v_h})\| \leq Ch \|\boldsymbol{f}\|. \tag{5.23}$$

Before estimating $\left\|\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}\right\|$, we recall equations (5.22) and (5.23) as well as the inf-sup stability condition (3.8) which allows us to bound $\|\tilde{r}_h\|$:

$$\beta \|\tilde{r}_h\| \leq \sup_{\boldsymbol{\psi} \in H_h^d} \frac{(\tilde{r}_h, \nabla \cdot \boldsymbol{\psi})}{\|\nabla\boldsymbol{\psi}\|} = \sup_{\boldsymbol{\psi} \in H_h^d} \frac{(\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}), \nabla\boldsymbol{\psi})}{\|\nabla\boldsymbol{\psi}\|} \leq \left\|\nabla(\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v})\right\| \leq Ch \|\boldsymbol{f}\|. \tag{5.24}$$

In order to estimate $\left\|\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} - \boldsymbol{v}\right\|$, we use a duality argument due to Aubin [3] and Nitsche [85]: Let $(\boldsymbol{w}, q)^T \in H_0^1(\Omega)^d \times L^2(\Omega)/\mathbb{R}$ be the unique solution of

$$\begin{aligned}(\nabla\boldsymbol{\psi}, \nabla\boldsymbol{w}) + (\nabla \cdot \boldsymbol{\psi}, q) &= \frac{1}{\left\|\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\right\|}(\boldsymbol{\psi}, \boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}) \qquad \forall \boldsymbol{\psi} \in H_0^1(\Omega)^d, \\ -(\chi, \nabla \cdot \boldsymbol{w}) &= 0 \qquad \forall \chi \in L^2(\Omega)/\mathbb{R}.\end{aligned} \tag{5.25}$$

Since $\frac{\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}}{\|\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\|} \in L^2(\Omega)^d$, we have as for the primal problem (5.4) the a priori estimate

$$\|\boldsymbol{w}\|_{H^2} + \|q\|_{H^1} \leq C \left\|\frac{\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}}{\|\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\|}\right\| = C. \tag{5.26}$$

Testing the first equation of (5.25) with $\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v} \in H_0^1(\Omega)^d$, we obtain

$$\left\|\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\right\| = (\nabla(\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}), \nabla\boldsymbol{w}) + (\nabla\cdot(\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}), q).$$

Recalling (5.22) and the fact that $(\nabla\cdot\boldsymbol{v},\chi) = (\nabla\cdot\boldsymbol{w},\chi) = (\nabla\cdot\widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v},\chi) = 0\ \forall\chi\in L_h$, we may write for arbitrary $\hat{\boldsymbol{w}}_{\boldsymbol{h}} \in H_h^d$ and $\hat{q}_h \in L_h$

$$\begin{aligned}\left\|\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\right\| &= (\nabla(\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}), \nabla(\boldsymbol{w} - \hat{\boldsymbol{w}}_{\boldsymbol{h}}) + (\nabla\cdot(\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}), q - \hat{q}_h) - (\tilde{r}_h, \nabla\cdot(\boldsymbol{w} - \hat{\boldsymbol{w}}_{\boldsymbol{h}}))\\ &\le C\Big\{\left\|\nabla(\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v})\right\| + \|\tilde{r}_h\|\Big\}\Big\{\|\nabla(\boldsymbol{w} - \hat{\boldsymbol{w}}_{\boldsymbol{h}})\| + \|q - \hat{q}_h\|\Big\}.\end{aligned}$$

Inserting the approximation properties of Lemma 3.1 and Lemma 3.2

$$\begin{aligned}\|\nabla(\boldsymbol{w} - \hat{\boldsymbol{w}}_{\boldsymbol{h}})\| &\le Ch\,\|\boldsymbol{w}\|_{H^2},\\ \|q - \hat{q}_h\| &\le Ch\,\|q\|_{H^1}\end{aligned}$$

as well as the estimates (5.23) and (5.24), we have

$$\left\|\boldsymbol{v} - \widetilde{\boldsymbol{R}}_{\boldsymbol{h}}\boldsymbol{v}\right\| \le Ch^2\,\|\boldsymbol{f}\|\left\{\|\boldsymbol{w}\|_{H^2} + \|q\|_{H^1}\right\} \le Ch^2\,\|\boldsymbol{f}\|, \tag{5.27}$$

where the last inequality is obtained by applying the a priori estimate (5.26). Finally, we obviously have

$$\|\Delta\boldsymbol{v}\| \le \|\boldsymbol{v}\|_{H^2} \le C\,\|\boldsymbol{f}\|. \tag{5.28}$$

Inserting the estimates (5.20), (5.21), (5.23), (5.27), and (5.28) into (5.18) then yields

$$\left|(\nabla\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \nabla\boldsymbol{\psi})\right| \le C\Big(1 + (\tfrac{H}{h})^2\Big)\,\|\boldsymbol{f}\|\,\|\boldsymbol{\psi}\|.$$

Since we only consider uniform refinement or uniform coarsening, we have $H = 2h$ or $H = \frac{1}{2}h$ and thus

$$\left|(\nabla\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \nabla\boldsymbol{\psi})\right| \le C\,\|\boldsymbol{f}\|\,\|\boldsymbol{\psi}\|. \tag{5.29}$$

This allows us to conclude from (5.14):

$$\begin{aligned}\frac{1}{k}\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}\right\|^2 + \left\|\nabla(\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}})\right\|^2 &\le \|\boldsymbol{f}\|\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}\right\| + \left|(\nabla\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \nabla(\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}))\right|\\ &\le C\,\|\boldsymbol{f}\|\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}\right\|\end{aligned}$$

and hence

$$\frac{1}{k}\left\|\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}\right\| \le C\,\|\boldsymbol{f}\|. \tag{5.30}$$

By using the Poincaré inequality, we also obtain from (5.29)

$$\left|(\nabla\widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}}, \nabla\boldsymbol{\psi})\right| \le C\,\|\boldsymbol{f}\|\,\|\nabla\boldsymbol{\psi}\| \tag{5.31}$$

and therefore from (5.14) also

$$\left\|\nabla(\boldsymbol{v}_{\boldsymbol{h}}^{\boldsymbol{k}} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}}\boldsymbol{v}_{\boldsymbol{H}})\right\| \le C\,\|\boldsymbol{f}\|. \tag{5.32}$$

To show that $\left\|\hat{p}_h^k\right\|$ remains bounded for $k \to 0$, we use the inf-sup condition and the fact that $\boldsymbol{v_h^k} = \hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}$ which allows us to replace $\hat{\boldsymbol{v}}_{\boldsymbol{h}}^{\boldsymbol{k}}$ by $\boldsymbol{v_h^k}$ in (5.8):

$$\begin{aligned}
\beta \left\|\hat{p}_h^k\right\| &\leq \sup_{\boldsymbol{\psi} \in H_h^d} \frac{(\hat{p}_h^k, \nabla \cdot \boldsymbol{\psi})}{\|\nabla \boldsymbol{\psi}\|} \\
&= \sup_{\boldsymbol{\psi} \in H_h^d} \frac{\frac{1}{k}(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \boldsymbol{\psi}) + (\nabla(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}), \nabla \boldsymbol{\psi}) + (\nabla \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \nabla \boldsymbol{\psi}) - (\boldsymbol{f}, \boldsymbol{\psi})}{\|\nabla \boldsymbol{\psi}\|} \\
&\leq C \frac{1}{k} \left\|\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}\right\| + \left\|\nabla(\boldsymbol{v_h^k} - \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H})\right\| + \sup_{\boldsymbol{\psi} \in H_h^d} \frac{\left|(\nabla \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \nabla \boldsymbol{\psi})\right|}{\|\nabla \boldsymbol{\psi}\|} + C \|\boldsymbol{f}\| \\
&\leq C \|\boldsymbol{f}\| .
\end{aligned}$$

Here, the estimates (5.30), (5.31), and (5.32) have been used. Since the right-hand side is independent of k, we have shown that $\left\|\hat{p}_h^k\right\|$ remains bounded for $k \to 0$.

Remark 5.2. The arguments used above to show that $\left\|p_h^k\right\| \geq C \frac{1}{k}$ if there is a $\chi \in L_h$ such that $(\nabla \cdot v_H, \chi) \neq 0$ are not restricted to the case of uniform refinement or coarsening of the meshes. Actually, they directly carry over to the case of arbitrary refinement or coarsening of cells. The estimate for the boundedness of $\left\|\hat{p}_h^k\right\|$ can be generalized as long we are able to bound $\left\|\nabla \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}\right\|$ and $\left\|\tilde{\Delta}_h \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}\right\|$ where $\tilde{\Delta}_h \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}$ denotes the discrete Stokes operator of $\widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}$ given by

$$(\tilde{\Delta}_h \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \boldsymbol{\psi}) = -(\nabla \widetilde{\boldsymbol{P}}_{\boldsymbol{h}} \boldsymbol{v_H}, \nabla \boldsymbol{\psi}) \quad \forall \boldsymbol{\psi} \in H_h^d \cap \{\, \boldsymbol{\psi} \mid (\nabla \cdot \boldsymbol{\psi}, \chi) = 0 \ \forall \chi \in L_h \,\} .$$

This theoretical result also substantiates the numerical experiences of the previous section where we obtained bounded approximations of the pressure by applying the H-projection for determining the initial value when switching the spatial mesh. We showed that on dynamic spatial meshes bounded pressure approximations are only possible if the velocity from the first mesh is divergence-free with respect to the test functions of the new mesh. Otherwise, the pressure approximation contains $\frac{1}{k}$ times the Lagrange multiplier occurring in the H-projection of the old velocity field into the new finite element space which leads to the unbounded behavior for $k \to 0$.

6 Applications

In this chapter, we apply the developed adaptive algorithm and a posteriori error estimators to three concrete applications.

The first two applications are the two- and three-dimensional benchmark configurations "Laminar Flow Around a Cylinder", see Schäfer and Turek [95]. Here, a time-dependent inflow profile is prescribed leading to a nonstationary flow. Aim of the simulation is the efficient computation of the mean drag- and lift-coefficients in the two-dimensional case as well as the mean drag-coefficient in the three-dimensional case.

As the third application we consider the simulation of the filling process of a lab-on-a-chip. This problem is formulated as a two-phase flow problem involving two incompressible phases, namely the liquid and the gaseous phase. At the beginning of the simulation, the computational domain is completely filled with the gaseous phase which is expelled by the liquid phase. Aim of the simulation here is the precise capturing of the interface between both phases.

6.1 Laminar flow around a cylinder

In these applications, the flow of an incompressible Newtonian fluid around an obstacle is considered. The governing equations are the incompressible Navier-Stokes equations for homogeneous fluids:

$$\begin{aligned} \partial_t \boldsymbol{v} - \nu \Delta \boldsymbol{v} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + \rho^{-1} \nabla p &= \mathbf{0}, \\ \nabla \cdot \boldsymbol{v} &= 0. \end{aligned}$$

The kinematic viscosity is set to $\nu = 10^{-3}\,\mathrm{m}^2\,\mathrm{s}^{-1}$ while the density is given by $\rho = 1\,\mathrm{kg}\,\mathrm{m}^{-3}$. As initial condition $\boldsymbol{v}(0\,\mathrm{s}, \boldsymbol{x}) = \mathbf{0}\,\mathrm{m}\,\mathrm{s}^{-1}$ is chosen.

6.1.1 Two-dimensional test case

For the two-dimensional case, flow around an obstacle with circular cross-section is considered. The geometry is depicted in Figure 6.1.

The inflow condition on the left side of the domain is prescribed as

$$v_1(t, \boldsymbol{x}) = \frac{6 \sin(\frac{\pi t}{8\,\mathrm{s}})}{(0.41\,\mathrm{m})^2} x_2 (0.41\,\mathrm{m} - x_2)\,\mathrm{m}\,\mathrm{s}^{-1}, \quad v_2(t, \boldsymbol{x}) = 0\,\mathrm{m}\,\mathrm{s}^{-1}.$$

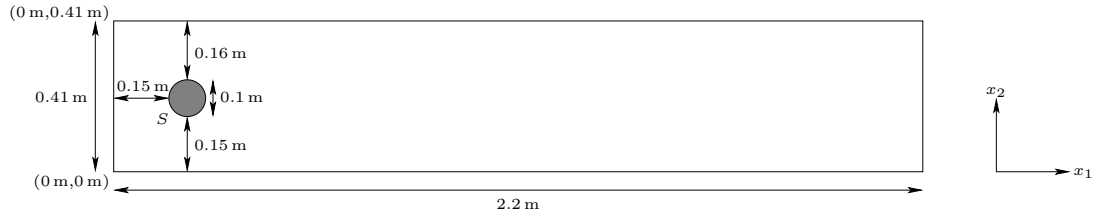


Figure 6.1. Geometry for the two-dimensional benchmark configuration

On the outflow boundary on the right side of the computational domain, we apply natural boundary conditions. We refer to Heywood, Rannacher, and Turek [67] for more information on these boundary conditions. On all other boundaries, we prescribe no-slip Dirichlet boundary conditions. The final time is set to $T = 8\,\mathrm{s}$. This setting leads to a Reynolds number $\mathrm{Re}(t) = \frac{\bar{U}(t)D}{\nu}$ based on the mean inflow velocity

$$\bar{U}(t) = \frac{2}{3} v_1(t, 0\,\mathrm{m}, 0.205\,\mathrm{m}) = \sin\Big(\frac{\pi t}{8\,\mathrm{s}}\Big)\,\mathrm{m\,s^{-1}}$$

and the diameter of the obstacle $D = 0.1\,\mathrm{m}$ of $0 \leq \mathrm{Re}(t) \leq 100$ for $0\,\mathrm{s} \leq t \leq 8\,\mathrm{s}$.

Remark 6.1. Due to the nonhomogeneous Dirichlet boundary conditions, the variational formulation has to be modified. Instead of seeking $\boldsymbol{u} = (\boldsymbol{v}, p)^T \in X$ satisfying

$$\int_I (\partial_t \boldsymbol{v}, \boldsymbol{\psi})\ \mathrm{d}t + a(\boldsymbol{u})(\boldsymbol{\varphi}) + (\boldsymbol{v}(0) - \boldsymbol{v^0}, \boldsymbol{\psi}(0)) = 0 \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T \in X, \tag{6.1}$$

we seek a solution $\boldsymbol{u} = (\boldsymbol{v}, p)^T \in (\boldsymbol{v_\Gamma}, 0)^T + X$ satisfying (6.1) where $\boldsymbol{v_\Gamma}$ is a divergence-free extension of the Dirichlet boundary conditions.

This modification also influences the derivation of the a posteriori error estimators because now the primal solution no longer is an admissible test function for the dual problem and hence in Lemma 4.1, we have

$$L'(y_1)(e) \neq 0.$$

This leads to additional terms in the a posteriori error estimators which are of the following form:

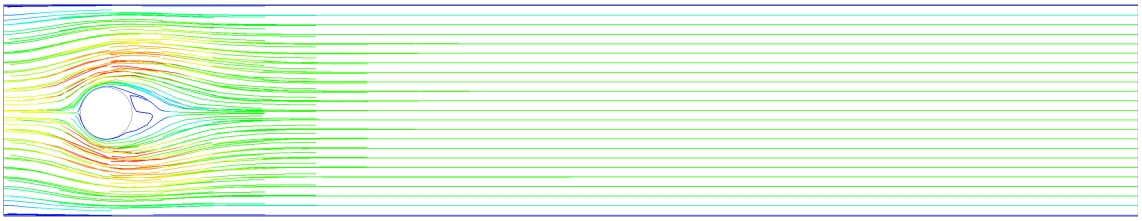
$$\rho^*(\boldsymbol{u}, \boldsymbol{z})(\boldsymbol{u} - \tilde{\boldsymbol{u}}_{\boldsymbol{k}}) \quad \text{and} \quad \rho^*(\boldsymbol{u_k}, \boldsymbol{z_k})(\boldsymbol{u_k} - \tilde{\boldsymbol{u}}_{\boldsymbol{kh}}). \tag{6.2}$$

However, these terms can be approximated using the same higher order reconstruction techniques as in Section 4.3 to replace (6.2) by

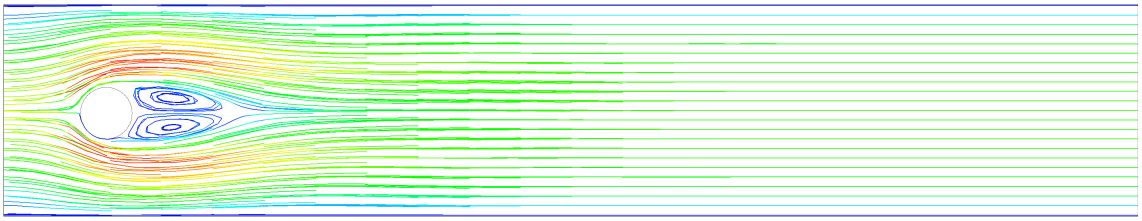
$$\rho^*(\boldsymbol{I_k u_{kh}}, \boldsymbol{I_k z_{kh}})(\boldsymbol{\Pi_k u_{kh}}) \quad \text{and} \quad \rho^*(\boldsymbol{I_h u_{kh}}, \boldsymbol{I_h z_{kh}})(\boldsymbol{\Pi_h u_{kh}})$$

with some interpolation operators $\boldsymbol{I_k}$ and $\boldsymbol{I_h}$.

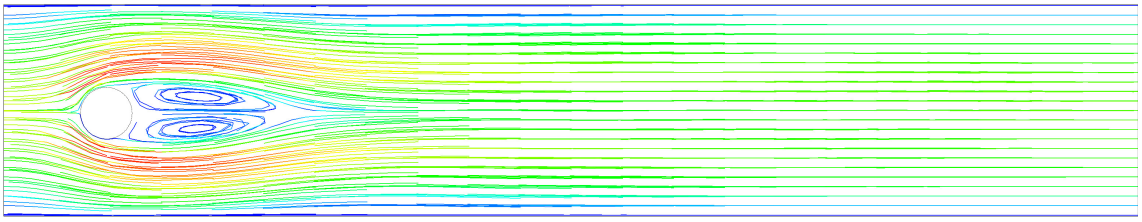
In Figure 6.2, we present the temporal development of the flow for $0\,\mathrm{s} \leq t \leq 8\,\mathrm{s}$. At $t = 2\,\mathrm{s}$ we note the development of two vortices behind the obstacle. Due to the increasing inflow velocity, these vortices detach from the obstacle between $t = 4\,\mathrm{s}$ and $t = 5\,\mathrm{s}$ and a van



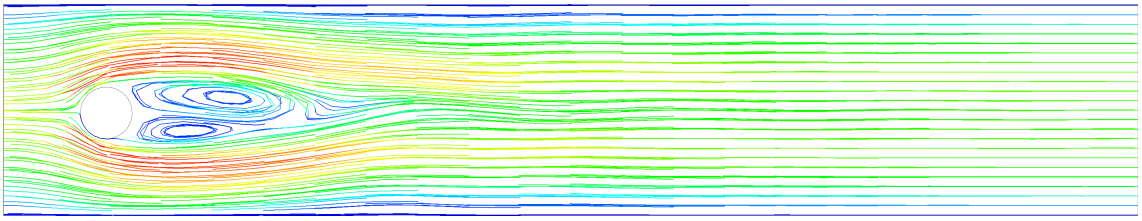
(a) $t = 1\,\mathrm{s}$



(b) $t = 2\,\mathrm{s}$

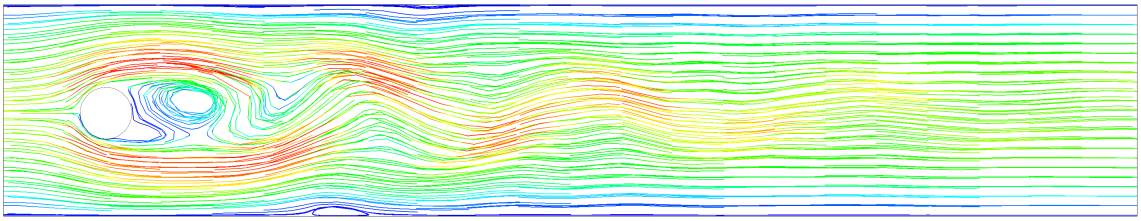


(c) $t = 3\,\mathrm{s}$

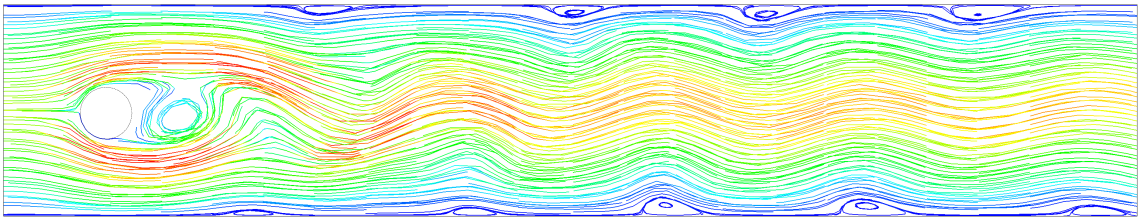


(d) $t = 4\,\mathrm{s}$

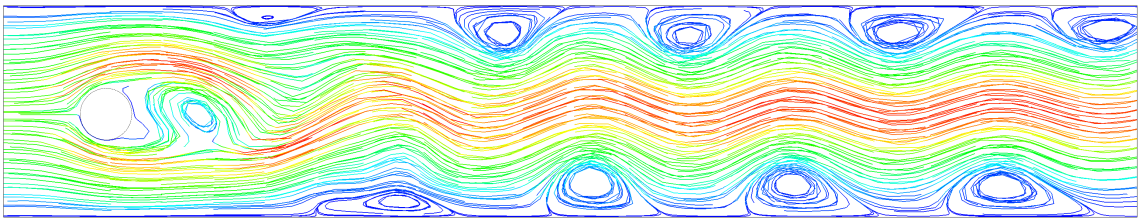
Figure 6.2. Two-dimensional test case: Streamlines at different times t



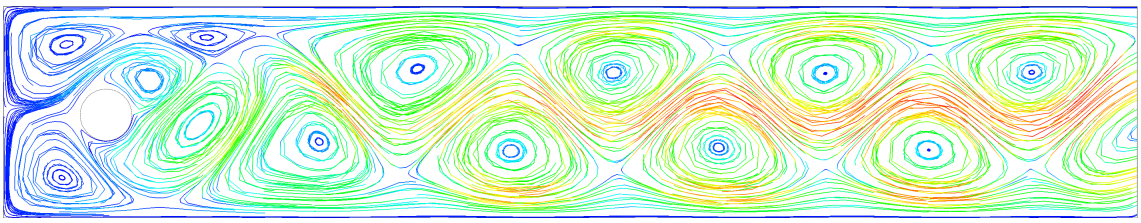
(e) $t = 5\,\mathrm{s}$



(f) $t = 6\,\mathrm{s}$



(g) $t = 7\,\mathrm{s}$



(h) $t = 8\,\mathrm{s}$

Figure 6.2. Two-dimensional test case: Streamlines at different times t (cont.)

Kármán vortex street develops. From $t = 6\,\mathrm{s}$ on, vortices near the boundary develop and get stronger while the inflow velocity is further decreased. At final time $t = 8\,\mathrm{s}$, the vortices are still visible even though the maximum velocity is much slower now.

The drag- and lift-coefficient are given as

$$c_{\mathrm{drag}} = C \int_S (\rho\nu\partial_{\boldsymbol{n}}(\boldsymbol{v}\cdot\boldsymbol{t})n_2 - pn_1)\,\mathrm{d}o \quad \text{and} \quad c_{\mathrm{lift}} = -C \int_S (\rho\nu\partial_{\boldsymbol{n}}(\boldsymbol{v}\cdot\boldsymbol{t})n_1 + pn_2)\,\mathrm{d}o,$$

where S denotes the surface of the obstacle, $\boldsymbol{n} = (n_1, n_2)^T$ is the normal vector on S, $(\boldsymbol{v}\cdot\boldsymbol{t})$ denotes the tangential component of $\boldsymbol{v}$ with tangential vector $\boldsymbol{t} = (n_2, -n_1)^T$, and the constant C is chosen as

$$C = \frac{2}{\rho\bar{U}^2 D}.$$

Hence, the quantities of interest are given by

$$\tilde{J}_{\mathrm{drag}}(\boldsymbol{u}) = \frac{1}{|I|}\int_I c_{\mathrm{drag}}\,\mathrm{d}t \quad \text{and} \quad \tilde{J}_{\mathrm{lift}}(\boldsymbol{u}) = \frac{1}{|I|}\int_I c_{\mathrm{lift}}\,\mathrm{d}t.$$

Note that these values can also be expressed using the semi-linear form $a(\boldsymbol{u})(\boldsymbol{\varphi})$ via

$$\begin{aligned} J_{\mathrm{drag}}(\boldsymbol{u}) &= -\int_I (\partial_t \boldsymbol{v}, \hat{\boldsymbol{\psi}}_{\mathbf{drag}})\,\mathrm{d}t - a(\boldsymbol{u})(\hat{\boldsymbol{\varphi}}_{\mathbf{drag}}) \quad \text{and} \\ J_{\mathrm{lift}}(\boldsymbol{u}) &= -\int_I (\partial_t \boldsymbol{v}, \hat{\boldsymbol{\psi}}_{\mathbf{lift}})\,\mathrm{d}t - a(\boldsymbol{u})(\hat{\boldsymbol{\varphi}}_{\mathbf{lift}}) \end{aligned} \tag{6.3}$$

with $\hat{\boldsymbol{\varphi}}_{\mathbf{drag}} = (\hat{\boldsymbol{\psi}}_{\mathbf{drag}}, 0)^T$ and $\hat{\boldsymbol{\varphi}}_{\mathbf{lift}} = (\hat{\boldsymbol{\psi}}_{\mathbf{lift}}, 0)^T$ fulfilling

$$\hat{\boldsymbol{\psi}}_{\mathbf{drag}}\big|_S = \begin{pmatrix} |I|^{-1} C \\ 0 \end{pmatrix}, \quad \hat{\boldsymbol{\psi}}_{\mathbf{drag}}\big|_{\partial\Omega\setminus S} = \mathbf{0}$$

and

$$\hat{\boldsymbol{\psi}}_{\mathbf{lift}}\big|_S = \begin{pmatrix} 0 \\ |I|^{-1} C \end{pmatrix}, \quad \hat{\boldsymbol{\psi}}_{\mathbf{lift}}\big|_{\partial\Omega\setminus S} = \mathbf{0}.$$

On the discrete level, both representations are not equivalent anymore. However, it can be shown that formulas (6.3) lead to a higher order of convergence with respect to h. Actually, we have

$$\tilde{J}_{\mathrm{drag}}(\boldsymbol{u_k}) - \tilde{J}_{\mathrm{drag}}(\boldsymbol{u_{kh}}) = O(h^s)$$

whereas

$$J_{\mathrm{drag}}(\boldsymbol{u_k}) - J_{\mathrm{drag}}(\boldsymbol{u_{kh}}) = O(h^{2s}).$$

The same holds true for $\tilde{J}_{\mathrm{lift}}$ and J_{lift}. For further information on this technique, we refer to Giles, Larson, Levenstam, and Süli [57].

Mean drag-coefficient

Let us first consider the mean drag-coefficient. Using the cG(2)dG(1) discretization with uniform refinement of the temporal and spatial discretizations, we obtain for the mean drag-coefficient the values listed in Table 6.1. As in Chapter 4, N denotes the number of spatial degrees of freedom and M denotes the number of time steps.

Table 6.1. Mean drag-coefficient obtained with the cG(2)dG(1) discretization and uniform refinement

N	M	$J_{\text{drag}}(\boldsymbol{u_{kh}})$
2124	80	1.6178974
8088	160	1.5695421
31536	320	1.6048954
124512	640	1.6071242
494784	1280	1.6072465
extrapolated		1.6072872

Now we employ the cG(1)cG(1), cG(2)cG(1), cG(1)dG(1), and cG(2)dG(1) discretization in combination with the adaptive Algorithm 4.1 to this problem. The results of these computations using adaptively refined spatial meshes which are kept constant over the whole time interval $I = (0\,\text{s}, 8\,\text{s})$ are shown in Tables 6.2–6.5. In these tables, we use the extrapolated value $J_{\text{drag}}(\boldsymbol{u}) = 1.6072872$.

Remark 6.2. The dG(0) discretization in time is not considered in this chapter due to its high numerical dissipation which makes the use of this discretization unfavorable for simulations in computational fluid dynamics.

Table 6.2. Mean drag-coefficient for the two-dimensional test case: Adaptive refinement with equilibration for the cG(1)cG(1) discretization

N	M	η_h	η_k	$J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})$	I_{eff}
582	80	$-1.92 \cdot 10^{-01}$	$7.61 \cdot 10^{-05}$	$2.75 \cdot 10^{-01}$	-1.43
1374	80	$-8.74 \cdot 10^{-02}$	$1.60 \cdot 10^{-04}$	$1.77 \cdot 10^{-01}$	-2.03
2238	80	$-1.02 \cdot 10^{-02}$	$1.72 \cdot 10^{-04}$	$6.78 \cdot 10^{-02}$	-6.75
5586	80	$5.40 \cdot 10^{-03}$	$-7.86 \cdot 10^{-03}$	$3.47 \cdot 10^{-02}$	-14.10
13872	132	$4.20 \cdot 10^{-03}$	$-7.52 \cdot 10^{-04}$	$1.11 \cdot 10^{-02}$	3.22
41340	132	$8.80 \cdot 10^{-04}$	$-8.00 \cdot 10^{-04}$	$8.53 \cdot 10^{-03}$	107.77
100794	134	$3.46 \cdot 10^{-04}$	$-6.39 \cdot 10^{-04}$	$8.21 \cdot 10^{-03}$	-27.96
322278	268	$5.90 \cdot 10^{-05}$	$8.54 \cdot 10^{-04}$	$1.99 \cdot 10^{-03}$	2.18
322278	452	$6.87 \cdot 10^{-05}$	$2.98 \cdot 10^{-04}$	$5.64 \cdot 10^{-04}$	1.54
322278	460	$6.82 \cdot 10^{-05}$	$2.94 \cdot 10^{-04}$	$5.74 \cdot 10^{-04}$	1.58
322278	920	$6.79 \cdot 10^{-05}$	$1.06 \cdot 10^{-04}$	$2.10 \cdot 10^{-04}$	1.20
993468	922	$6.20 \cdot 10^{-06}$	$1.06 \cdot 10^{-04}$	$1.61 \cdot 10^{-04}$	1.43

Table 6.3. Mean drag-coefficient for the two-dimensional test case: Adaptive refinement with equilibration for the cG(2)cG(1) discretization

N	M	η_h	η_k	$J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})$	I_{eff}
2124	80	$-4.73 \cdot 10^{-02}$	$-5.43 \cdot 10^{-03}$	$4.26 \cdot 10^{-02}$	-0.81
5304	80	$2.64 \cdot 10^{-02}$	$-1.12 \cdot 10^{-02}$	$5.55 \cdot 10^{-02}$	3.66
10992	160	$2.79 \cdot 10^{-03}$	$-4.44 \cdot 10^{-05}$	$9.18 \cdot 10^{-03}$	3.35
27000	160	$-6.75 \cdot 10^{-05}$	$-1.88 \cdot 10^{-04}$	$8.02 \cdot 10^{-03}$	-31.36
76080	268	$-2.63 \cdot 10^{-05}$	$8.22 \cdot 10^{-04}$	$1.93 \cdot 10^{-03}$	2.42
76080	450	$-1.85 \cdot 10^{-05}$	$2.97 \cdot 10^{-04}$	$4.69 \cdot 10^{-04}$	1.68
76080	460	$-1.76 \cdot 10^{-05}$	$2.94 \cdot 10^{-04}$	$4.72 \cdot 10^{-04}$	1.70
76080	920	$-1.19 \cdot 10^{-05}$	$1.06 \cdot 10^{-04}$	$1.15 \cdot 10^{-04}$	1.22
76080	922	$-1.19 \cdot 10^{-05}$	$1.06 \cdot 10^{-04}$	$1.15 \cdot 10^{-04}$	1.23
76080	1844	$-5.31 \cdot 10^{-06}$	$3.00 \cdot 10^{-05}$	$2.84 \cdot 10^{-05}$	1.15

Table 6.4. Mean drag-coefficient for the two-dimensional test case: Adaptive refinement with equilibration for the cG(1)dG(1) discretization

N	M	η_h	η_k	$J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})$	I_{eff}
582	80	$-2.99 \cdot 10^{-01}$	$2.16 \cdot 10^{-04}$	$1.35 \cdot 10^{-01}$	-0.45
1302	80	$-9.15 \cdot 10^{-02}$	$2.80 \cdot 10^{-04}$	$1.54 \cdot 10^{-01}$	-1.69
2280	80	$-7.39 \cdot 10^{-03}$	$3.26 \cdot 10^{-04}$	$7.03 \cdot 10^{-02}$	-9.94
5394	80	$4.29 \cdot 10^{-03}$	$3.96 \cdot 10^{-03}$	$3.64 \cdot 10^{-02}$	4.41
10998	120	$4.78 \cdot 10^{-03}$	$5.82 \cdot 10^{-03}$	$1.11 \cdot 10^{-02}$	1.05
25044	128	$1.64 \cdot 10^{-03}$	$4.76 \cdot 10^{-03}$	$6.89 \cdot 10^{-03}$	1.08
70146	256	$1.14 \cdot 10^{-04}$	$7.48 \cdot 10^{-04}$	$8.34 \cdot 10^{-04}$	0.97
70146	258	$1.14 \cdot 10^{-04}$	$7.32 \cdot 10^{-04}$	$8.15 \cdot 10^{-04}$	0.96
70146	516	$1.22 \cdot 10^{-04}$	$9.85 \cdot 10^{-05}$	$1.68 \cdot 10^{-04}$	0.76
235554	1032	$4.25 \cdot 10^{-05}$	$1.33 \cdot 10^{-05}$	$6.93 \cdot 10^{-05}$	1.24

Table 6.5. Mean drag-coefficient for the two-dimensional test case: Adaptive refinement with equilibration for the cG(2)dG(1) discretization

N	M	η_h	η_k	$J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})$	I_{eff}
2124	80	$-6.19 \cdot 10^{-02}$	$9.37 \cdot 10^{-04}$	$-1.06 \cdot 10^{-02}$	0.17
5448	80	$2.40 \cdot 10^{-02}$	$8.20 \cdot 10^{-03}$	$5.50 \cdot 10^{-02}$	1.71
11148	160	$2.38 \cdot 10^{-03}$	$4.22 \cdot 10^{-03}$	$5.47 \cdot 10^{-03}$	0.83
27132	252	$-2.29 \cdot 10^{-04}$	$7.79 \cdot 10^{-04}$	$2.09 \cdot 10^{-04}$	0.38
27132	258	$-2.29 \cdot 10^{-04}$	$7.31 \cdot 10^{-04}$	$1.55 \cdot 10^{-04}$	0.31
27132	516	$-1.92 \cdot 10^{-04}$	$9.88 \cdot 10^{-05}$	$-4.06 \cdot 10^{-04}$	4.33
76656	1032	$-3.08 \cdot 10^{-05}$	$1.33 \cdot 10^{-05}$	$-2.52 \cdot 10^{-05}$	1.44

We observe that for the cG(1) discretizations in space the spatial and the temporal discretization error are quite equilibrated, see Tables 6.2 and 6.4. On the other hand, Tables 6.3 and 6.5 show that for the cG(2) discretizations in space, the spatial discretization error converges with a higher order than the temporal discretization error. This is due to the evaluation of the mean drag-coefficient by J_{drag} instead of $\tilde{J}_{\text{drag}}$. Consequently, the equilibration algorithm keeps the spatial discretization fixed from time to time and only refines the temporal discretization. For all discretizations, we observe quite a good agreement of the estimated and the actual discretization error especially on finer discretizations which is indicated by $I_{\text{eff}} \approx 1$.

In Figure 6.3, we show the temporal evolution of the drag-coefficient c_{drag} for all four discretizations considered here. These values correspond to the finest adaptive discretization described in the last lines of Tables 6.2–6.5 with a relative error in the mean drag-coefficient of less than $5 \cdot 10^{-5}$ which corresponds to an absolute error of less than $8 \cdot 10^{-5}$ except for the cG(1)cG(1) discretization where we did not reach this accuracy. We note a perfect agreement of all four curves.

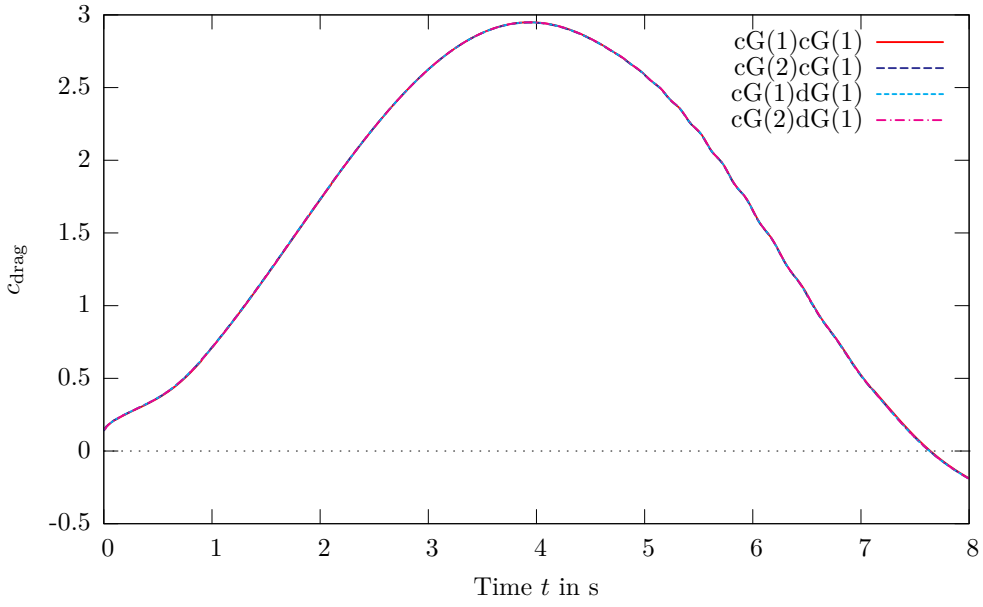


Figure 6.3. Temporal evolution of the drag-coefficient c_{drag} in the two-dimensional test case obtained by different discretizations with adaptive refinement

Even though we aim at efficiently computing the mean drag-coefficient in this example, the maximum drag-coefficient is also computed very accurately. A comparison of the results produced by the presented adaptive discretizations with the reference values of John [73] is given in Table 6.6.

Figures 6.4–6.7 show a comparison of different refinement strategies for the cG(1)cG(1), cG(2)cG(1), cG(1)dG(1), and cG(2)dG(1) discretization, respectively. We use the same

Table 6.6. Maximum drag-coefficient in the two-dimensional test case: Comparison with reference values for different discretizations

	$c^{(\mathrm{ref})}_{\mathrm{drag,max}} = 2.950921575$		$t^{(\mathrm{ref})}(c_{\mathrm{drag,max}}) = 3.93625\,\mathrm{s}$	
Discretization	$c_{\mathrm{drag,max}}$	$\left\lvert\frac{c_{\mathrm{drag,max}} - c^{(\mathrm{ref})}_{\mathrm{drag,max}}}{c^{(\mathrm{ref})}_{\mathrm{drag,max}}}\right\rvert$	$t(c_{\mathrm{drag,max}})$	$\left\lvert\frac{t(c_{\mathrm{drag,max}}) - t^{(\mathrm{ref})}(c_{\mathrm{drag,max}})}{t^{(\mathrm{ref})}(c_{\mathrm{drag,max}})}\right\rvert$
cG(1)cG(1)	2.950897238	$8.2 \cdot 10^{-6}$	3.9375 s	$3.2 \cdot 10^{-4}$
cG(2)cG(1)	2.950899555	$7.5 \cdot 10^{-6}$	3.9375 s	$3.2 \cdot 10^{-4}$
cG(1)dG(1)	2.950833347	$3.0 \cdot 10^{-5}$	3.9375 s	$3.2 \cdot 10^{-4}$
cG(2)dG(1)	2.950914600	$2.4 \cdot 10^{-6}$	3.9375 s	$3.2 \cdot 10^{-4}$

labeling as in Chapter 4:

- "uniform": We apply uniform refinement to the temporal and spatial discretization in each iteration.
- "adaptive": We apply adaptive refinement to the temporal and spatial discretization together with the proposed equilibration strategy. The spatial mesh is fixed on the whole time interval.
- "dynamic": We apply adaptive refinement to the temporal and spatial discretization together with the proposed equilibration strategy. The spatial meshes may vary from subinterval to subinterval.

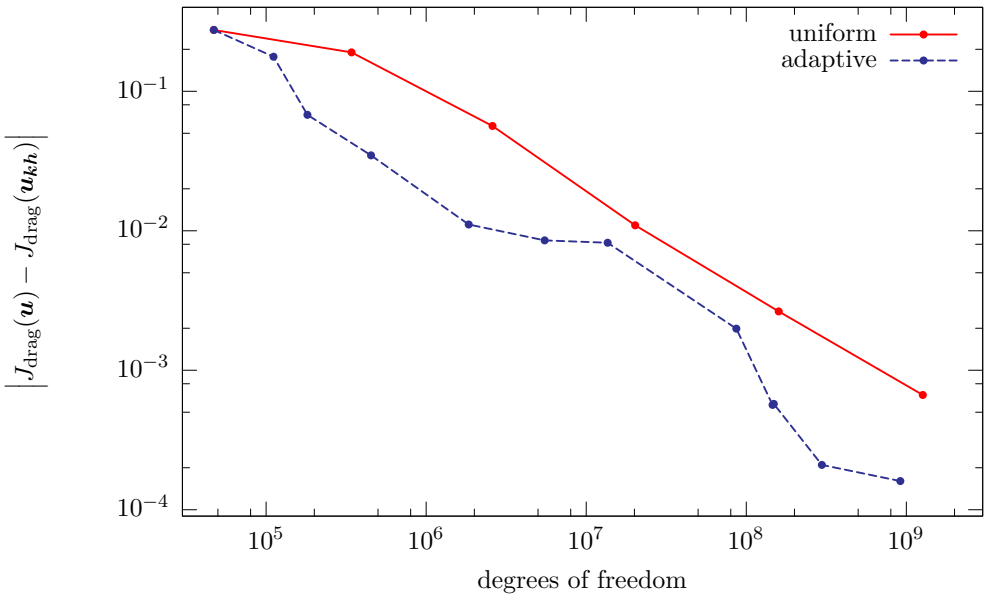


Figure 6.4. Comparison of the error $\left|J_{\mathrm{drag}}(\boldsymbol{u}) - J_{\mathrm{drag}}(\boldsymbol{u_{kh}})\right|$ in the two-dimensional test case for different refinement strategies with the cG(1)cG(1) discretization

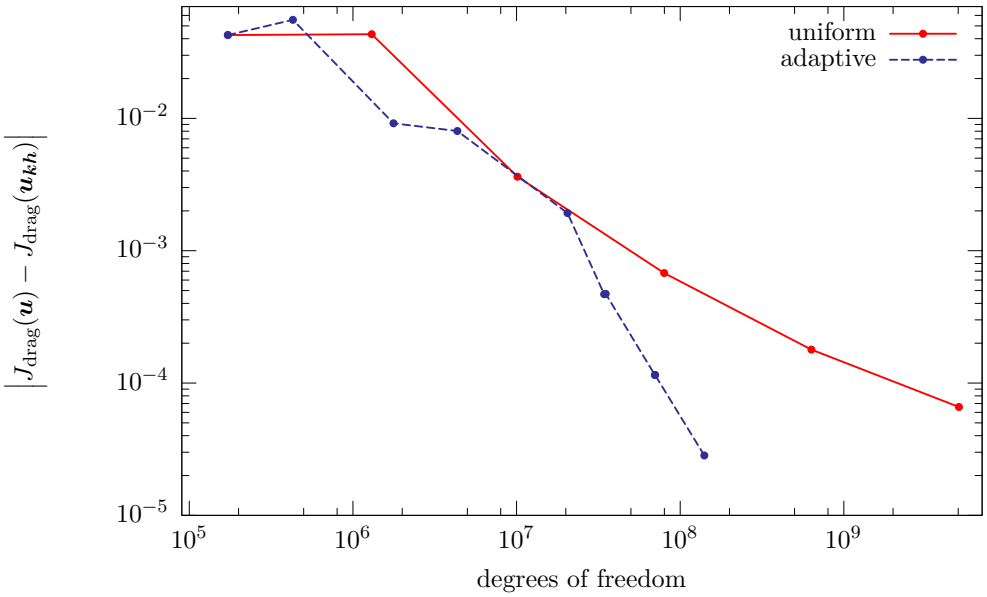


Figure 6.5. Comparison of the error $\left|J_{\mathrm{drag}}(\boldsymbol{u}) - J_{\mathrm{drag}}(\boldsymbol{u_{kh}})\right|$ in the two-dimensional test case for different refinement strategies with the cG(2)cG(1) discretization

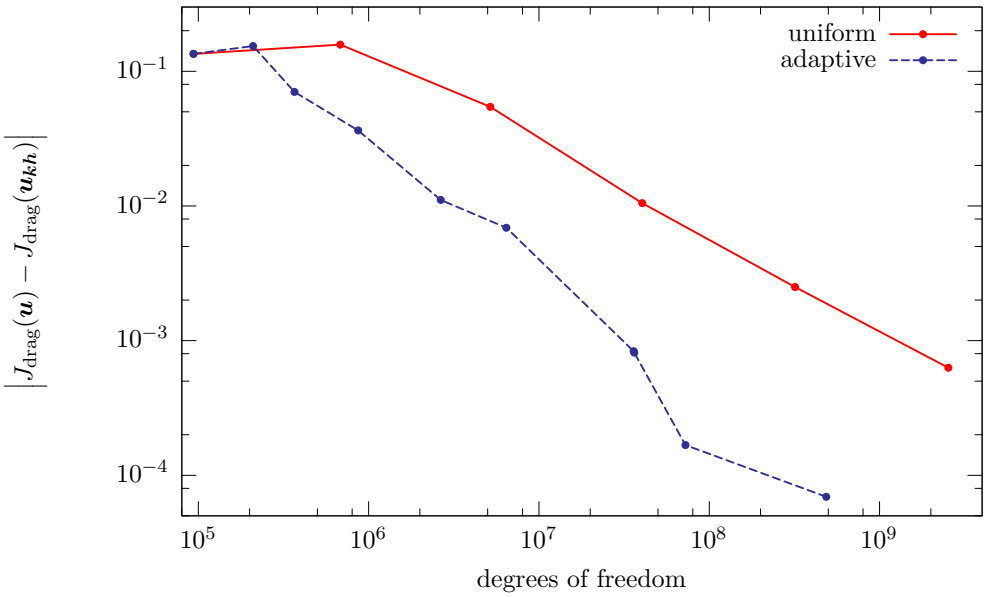


Figure 6.6. Comparison of the error $\left|J_{\mathrm{drag}}(\boldsymbol{u}) - J_{\mathrm{drag}}(\boldsymbol{u_{kh}})\right|$ in the two-dimensional test case for different refinement strategies with the cG(1)dG(1) discretization

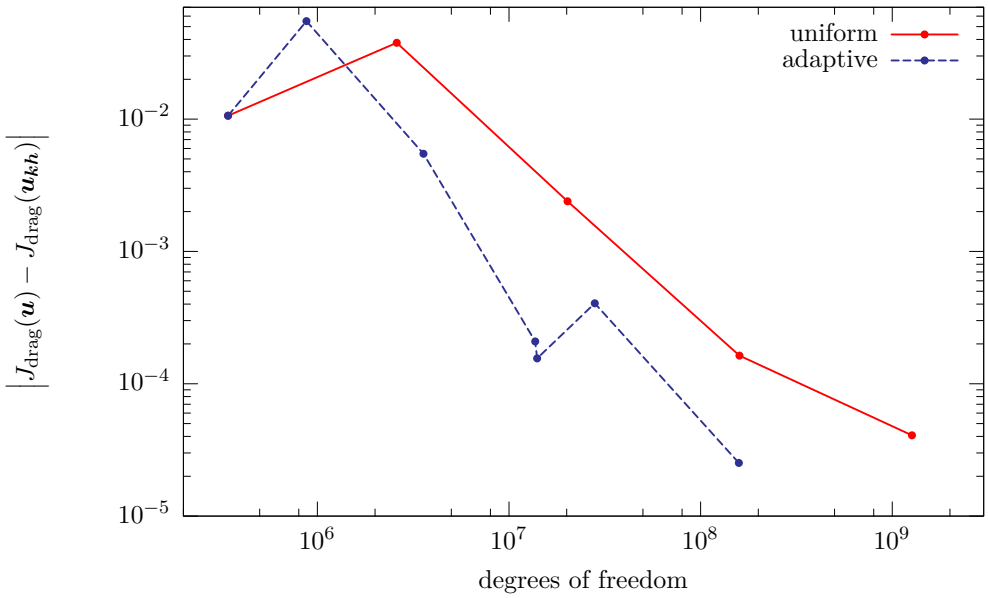


Figure 6.7. Comparison of the error $\left|J_{\mathrm{drag}}(\boldsymbol{u}) - J_{\mathrm{drag}}(\boldsymbol{u_{kh}})\right|$ in the two-dimensional test case for different refinement strategies with the cG(2)dG(1) discretization

If we compare the number of degrees of freedom needed to reach a relative error of 1 %, the required degrees of freedom can be reduced by a factor of 5–15 using adaptive refinement with equilibration instead of uniform refinement, depending on the chosen discretization. This ratio increases when aiming at higher accuracies. Using uniform refinement, we were not even able to reach an accuracy of $5 \cdot 10^{-5}$.

In Figure 6.8, the adaptive distribution of the time steps is depicted exemplarily for the cG(1)cG(1) discretization. The other discretizations lead to qualitatively similar results. The smallest time steps are obtained for $t \in [3\,\mathrm{s}, 6\,\mathrm{s}]$ where the dynamics of the flow are highest and most of the energy in concentrated in the vortices.

We do not use dynamic spatial meshes in this example for two reasons: On the one hand, recalling the results of Chapter 5, the use of dynamic meshes leads to wrong approximations of the drag- and lift-coefficient if no additional projection steps are applied each time the spatial mesh is changed. However, this would be too costly. The other reason becomes clear if we have a look at Figure 6.9 where the adaptive spatial meshes corresponding to the last lines in Tables 6.2–6.5 are presented. Of course, the meshes corresponding to cG(2) discretizations in space are much coarser than those corresponding to cG(1) discretizations. Note, however, the similarity between the meshes produced by the cG(1)cG(1) and cG(1)dG(1) (Figures 6.9(a) and 6.9(c)) as well as the cG(2)cG(1) and cG(2)dG(1) discretization (Figures 6.9(b) and 6.9(d)). Furthermore, we observe that in order to precisely determine the mean drag-coefficient it is not necessary to resolve the whole van Kármán vortex street. Only a small recirculation zone behind the obstacle is

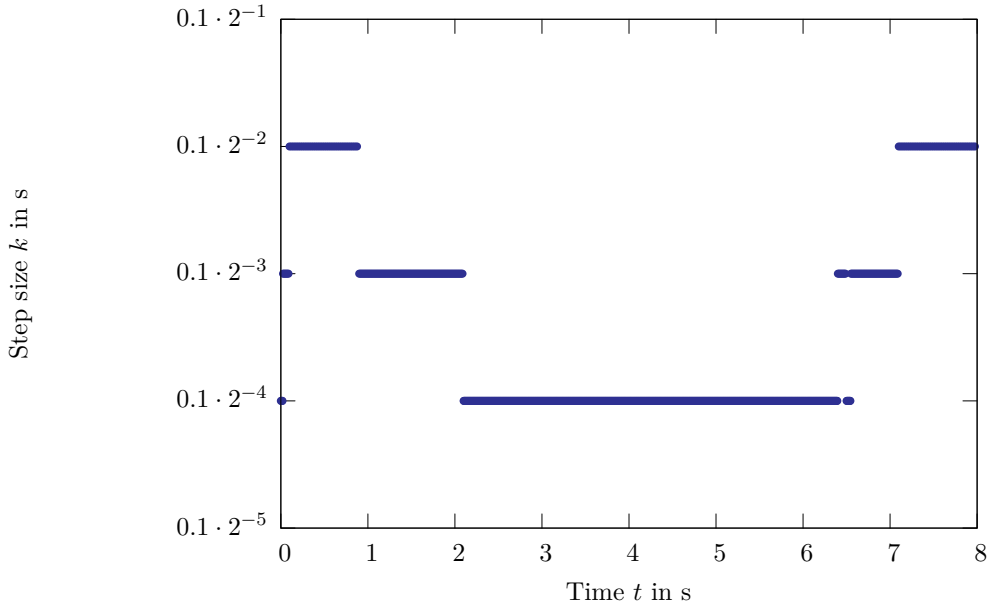
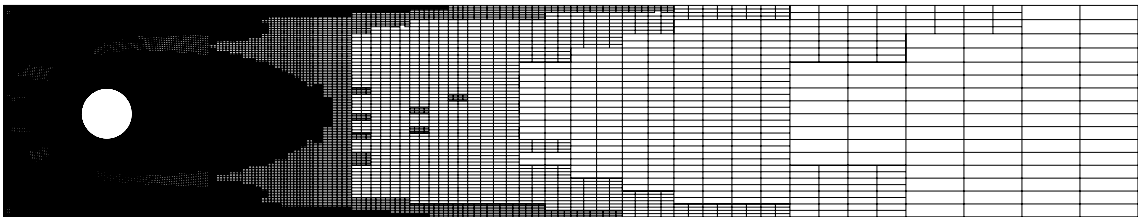


Figure 6.8. Mean drag-coefficient for the two-dimensional test case: Adaptively determined time step size k with the cG(1)cG(1) discretization

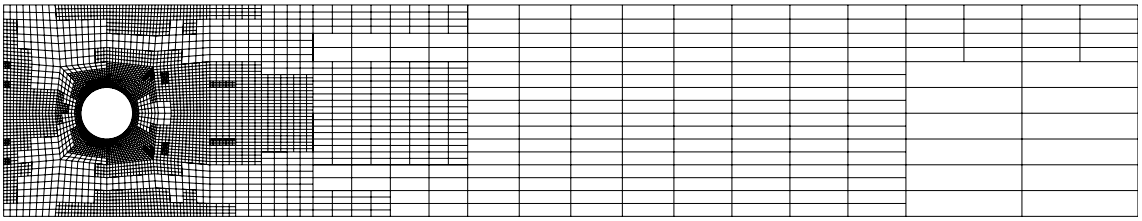
strongly refined. Since the vortices in this region develop relatively early, we may conclude that allowing dynamic meshes would not provide a further notable reduction in the degrees of freedom needed to reach the same accuracy as with adaptively refined, but fixed spatial meshes. In virtue of the additional effort on dynamic meshes due to more frequent matrix reassembling and the additional projection steps, we arrive at the conclusion that the use of dynamic spatial meshes does not make sense in this case.

To substantiate this conclusion, we again apply exemplarily the cG(1)dG(1) discretization to this example, but now allow dynamic spatial meshes. Each time the spatial mesh is changed, we use the H-projection of the old velocity as initial value on the next mesh. The results of this computation are given in Table 6.7. In Figure 6.10, we show a sequence of adaptively refined spatial meshes generated by the adaptive algorithm. These meshes correspond to the discretization described in the last line of Table 6.7 leading to an absolute error of $2 \cdot 10^{-3}$ which corresponds to an accuracy of about 0.1 %.

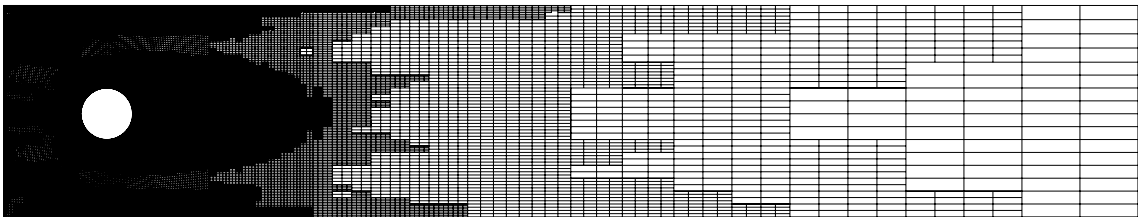
We observe that the meshes are strongly refined around the obstacle over the whole time interval. Furthermore, the meshes show strong refinement in the recirculation zone behind the obstacle for approximately $t \in [0\,\mathrm{s}, 5\,\mathrm{s}]$ as well as refinement in front of the obstacle at the beginning of the computation. Near the end of the time interval, the meshes become coarser. As already mentioned before, the meshes do not track the whole vortex street. Instead, they are kept relatively constant, except for the end of the time interval where they are coarsened. This explains why we do not benefit from using dynamic meshes. If we compare the required degrees of freedom to reach a certain accuracy, we observe that there is not much difference between fixed adaptively refined meshes and dynamic meshes,



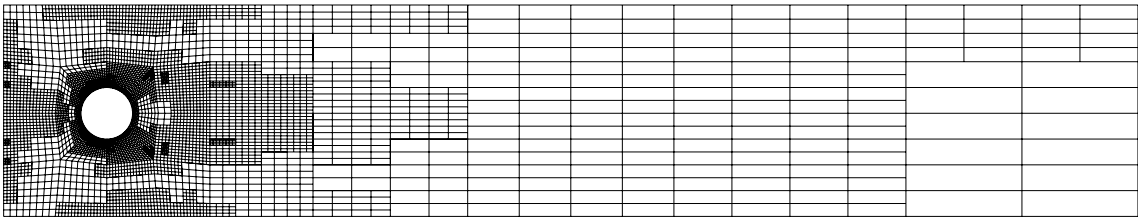
(a) cG(1)cG(1) discretization



(b) cG(2)cG(1) discretization

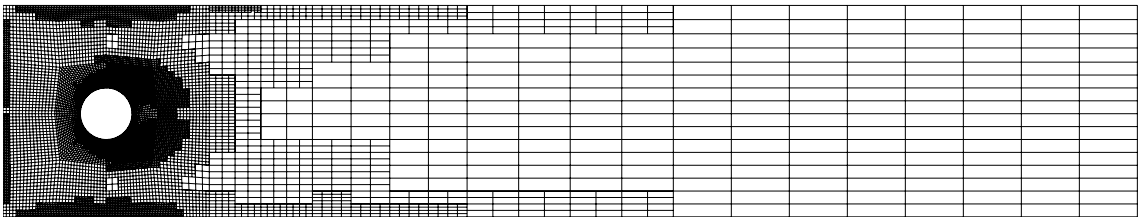


(c) cG(1)dG(1) discretization

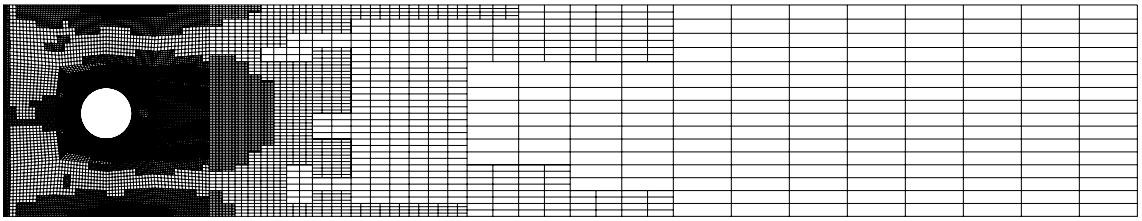


(d) cG(2)dG(1) discretization

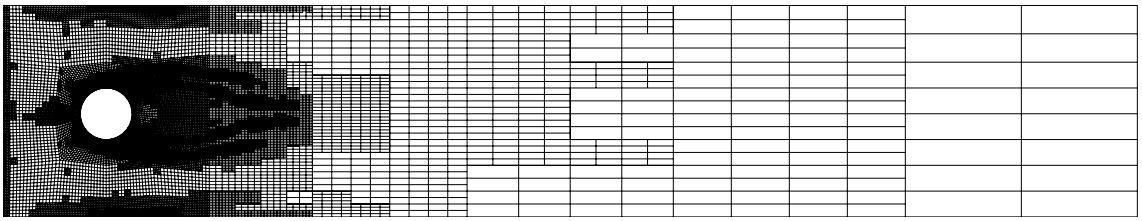
Figure 6.9. Spatial meshes for the computation of the mean drag-coefficient with different discretizations



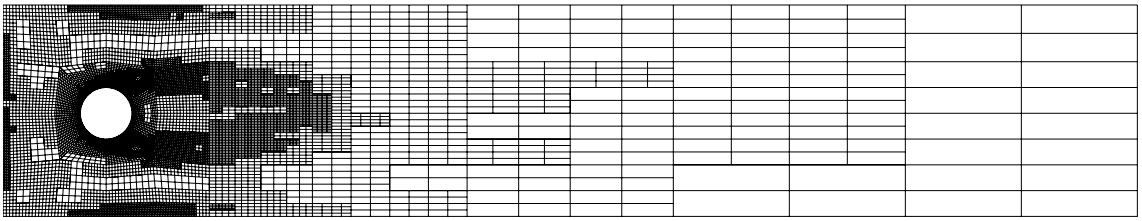
(a) $t = 1\,\mathrm{s}$



(b) $t = 2\,\mathrm{s}$

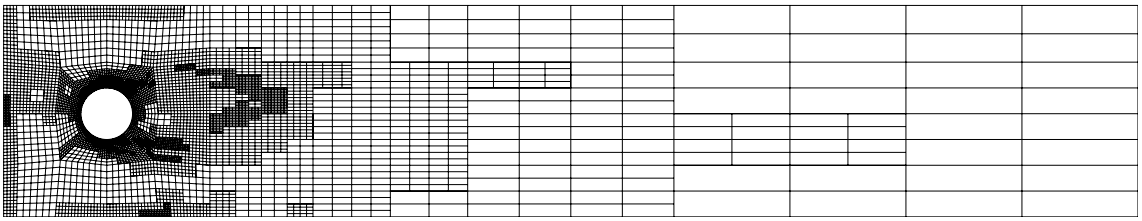


(c) $t = 3\,\mathrm{s}$

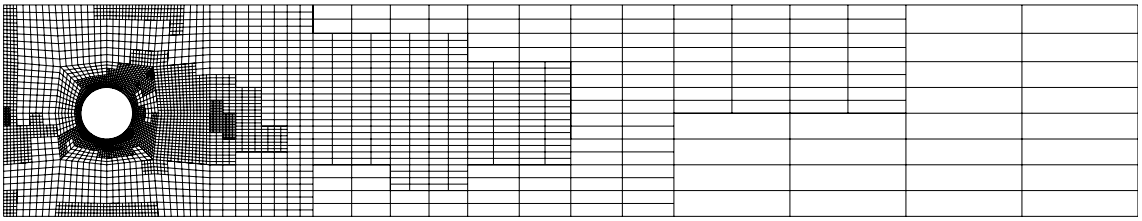


(d) $t = 4\,\mathrm{s}$

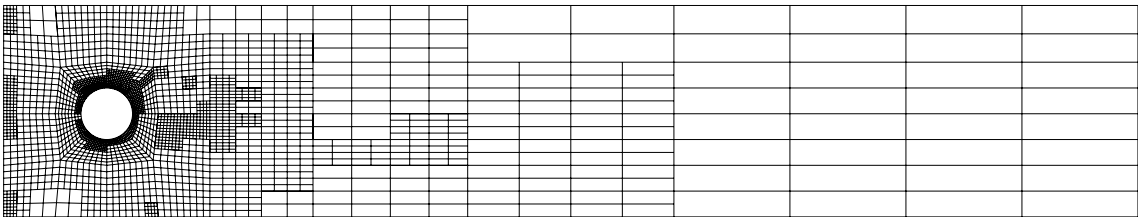
Figure 6.10. Two-dimensional test case: Dynamic meshes at different times t for the cG(1)dG(1) discretization



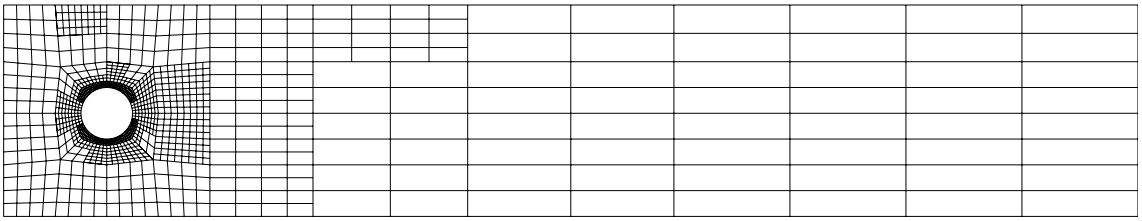
(e) $t = 5\,\mathrm{s}$



(f) $t = 6\,\mathrm{s}$



(g) $t = 7\,\mathrm{s}$



(h) $t = 8\,\mathrm{s}$

Figure 6.10. Two-dimensional test case: Dynamic meshes at different times t for the cG(1)dG(1) discretization (cont.)

Table 6.7. Mean drag-coefficient for the two-dimensional test case: Adaptive refinement with dynamic meshes and equilibration for the cG(1)dG(1) discretization

N_{tot}	$N_{\max}$	M	η_h	η_k	$J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})$	I_{eff}
93702	582	80	$-2.99 \cdot 10^{-01}$	$2.16 \cdot 10^{-04}$	$1.35 \cdot 10^{-01}$	-0.45
189414	1374	80	$-9.09 \cdot 10^{-02}$	$-1.34 \cdot 10^{-04}$	$1.54 \cdot 10^{-01}$	-1.69
340254	2994	80	$7.48 \cdot 10^{-03}$	$4.44 \cdot 10^{-03}$	$5.65 \cdot 10^{-02}$	4.74
1265994	7362	122	$2.02 \cdot 10^{-03}$	$5.74 \cdot 10^{-03}$	$1.61 \cdot 10^{-03}$	0.21
3042798	20766	134	$-1.88 \cdot 10^{-03}$	$3.56 \cdot 10^{-03}$	$1.24 \cdot 10^{-02}$	7.34
7915890	58236	150	$3.52 \cdot 10^{-03}$	$3.88 \cdot 10^{-03}$	$9.72 \cdot 10^{-03}$	1.31
40086354	163386	300	$1.16 \cdot 10^{-03}$	$6.72 \cdot 10^{-04}$	$2.08 \cdot 10^{-03}$	1.13

see Figure 6.11. Actually, the use of dynamic meshes leads to a slightly higher number of degrees of freedom required to obtain the same accuracy as with adaptively refined, but fixed spatial meshes.

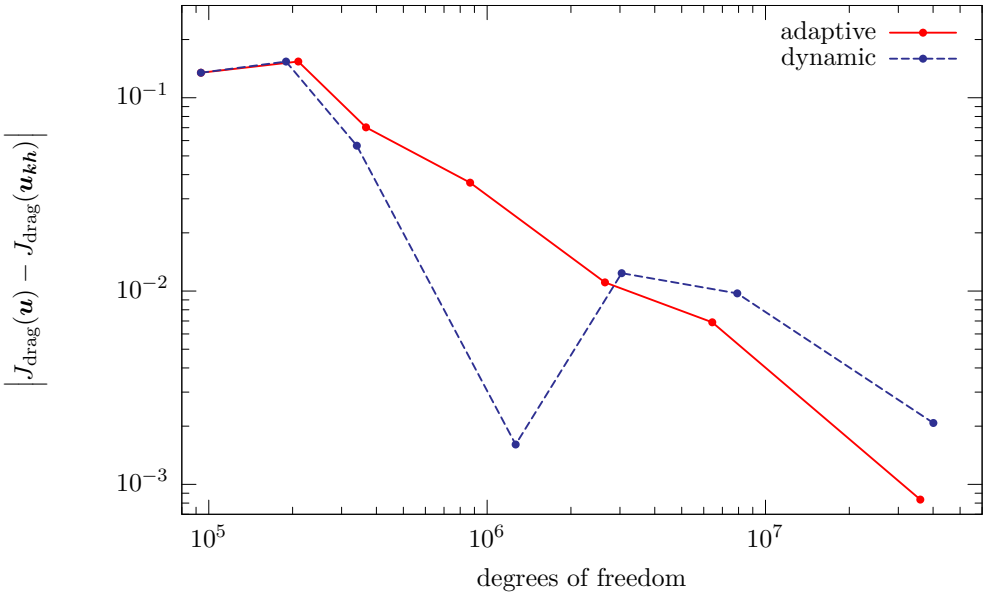


Figure 6.11. Comparison of the error $\left|J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})\right|$ in the two-dimensional test case for adaptive refinement without and with dynamic meshes in the case of a cG(1)dG(1) discretization

Before we focus on the mean lift-coefficient, we like to point out another remarkable feature of the developed error estimator. To this end, we consider the following quantity of interest:

$$J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u}) = -\int_{I_{\text{sub}}} \left\{ (\partial_t \boldsymbol{v}, \hat{\boldsymbol{\psi}}_{\mathbf{drag}}^{(\mathbf{sub})}) + \bar{a}(\boldsymbol{u})(\hat{\boldsymbol{\varphi}}_{\mathbf{drag}}^{(\mathbf{sub})}) \right\} \mathrm{d}t$$

with $I_{\text{sub}} = [7\,\text{s}, 8\,\text{s}]$ where $\hat{\boldsymbol{\varphi}}_{\mathbf{drag}}^{(\mathbf{sub})} = (\hat{\boldsymbol{\psi}}_{\mathbf{drag}}^{(\mathbf{sub})}, 0)^T$ fulfills

$$\hat{\boldsymbol{\psi}}_{\mathbf{drag}}^{(\mathbf{sub})}\Big|_S = \begin{pmatrix} |I_{\text{sub}}|^{-1} C \\ 0 \end{pmatrix}, \quad \hat{\boldsymbol{\psi}}_{\mathbf{drag}}^{(\mathbf{sub})}\Big|_{\partial\Omega \setminus S} = \mathbf{0}.$$

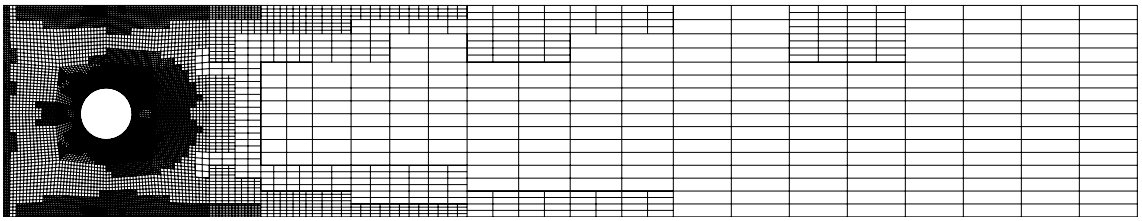
In this way, we compute the mean drag-coefficient not over the whole time interval, but only on the subinterval $I_{\text{sub}} = [7\,\text{s}, 8\,\text{s}]$. Here, we also apply exemplarily the cG(1)dG(1) discretization. Allowing dynamically changing meshes and adaptive refinement of the temporal discretization yields the results presented in Table 6.8. As reference value, an extrapolated value of $J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u}) = 0.11936$ is used.

Table 6.8. Mean drag-coefficient on I_{sub} for the two-dimensional test case: Adaptive refinement with dynamic meshes and equilibration for the cG(1)dG(1) discretization

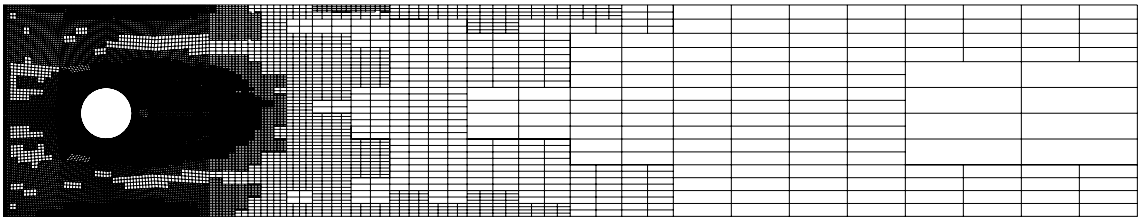
N_{tot}	$N_{\max}$	M	η_h	η_k	$J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u}) - J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u_{kh}})$	I_{eff}
93702	582	80	$-1.16 \cdot 10^{-02}$	$2.69 \cdot 10^{-03}$	$9.52 \cdot 10^{-02}$	-10.72
127470	1458	80	$3.02 \cdot 10^{-02}$	$6.01 \cdot 10^{-03}$	$6.15 \cdot 10^{-02}$	1.70
279822	2772	80	$1.13 \cdot 10^{-02}$	$7.23 \cdot 10^{-03}$	$4.02 \cdot 10^{-02}$	2.17
771474	5046	114	$5.24 \cdot 10^{-03}$	$-7.86 \cdot 10^{-05}$	$2.41 \cdot 10^{-02}$	4.67
1316082	11496	114	$5.12 \cdot 10^{-03}$	$4.57 \cdot 10^{-03}$	$1.89 \cdot 10^{-02}$	1.95
3911514	28878	146	$1.40 \cdot 10^{-03}$	$7.31 \cdot 10^{-04}$	$7.47 \cdot 10^{-03}$	3.51
9922530	77826	156	$3.43 \cdot 10^{-04}$	$5.78 \cdot 10^{-04}$	$3.97 \cdot 10^{-03}$	4.31
29345742	229542	162	$1.01 \cdot 10^{-04}$	$4.85 \cdot 10^{-04}$	$2.16 \cdot 10^{-03}$	3.69
30976470	229542	176	$1.07 \cdot 10^{-04}$	$3.11 \cdot 10^{-04}$	$1.25 \cdot 10^{-03}$	2.98

A sequence of dynamic spatial meshes for different times t produced by the adaptive algorithm is depicted in Figure 6.12. These meshes correspond to the discretization described in the last line of Table 6.8. One might suppose that only meshes on the subinterval I_{sub} are strongly refined because the chosen functional only acts here. However, in order to capture the correct drag-coefficient on I_{sub}, also the spatial meshes at the beginning of the time interval have to be refined because there a bifurcation occurs which mainly determines the overall flow behavior at later times. If the flow at this early time is not resolved correctly, the flow pattern and hence the drag-coefficient will also not be determined correctly for $t \in I_{\text{sub}}$. Due to the sensitivity information available through the dual solution, the application of the developed a posteriori error estimator thus leads to refinement also at early times. If we compare the corresponding meshes with those of Figure 6.10, we actually observe not much difference until approximately $t = 5\,\text{s}$. However, the meshes at the end of the time interval are stronger refined because errors occurring there now have a larger influence on the mean drag-coefficient over I_{sub} than on the mean drag-coefficient over the whole time interval.

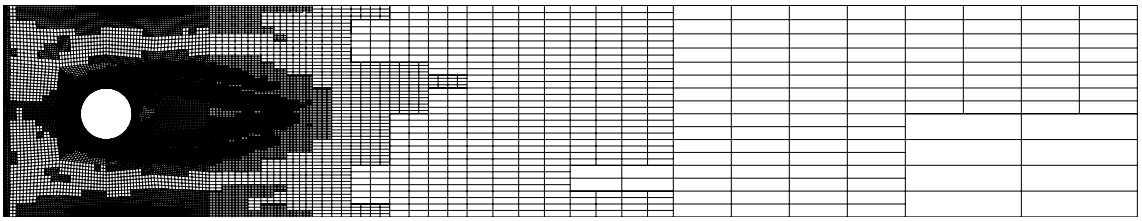
Thus, it becomes clear that the use of dynamic meshes again does not make sense for computing the mean drag-coefficient on I_{sub}. This is confirmed by a computation without dynamic meshes whose results are given in Table 6.9 and Figure 6.13.



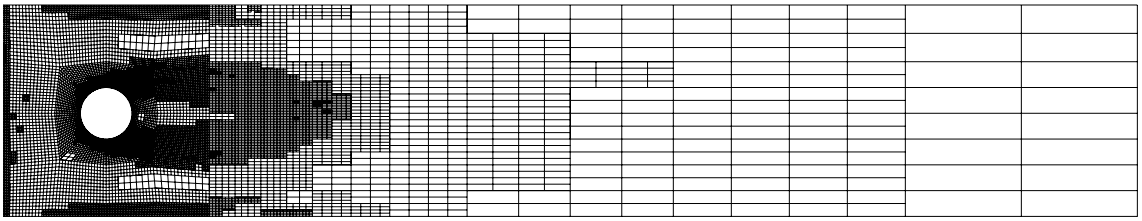
(a) $t = 1\,\mathrm{s}$



(b) $t = 2\,\mathrm{s}$

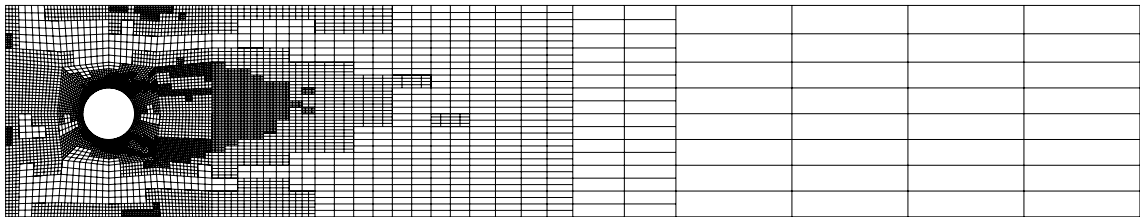


(c) $t = 3\,\mathrm{s}$

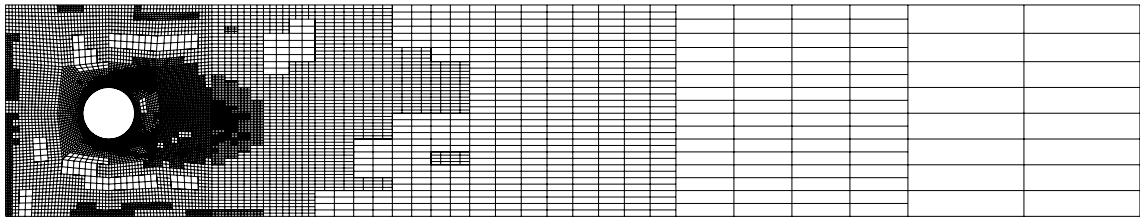


(d) $t = 4\,\mathrm{s}$

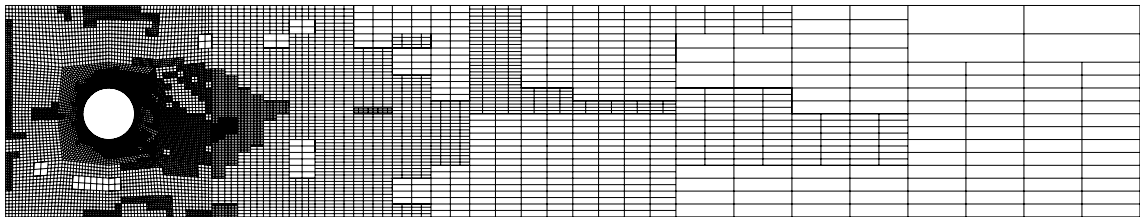
Figure 6.12. Mean drag-coefficient over subinterval: Dynamic meshes at different times t for the cG(1)dG(1) discretization



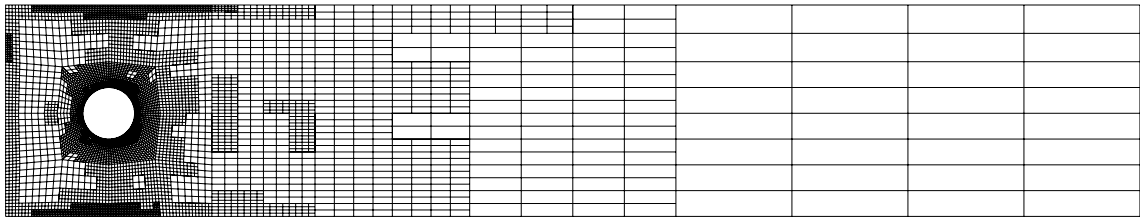
(e) $t = 5\,\mathrm{s}$



(f) $t = 6\,\mathrm{s}$



(g) $t = 7\,\mathrm{s}$



(h) $t = 8\,\mathrm{s}$

Figure 6.12. Mean drag-coefficient over subinterval: Dynamic meshes at different times t for the cG(1)dG(1) discretization (cont.)

Table 6.9. Mean drag-coefficient on I_{sub} for the two-dimensional test case: Adaptive refinement with equilibration for the cG(1)dG(1) discretization

N	M	η_h	η_k	$J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u}) - J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u_{kh}})$	I_{eff}
582	80	$-1.16 \cdot 10^{-02}$	$2.69 \cdot 10^{-03}$	$9.52 \cdot 10^{-02}$	-10.72
1374	80	$-2.07 \cdot 10^{-03}$	$2.98 \cdot 10^{-03}$	$6.86 \cdot 10^{-02}$	75.04
2712	82	$7.83 \cdot 10^{-03}$	$9.51 \cdot 10^{-03}$	$3.31 \cdot 10^{-02}$	1.91
6822	122	$1.69 \cdot 10^{-03}$	$2.33 \cdot 10^{-03}$	$7.45 \cdot 10^{-03}$	1.85
17454	136	$1.90 \cdot 10^{-04}$	$3.30 \cdot 10^{-04}$	$3.81 \cdot 10^{-03}$	7.31
58410	146	$8.54 \cdot 10^{-05}$	$5.57 \cdot 10^{-04}$	$2.29 \cdot 10^{-03}$	3.57
58410	168	$8.80 \cdot 10^{-05}$	$3.02 \cdot 10^{-04}$	$1.28 \cdot 10^{-03}$	3.29
58410	176	$9.40 \cdot 10^{-05}$	$2.71 \cdot 10^{-04}$	$8.85 \cdot 10^{-04}$	2.43
205764	184	$3.54 \cdot 10^{-05}$	$2.80 \cdot 10^{-04}$	$5.76 \cdot 10^{-04}$	1.83
205764	278	$4.53 \cdot 10^{-05}$	$1.20 \cdot 10^{-04}$	$1.91 \cdot 10^{-04}$	1.15

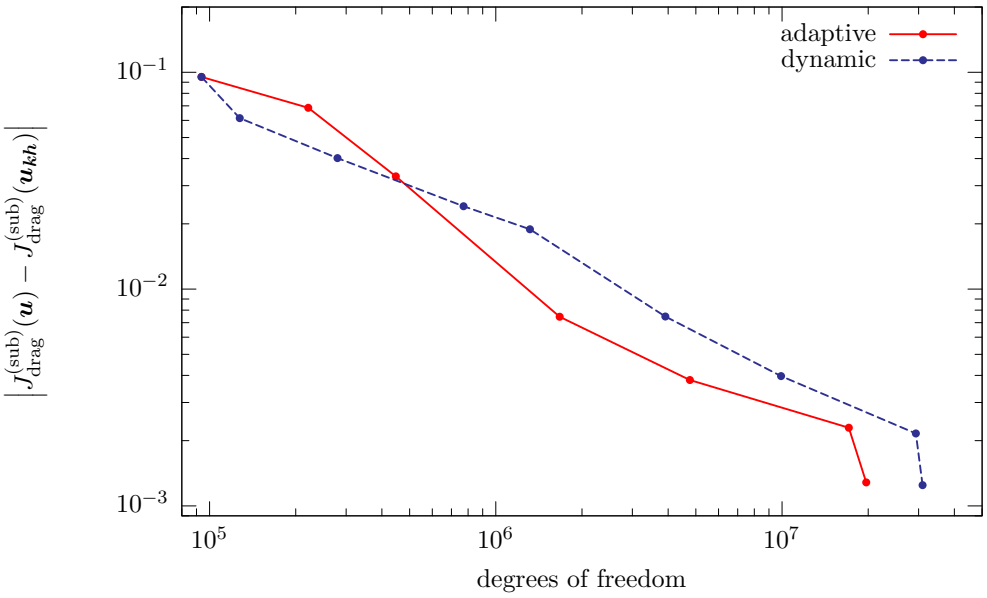


Figure 6.13. Comparison of the error $\left|J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u}) - J_{\text{drag}}^{(\text{sub})}(\boldsymbol{u_{kh}})\right|$ in the two-dimensional test case for adaptive refinement without and with dynamic meshes in the case of a cG(1)dG(1) discretization

Mean lift-coefficient

We now turn to the mean lift-coefficient. We proceed in the same way as for the mean drag-coefficient and apply the cG(2)dG(1) discretization with uniform spatial and temporal refinement. The obtained values are given in Table 6.10.

Table 6.10. Mean lift-coefficient obtained with the cG(2)dG(1) discretization and uniform refinement

N	M	$J_{\text{lift}}(\boldsymbol{u_{kh}})$
8088	80	$-1.2531245 \cdot 10^{-3}$
31536	160	$-1.1119757 \cdot 10^{-2}$
124512	320	$-1.0361390 \cdot 10^{-2}$
494784	640	$-1.0245390 \cdot 10^{-2}$
extrapolated		$-1.0206723 \cdot 10^{-2}$

Now, we employ the cG(2)cG(1) and the cG(2)dG(1) discretization in combination with the adaptive algorithm. The results of these computations using adaptively refined spatial meshes which are kept constant over the whole time interval $I = (0\,\text{s}, 8\,\text{s})$ are shown in Tables 6.11 and 6.12. By comparing the results, we observe that only the first three digits of the extrapolated value $-1.0206723 \cdot 10^{-2}$ of the mean lift-coefficient seem to be correct. The "true" mean lift-coefficient is rather given by $J_{\text{lift}}(\boldsymbol{u}) \approx -1.023 \cdot 10^{-2}$. This shows that the computation of the mean lift-coefficient is a much harder task than the computation of the mean drag-coefficient. In Tables 6.11 and 6.12, the value $J_{\text{lift}}(\boldsymbol{u}) = -1.023 \cdot 10^{-2}$ is used as reference value.

Table 6.11. Mean lift-coefficient for the two-dimensional test case: Adaptive refinement with equilibration for the cG(2)cG(1) discretization

N	M	η_h	η_k	$J_{\text{lift}}(\boldsymbol{u}) - J_{\text{lift}}(\boldsymbol{u_{kh}})$	I_{eff}
8088	80	$-6.16 \cdot 10^{-04}$	$-3.98 \cdot 10^{-03}$	$3.71 \cdot 10^{-04}$	-0.08
8088	160	$-1.18 \cdot 10^{-03}$	$7.72 \cdot 10^{-03}$	$-8.38 \cdot 10^{-03}$	-1.28
8088	254	$-2.95 \cdot 10^{-04}$	$-3.81 \cdot 10^{-03}$	$-1.54 \cdot 10^{-03}$	0.37
8088	508	$-7.06 \cdot 10^{-04}$	$-1.80 \cdot 10^{-05}$	$5.82 \cdot 10^{-04}$	-0.80
13572	508	$-4.76 \cdot 10^{-05}$	$4.32 \cdot 10^{-04}$	$1.22 \cdot 10^{-03}$	3.17
13572	516	$-4.66 \cdot 10^{-05}$	$4.30 \cdot 10^{-04}$	$1.20 \cdot 10^{-03}$	3.13
13572	1032	$-2.23 \cdot 10^{-05}$	$1.91 \cdot 10^{-04}$	$7.31 \cdot 10^{-04}$	4.33
13572	1036	$-1.88 \cdot 10^{-05}$	$1.90 \cdot 10^{-04}$	$7.29 \cdot 10^{-04}$	4.26
13572	2072	$-2.58 \cdot 10^{-05}$	$5.41 \cdot 10^{-05}$	$5.56 \cdot 10^{-04}$	19.60
32328	2084	$2.08 \cdot 10^{-05}$	$6.43 \cdot 10^{-05}$	$1.70 \cdot 10^{-04}$	2.00
32328	4168	$1.48 \cdot 10^{-05}$	$1.65 \cdot 10^{-05}$	$1.11 \cdot 10^{-04}$	3.53
84240	4172	$2.21 \cdot 10^{-06}$	$1.67 \cdot 10^{-05}$	$2.78 \cdot 10^{-05}$	1.47
84240	8344	$9.56 \cdot 10^{-07}$	$4.22 \cdot 10^{-06}$	$1.29 \cdot 10^{-05}$	2.48

Table 6.12. Mean lift-coefficient for the two-dimensional test case: Adaptive refinement with equilibration for the cG(2)dG(1) discretization

N	M	η_h	η_k	$J_{\text{lift}}(\boldsymbol{u}) - J_{\text{lift}}(\boldsymbol{u_{kh}})$	I_{eff}
8088	80	$-7.57 \cdot 10^{-04}$	$-9.83 \cdot 10^{-03}$	$-8.98 \cdot 10^{-03}$	0.85
8088	124	$-7.79 \cdot 10^{-04}$	$-1.00 \cdot 10^{-03}$	$-6.26 \cdot 10^{-05}$	0.04
15996	134	$-6.69 \cdot 10^{-05}$	$6.47 \cdot 10^{-04}$	$9.22 \cdot 10^{-04}$	1.59
15996	136	$-6.57 \cdot 10^{-05}$	$6.42 \cdot 10^{-04}$	$8.84 \cdot 10^{-04}$	1.53
15996	272	$-4.01 \cdot 10^{-05}$	$1.08 \cdot 10^{-04}$	$1.94 \cdot 10^{-04}$	2.88
15996	296	$-4.40 \cdot 10^{-05}$	$8.50 \cdot 10^{-05}$	$1.65 \cdot 10^{-04}$	4.03
35520	314	$2.93 \cdot 10^{-05}$	$7.43 \cdot 10^{-05}$	$1.55 \cdot 10^{-04}$	1.49
35520	320	$2.94 \cdot 10^{-05}$	$7.05 \cdot 10^{-05}$	$1.51 \cdot 10^{-04}$	1.51
89748	492	$4.64 \cdot 10^{-06}$	$1.66 \cdot 10^{-05}$	$3.37 \cdot 10^{-05}$	1.59
89748	602	$4.12 \cdot 10^{-06}$	$8.92 \cdot 10^{-06}$	$2.53 \cdot 10^{-05}$	1.94
268248	686	$3.00 \cdot 10^{-07}$	$6.28 \cdot 10^{-06}$	$6.43 \cdot 10^{-06}$	0.98
268248	1004	$3.71 \cdot 10^{-07}$	$1.93 \cdot 10^{-06}$	$2.24 \cdot 10^{-06}$	0.97

For the mean lift-coefficient, we observe that the temporal discretization error is more dominant than for the mean drag-coefficient, especially for the cG(1) time discretization. The error estimator again yields results which quantitatively assess the true discretization error quite accurately.

In Figure 6.14, we show the temporal evolution of the lift-coefficient c_{lift}. As in the previous section, these values correspond to the finest adaptive discretizations described in the last lines of Tables 6.11 and 6.12 with a relative error in the mean lift-coefficient of less than $2 \cdot 10^{-3}$. This corresponds to an absolute error of less than $2.046 \cdot 10^{-5}$. Actually, for the cG(2)dG(1) discretization, these values correspond to a relative error of approximately $2 \cdot 10^{-4}$. Both curves again match very well. However, we note a slight phase shift between the different discretizations.

We also compare the maximum lift-coefficient computed by our adaptive discretizations with the reference values of John [73]. Even though the adaptive discretizations were chosen in order to efficiently compute the mean lift-coefficient, the maximum lift-coefficient is also assessed quite accurately as Table 6.13 shows. However, note that although the cG(2)dG(1) leads to a smaller error for the mean lift-coefficient, the cG(2)cG(1) yields a more accurate maximum lift-coefficient.

Table 6.13. Maximum lift-coefficient in the two-dimensional test case: Comparison with reference values for different discretizations

	$c_{\text{lift,max}}^{(\text{ref})} = 0.47795$		$t^{(\text{ref})}(c_{\text{lift,max}}) = 5.693125\,\text{s}$	
Discretization	$c_{\text{lift,max}}$	$\left\lvert \frac{c_{\text{lift,max}} - c_{\text{lift,max}}^{(\text{ref})}}{c_{\text{lift,max}}^{(\text{ref})}} \right\rvert$	$t(c_{\text{lift,max}})$	$\left\lvert \frac{t(c_{\text{lift,max}}) - t^{(\text{ref})}(c_{\text{lift,max}})}{t^{(\text{ref})}(c_{\text{lift,max}})} \right\rvert$
cG(2)cG(1)	0.47807	$2.6 \cdot 10^{-4}$	5.69375 s	$1.1 \cdot 10^{-4}$
cG(2)dG(1)	0.47702	$1.9 \cdot 10^{-3}$	5.70000 s	$1.2 \cdot 10^{-3}$

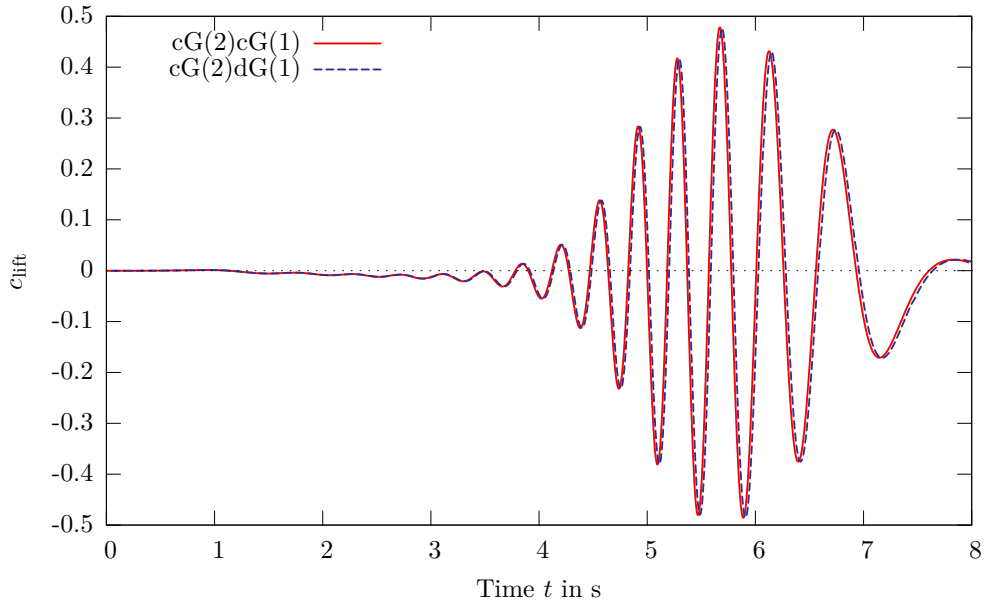


Figure 6.14. Temporal evolution of the lift-coefficient c_{lift} in the two-dimensional test case obtained by different discretizations with adaptive refinement

Figures 6.15 and 6.16 show a comparison of different refinement strategies for the cG(2)cG(1) and the cG(2)dG(1) discretization, respectively. Here, the labeling is again the same as in Chapter 4. Concerning the computational costs, we observe a reduction in the degrees of freedom needed to reach a relative error of 1 % by a factor of approximately 5 for the cG(2)dG(1) discretization. For the cG(2)cG(1) discretization, the required degrees of freedom are reduced by a factor of 50.

In Figure 6.17, the adaptive distribution of the time steps is depicted exemplarily for the cG(2)dG(1) discretization. The other discretization leads to qualitatively similar results. The smallest time steps are again obtained for $t \in [3\,\text{s}, 6\,\text{s}]$ where the dynamics of the flow are highest and most of the energy is concentrated in the vortices.

Figure 6.18 shows the adaptive spatial meshes corresponding to the last lines in Tables 6.11 and 6.12. We observe that the meshes are strongly refined only around the obstacle as well as in a small zone behind the obstacle. Hence, we can conclude as for the computation of the mean drag-coefficient that the use of dynamic meshes will not lead to a notable benefit in this case, either.

6.1.2 Three-dimensional test case

For the three-dimensional case, flow around an obstacle with square cross-section is considered. The geometry is depicted in Figure 6.19.

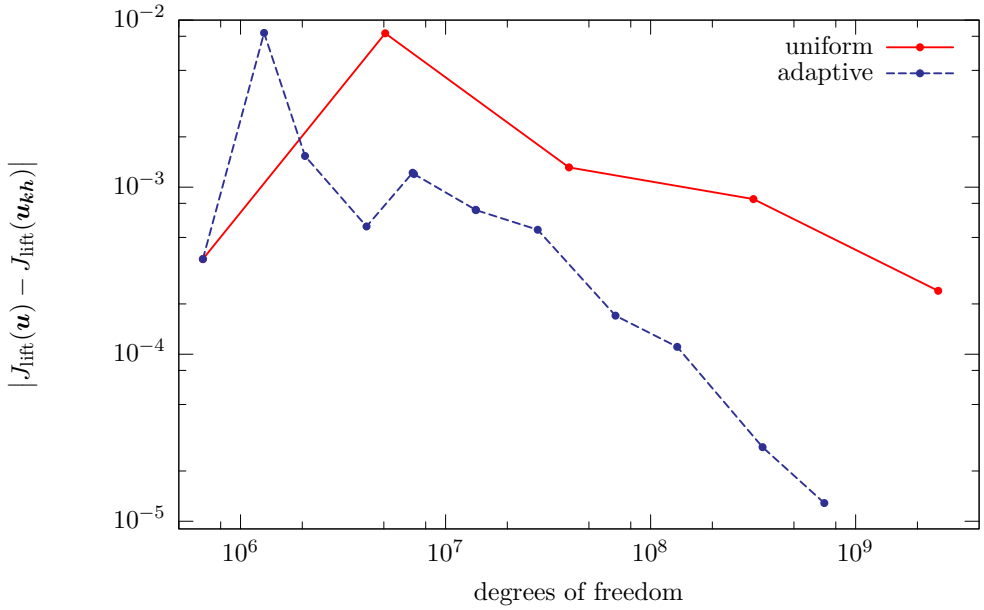


Figure 6.15. Comparison of the error $|J_{\text{lift}}(\boldsymbol{u}) - J_{\text{lift}}(\boldsymbol{u_{kh}})|$ in the two-dimensional test case for different refinement strategies with the cG(2)cG(1) discretization

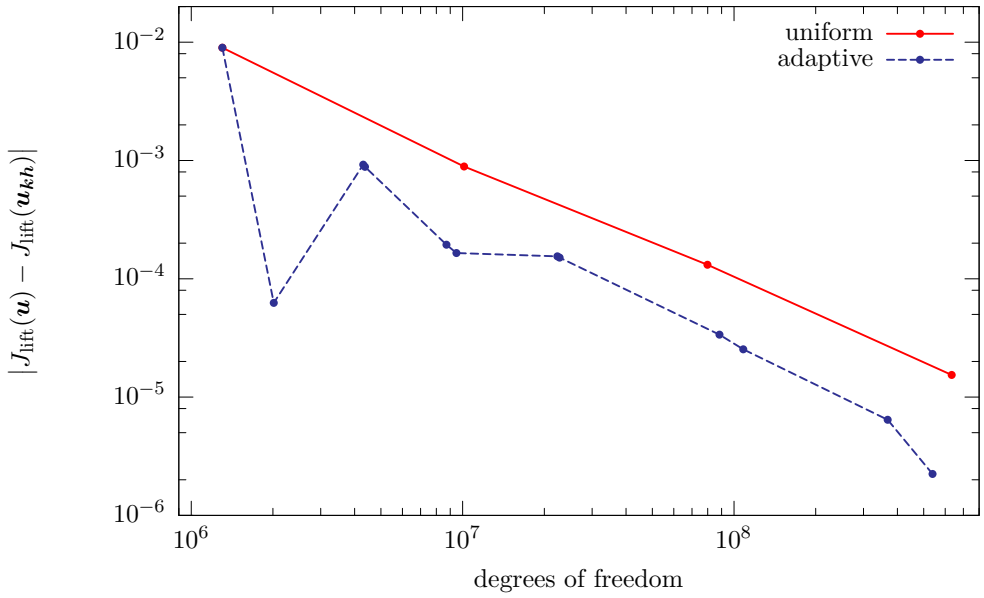


Figure 6.16. Comparison of the error $|J_{\text{lift}}(\boldsymbol{u}) - J_{\text{lift}}(\boldsymbol{u_{kh}})|$ in the two-dimensional test case for different refinement strategies with the cG(2)dG(1) discretization

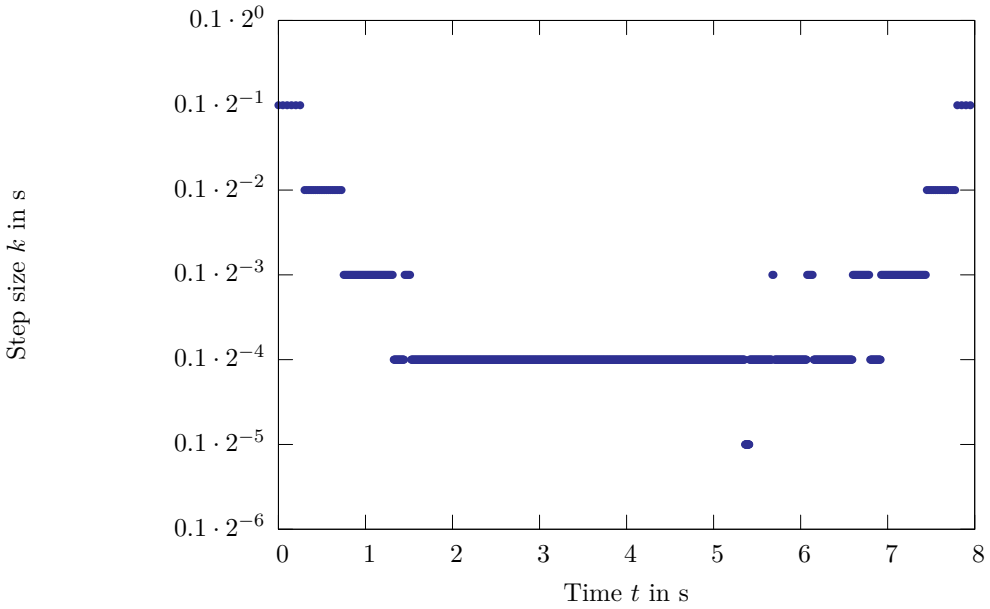
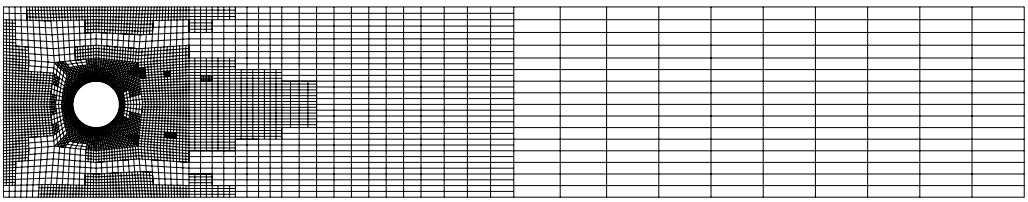


Figure 6.17. Mean lift-coefficient for the two-dimensional test case: Adaptively determined time step size k with the cG(2)dG(1) discretization



(a) cG(2)cG(1) discretization

(b) cG(2)dG(1) discretization

Figure 6.18. Spatial meshes for the computation of the mean lift-coefficient with different discretizations

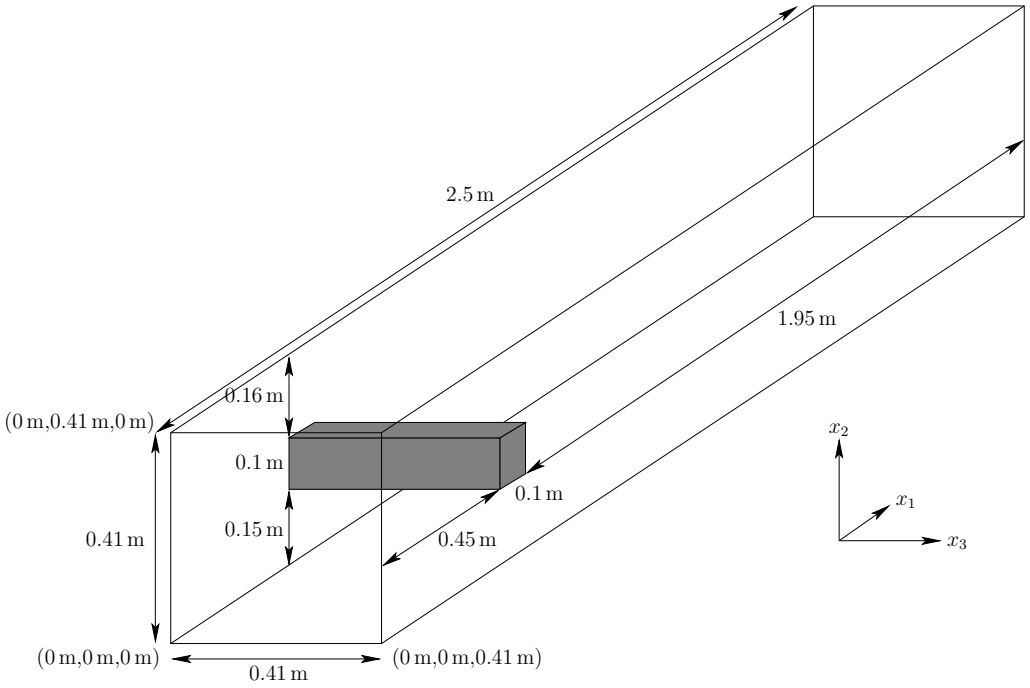


Figure 6.19. Geometry for the three-dimensional benchmark configuration

The inflow condition on the front plane is prescribed as

$$\begin{aligned} v_1(t,\boldsymbol{x}) &= \frac{36\sin(\frac{\pi t}{8\,\mathrm{s}})}{(0.41\,\mathrm{m})^4} x_2 x_3 (0.41\,\mathrm{m} - x_2)(0.41\,\mathrm{m} - x_3)\,\mathrm{m\,s^{-1}}, \\ v_2(t,\boldsymbol{x}) &= 0\,\mathrm{m\,s^{-1}}, \\ v_3(t,\boldsymbol{x}) &= 0\,\mathrm{m\,s^{-1}}. \end{aligned}$$

On the outflow boundary on the back plane of the computational domain, we again apply natural boundary conditions. On all other boundaries, no-slip Dirichlet boundary conditions are prescribed. The final time is set to $T = 8\,\mathrm{s}$. As for the two-dimensional case, this setting leads to a Reynolds number $\mathrm{Re}(t) = \frac{\bar{U}(t)D}{\nu}$ based on the mean inflow velocity

$$\bar{U}(t) = \frac{4}{9} v_1(t, 0\,\mathrm{m}, 0.205\,\mathrm{m}, 0.205\,\mathrm{m}) = \sin\Big(\frac{\pi t}{8\,\mathrm{s}}\Big)\,\mathrm{m\,s^{-1}}$$

and the side length of the obstacle $D = 0.1\,\mathrm{m}$ of $0 \leq \mathrm{Re}(t) \leq 100$ for $0\,\mathrm{s} \leq t \leq 8\,\mathrm{s}$.

The drag-coefficient is given as

$$c_{\mathrm{drag}} = C \int_S (\rho\nu \partial_{\boldsymbol{n}}(\boldsymbol{v}\cdot\boldsymbol{t}) n_2 - p n_1)\,\mathrm{d}o,$$

where S denotes the surface of the obstacle, $(\boldsymbol{v}\cdot\boldsymbol{t})$ is the tangential component of $\boldsymbol{v}$ for the tangent vector $\boldsymbol{t} = (n_2, -n_1, 0)^T$, and the constant C is chosen as

$$C = \frac{2}{\rho \bar{U}^2 D H}$$

with the height and width $H = 0.41\,\mathrm{m}$ of the channel. Hence, the quantity of interest is given as in the two-dimensional case by

$$\tilde{J}_{\mathrm{drag}}(\boldsymbol{u}) = \frac{1}{|I|} \int_I c_{\mathrm{drag}} \, \mathrm{d}t.$$

Of course, this value can also be expressed by

$$J_{\mathrm{drag}}(\boldsymbol{u}) = -\int_I (\partial_t \boldsymbol{v}, \hat{\boldsymbol{\psi}}_{\mathbf{drag}}) \, \mathrm{d}t - a(\boldsymbol{u})(\hat{\boldsymbol{\varphi}}_{\mathbf{drag}})$$

with $\hat{\boldsymbol{\varphi}}_{\mathbf{drag}} = (\hat{\boldsymbol{\psi}}_{\mathbf{drag}}, 0)^T$ fulfilling

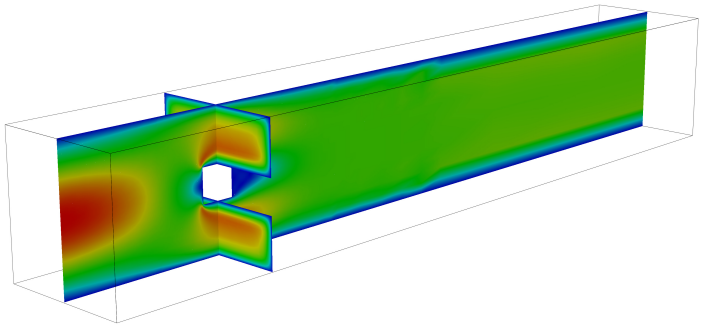
$$\hat{\boldsymbol{\psi}}_{\mathbf{drag}}\big|_S = \begin{pmatrix} |I|^{-1} C \\ 0 \\ 0 \end{pmatrix}, \quad \hat{\boldsymbol{\psi}}_{\mathbf{drag}}\big|_{\partial\Omega \setminus S} = \mathbf{0}.$$

In Figure 6.20, we show the temporal development of the flow by visualizing $|\boldsymbol{v}|$ in the planes $x_1 = 0.5\,\mathrm{m}$ and $x_3 = 0.205\,\mathrm{m}$. Compared to the two-dimensional case, the three-dimensional flow shows much less dynamics. It tends to become almost stationary, at least for $t \in (4\,\mathrm{s}, 6\,\mathrm{s})$.

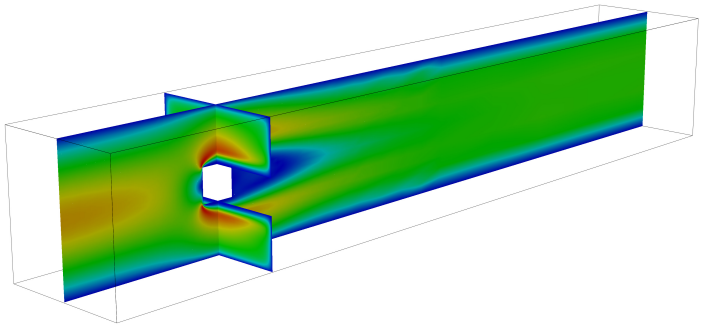
The obstacle with square cross-section brings about singularities in the pressure as well as in the derivatives of the velocity. This leads to a reduction of the order of convergence to approximately $O(h)$ with respect to uniform refinement of the spatial discretization, see, for instance, Braack and Richter [21]. Thus, it becomes impossible to obtain reliable values for the mean drag-coefficient by extrapolation of values computed on a sequence of uniformly refined discretizations. In what follows, we use an estimated value of $J_{\mathrm{drag}}(\boldsymbol{u}) \approx 2.3182$ as reference value. This value was deduced by comparing the results of all discretizations with adaptive as well as uniform refinement.

Due to the singularities mentioned above, the spatial discretization error is very dominant in the three-dimensional test case while the temporal discretization error becomes very small compared to the two-dimensional case. Therefore, we start the computations with a much coarser time step size which leads to more or less balanced temporal and spatial discretization errors. The development of the discretization errors in the mean drag-coefficient during an adaptive run with equilibration is shown in Tables 6.14 and 6.15 for the cG(1)cG(1) and cG(1)dG(1) discretization, respectively.

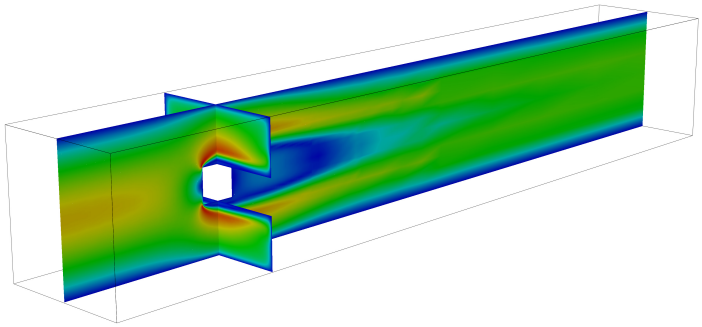
We can see that the overall discretization error is dominated by the spatial discretization error. Also note that the temporal discretization is almost uniformly refined, however, the adaptive algorithm keeps the temporal discretization fixed sometimes. This can be explained by the already mentioned singularities occurring in the solution at the corners and edges of the obstacle. On the other hand, the temporal variation of the flow is rather small and consequently the almost uniform refinement of the temporal discretization makes sense because the quantity of interest is acting over the whole time interval. Furthermore, we observe that the quantitative assessment of the discretization error by the a posteriori



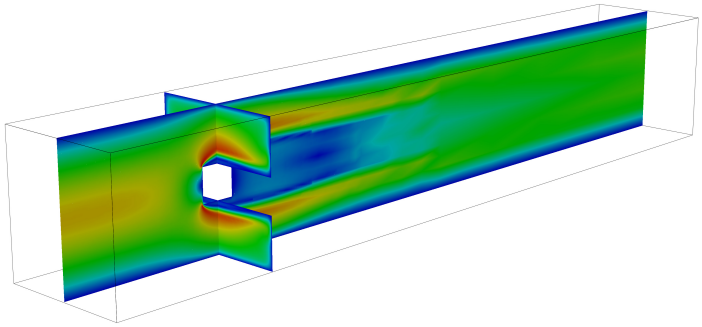
(a) $t = 1\,\mathrm{s}$



(b) $t = 2\,\mathrm{s}$

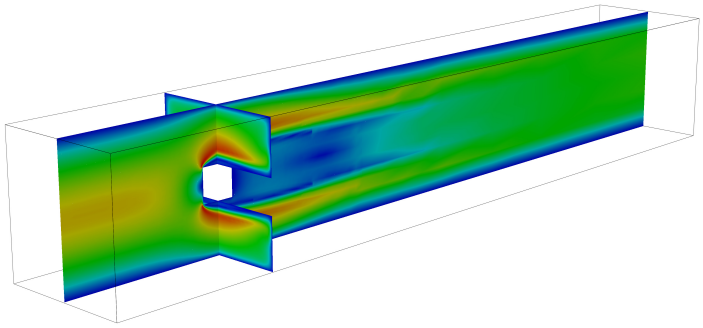


(c) $t = 3\,\mathrm{s}$

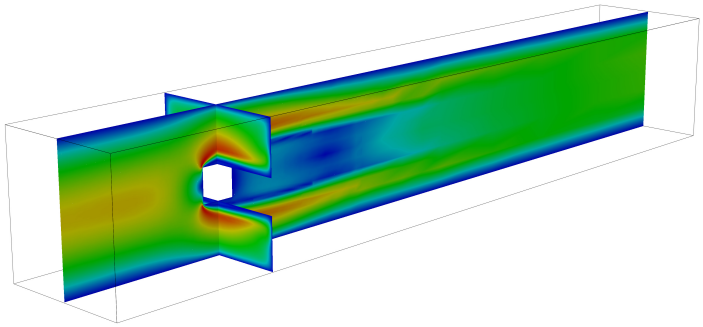


(d) $t = 4\,\mathrm{s}$

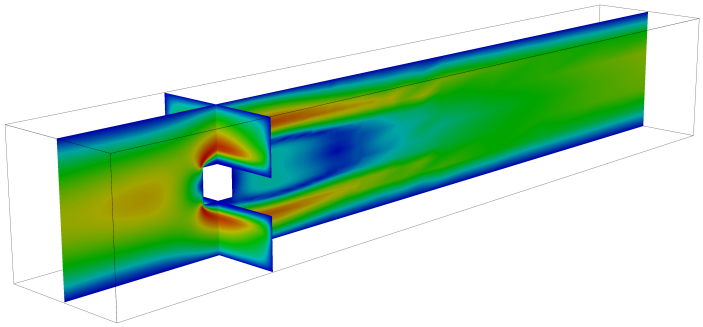
Figure 6.20. Three-dimensional test case: $|\boldsymbol{v}|$ at different times t (red: high velocity, blue: low velocity)



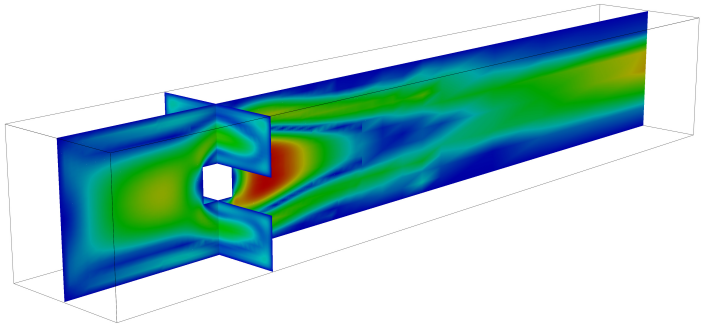
(e) $t = 5\,\mathrm{s}$



(f) $t = 6\,\mathrm{s}$



(g) $t = 7\,\mathrm{s}$



(h) $t = 8\,\mathrm{s}$

Figure 6.20. Three-dimensional test case: $|\boldsymbol{v}|$ at different times t (red: high velocity, blue: low velocity) (cont.)

Table 6.14. Mean drag-coefficient for the three-dimensional test case: Adaptive refinement with equilibration for the cG(1)cG(1) discretization

N	M	η_h	η_k	$J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})$	I_{eff}
3696	8	$-3.55 \cdot 10^{+00}$	$-1.67 \cdot 10^{-01}$	$-4.35 \cdot 10^{+00}$	1.17
17040	8	$-6.92 \cdot 10^{-01}$	$8.24 \cdot 10^{-02}$	$-5.04 \cdot 10^{-01}$	0.83
71080	8	$-1.95 \cdot 10^{-01}$	$4.09 \cdot 10^{-02}$	$-3.39 \cdot 10^{-01}$	2.19
194472	8	$-8.35 \cdot 10^{-02}$	$3.22 \cdot 10^{-02}$	$-1.63 \cdot 10^{-01}$	3.17
590952	16	$-3.86 \cdot 10^{-02}$	$9.90 \cdot 10^{-03}$	$-4.05 \cdot 10^{-02}$	1.41
1498440	16	$4.73 \cdot 10^{-04}$	$9.73 \cdot 10^{-03}$	$9.58 \cdot 10^{-03}$	0.94
1498440	18	$5.01 \cdot 10^{-04}$	$6.58 \cdot 10^{-03}$	$1.07 \cdot 10^{-02}$	1.51
1498440	36	$5.29 \cdot 10^{-04}$	$1.60 \cdot 10^{-03}$	$7.86 \cdot 10^{-03}$	3.70
1498440	62	$6.02 \cdot 10^{-04}$	$4.41 \cdot 10^{-04}$	$6.83 \cdot 10^{-03}$	6.55

Table 6.15. Mean drag-coefficient for the three-dimensional test case: Adaptive refinement with equilibration for the cG(1)dG(1) discretization

N	M	η_h	η_k	$J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})$	I_{eff}
3696	8	$-3.35 \cdot 10^{+00}$	$1.47 \cdot 10^{-01}$	$-4.90 \cdot 10^{+00}$	1.53
17040	8	$-5.63 \cdot 10^{-01}$	$6.67 \cdot 10^{-02}$	$-7.70 \cdot 10^{-01}$	1.55
71960	8	$-1.01 \cdot 10^{-01}$	$6.55 \cdot 10^{-02}$	$-4.56 \cdot 10^{-01}$	13.07
202296	16	$-6.22 \cdot 10^{-02}$	$1.56 \cdot 10^{-02}$	$-2.23 \cdot 10^{-01}$	4.81
601728	16	$-3.63 \cdot 10^{-02}$	$1.50 \cdot 10^{-02}$	$-6.24 \cdot 10^{-02}$	2.97
1501504	32	$-8.51 \cdot 10^{-04}$	$3.64 \cdot 10^{-03}$	$3.53 \cdot 10^{-03}$	0.98
1501504	64	$-7.50 \cdot 10^{-04}$	$9.11 \cdot 10^{-04}$	$1.82 \cdot 10^{-04}$	1.12

error estimator is still quite good, however, the effectivity index I_{eff} is not always as good as in the two-dimensional test case.

Figure 6.21 shows the temporal evolution of the drag-coefficient for both discretizations under consideration. The values depicted there correspond to the finest discretizations which are described in the last lines of Tables 6.14 and 6.15. As in the two-dimensional case, there is a quite good agreement of both curves, at least until $t = 7\,\text{s}$.

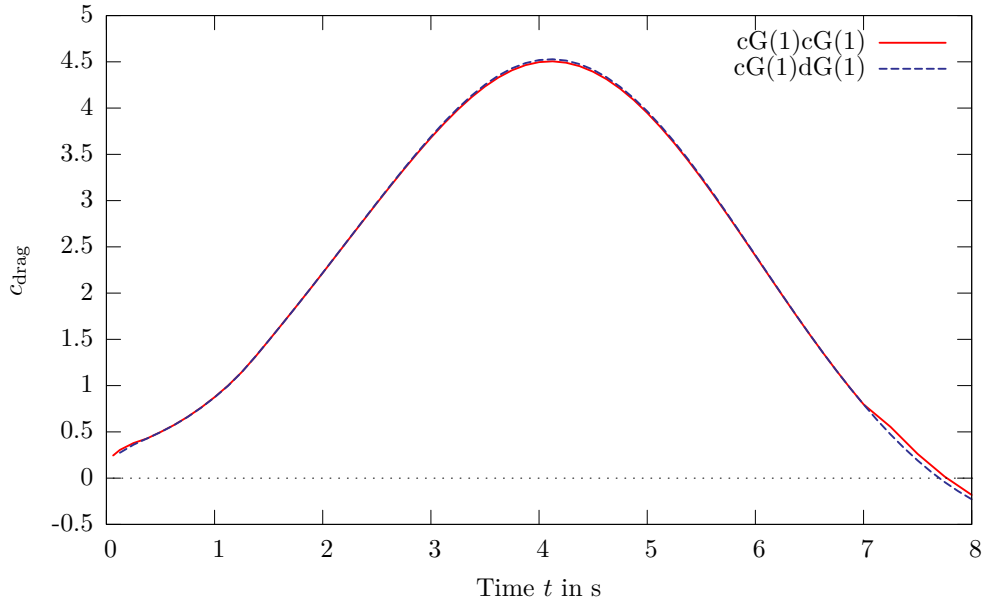
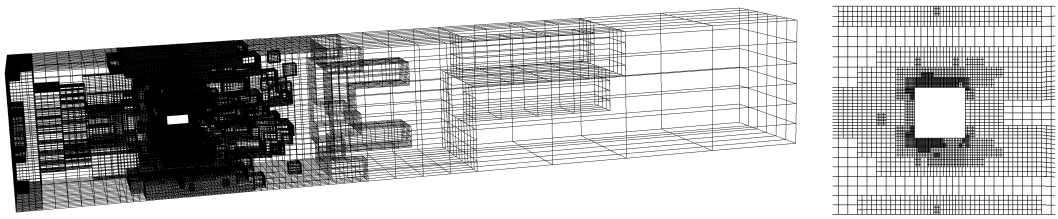


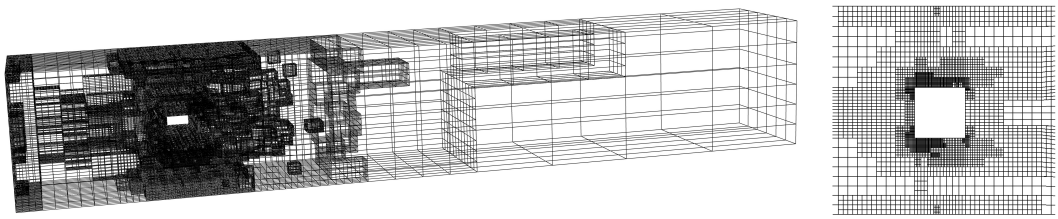
Figure 6.21. Temporal evolution of the drag-coefficient c_{drag} in the three-dimensional test case obtained by different discretizations with adaptive refinement

The adaptive spatial meshes corresponding to the finest discretizations in Tables 6.14 and 6.15 are depicted in Figure 6.22. Since it is difficult to visualize three-dimensional meshes, we also present close-ups of the area around the obstacle viewed in the plane $x_3 = 0.205\,\text{m}$. We observe that for both discretizations the meshes are strongly refined especially in the neighborhood of the corners and edges of the obstacle, while in the rest of the domain the cells are kept rather coarse. Looking at the presented meshes, we conclude that allowing dynamic spatial meshes will not provide a remarkable benefit—this is even more obvious here in the three-dimensional case than it was in the two-dimensional case.

Finally, we present in Figures 6.23 and 6.24 a comparison of different refinement strategies for the cG(1)cG(1) and the cG(1)dG(1) discretization, respectively. In these figures, the dotted line corresponds to an accuracy of 1 %. Even though the adaptive refinement of the temporal discretization leads to almost uniform refinement, we benefit from the application of the a posteriori error estimators. Firstly, the spatial discretization error is reduced much faster when using adaptive refinement. This of course is due to the corner singularities. Secondly, we also benefit from the temporal error estimator because



(a) cG(1)cG(1) discretization



(b) cG(1)dG(1) discretization

Figure 6.22. Spatial meshes for the computation of the mean drag-coefficient with different discretizations (left: whole computational domain, right: close-up of the area around the obstacle in the plane $x_3 = 0.205\,\mathrm{m}$)

without having quantitative information about the temporal discretization error, one would probably have started with a much finer time step size. However, this amounts to higher computational costs because then the temporal and spatial discretization errors would not be balanced.

Keeping this in mind, the reduction in degrees of freedom required to reach a certain accuracy would be even larger than what we obtain here: For reaching an error of 10 %, the uniformly refined discretizations need about 10 times as many degrees of freedom as the corresponding adaptive discretizations. Applying only uniform refinement, we were not even able to reach an accuracy of 1 % on a standard personal computer.

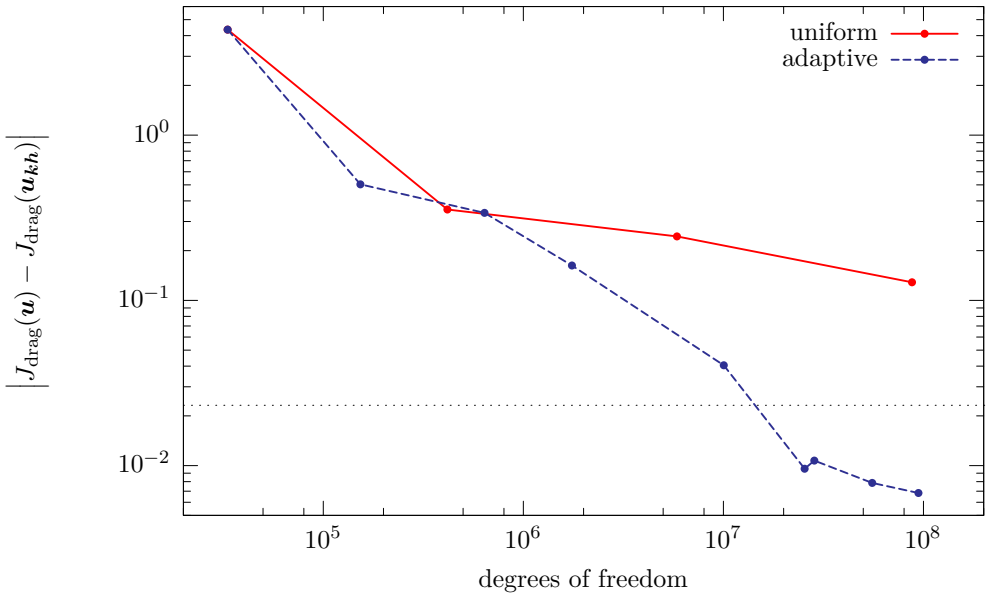


Figure 6.23. Comparison of the error $\left|J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})\right|$ in the three-dimensional test case for different refinement strategies with the cG(1)cG(1) discretization (dotted line: 1 % accuracy)

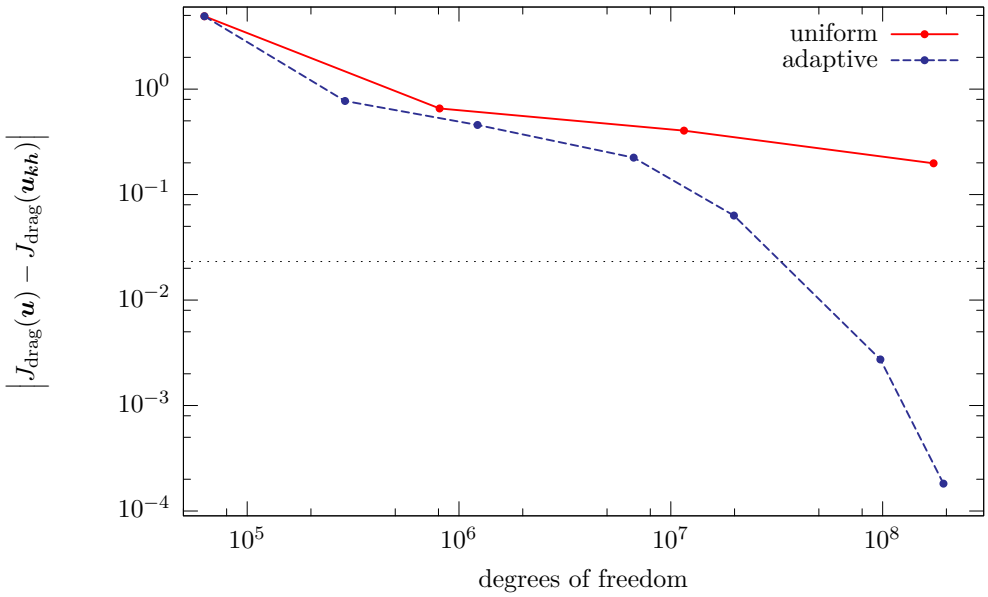


Figure 6.24. Comparison of the error $\left|J_{\text{drag}}(\boldsymbol{u}) - J_{\text{drag}}(\boldsymbol{u_{kh}})\right|$ in the three-dimensional test case for different refinement strategies with the cG(1)dG(1) discretization (dotted line: 1 % accuracy)

6.2 Filling process of a lab-on-a-chip

We consider the filling process of a lab-on-a-chip. This process is modeled as a two-phase flow problem. The first phase is the gaseous phase, for instance, air. At the beginning of the simulation, the whole chip is filled with this phase. The second phase is the liquid phase, for instance, water or blood. The liquid phase enters the chip through some inlet. During the simulation, an increasing part of the chip is occupied by the liquid phase due to capillary forces.

In this example, we have to deal with two main difficulties: First of all, the geometry of such a microdiagnosis chip is very complex as we have already shown in the introduction, see Figure 1.1 in Chapter 1. We will therefore only consider a representative cut-out of the whole geometry. Furthermore, the equations we have to deal with exhibit a highly nonlinear and indefinite structure. The next section gives more details on the implemented model for this two-phase flow problem.

We like to emphasize that the simulations presented here have to be understood as feasibility tests which show that the methods derived in this thesis are also applicable to "real-life" two-phase flow problems. The model developed in the next section incorporates microfluidic effects like surface tension and contact angles and allows for adaptive refinement based on the developed error estimator. However, especially the level set method employed for tracking the interface between the liquid and the gaseous phase is kept rather simple, see Section 6.2.3. The main focus here is to show how residual based error estimation can be performed for two-phase flow problems. In contrast to the previous examples, we do not consider simultaneous adaptation of the temporal and spatial discretization, but rather confine ourselves to adaptive refinement of the spatial discretization. It turns out that the use of the developed error estimator is advantageous compared to heuristic refinement strategies which, for example, only perform refinement of cells near the interface between both phases.

We refer to the survey article of Erickson [40] for an overview of numerical methods for simulating integrated microfluidic devices.

6.2.1 Formulation of the model

We consider each phase as an incompressible Newtonian fluid. So the motion of each phase is described by the incompressible Navier-Stokes equations. However, the domains occupied by each of the phases change in time. From the implementational point of view, another approach is therefore more desirable. We treat both phases together as one fluid, but with variable density $\rho(\boldsymbol{x})$ and viscosity $\mu(\boldsymbol{x})$. Using the precise form of the stress tensor for incompressible fluids

$$\boldsymbol{\sigma}(\boldsymbol{v}, p) = -p\mathbf{I} + \mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T),$$

the flow is governed by

$$\partial_t(\rho \boldsymbol{v}) + \rho(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + (\boldsymbol{v} \cdot \nabla\rho)\boldsymbol{v} + \nabla p - \nabla \cdot (\mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T)) = \rho \boldsymbol{f},$$
$$\nabla \cdot \boldsymbol{v} = 0.$$

This approach requires a method to identify whether a point $\boldsymbol{x} \in \Omega$ belongs to the liquid phase Ω_{liquid} or the gaseous phase Ω_{gas}. We will come back to this topic later.

Since both fluids are immiscible, there exists an interface Γ between both fluids or phases. On this internal boundary, we have to prescribe some conditions to link both phases together. For incompressible fluids which do not gain or lose mass, the normal component of the velocity on both sides of the interface must be equal:

$$[\boldsymbol{v} \cdot \boldsymbol{n}_{\boldsymbol{\Gamma}}]\Big|_{\Gamma} = 0,$$

where $[\cdot]\Big|_{\Gamma}$ denotes the jump of some quantity across the interface Γ and $\boldsymbol{n}_{\boldsymbol{\Gamma}}$ describes the unit normal vector on Γ. Furthermore, the internal forces must be balanced. If we also consider surface tension effects, this can be written as

$$\left[-p\mathbf{I} + \mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T)\right]\Big|_{\Gamma} \boldsymbol{n}_{\boldsymbol{\Gamma}} = \sigma\kappa \boldsymbol{n}_{\boldsymbol{\Gamma}}.$$

This essentially states that the difference of the pressure and the viscous forces between both phases is equal to a product of the surface tension coefficient σ and the curvature κ of the interface Γ.

Remark 6.3. Because of this interface condition, we have to work with the full symmetric tensor $\boldsymbol{\sigma}(\boldsymbol{v}, p)$. If we used only the unsymmetric tensor $-p\mathbf{I} + \mu\nabla \boldsymbol{v}$ and omitted the term $\mu\nabla \boldsymbol{v}^T$ (like it is usually done in the formulation of the Navier-Stokes equations (2.5)), the forces on the interface would not be represented correctly. However, in two-phase flow problems, these forces essentially determine the overall flow dynamics.

Following the works of Brackbill, Kothe, and Zemach [22] and Chang, Hou, Merriman, and Osher [30], the surface tension force term can be rewritten as a volume force which reads

$$\boldsymbol{f}_{\mathbf{st}}(t, \boldsymbol{x}) = \sigma\kappa(t, \boldsymbol{x})\delta_{\Gamma(t)}(\boldsymbol{x})\boldsymbol{n}_{\boldsymbol{\Gamma}}(t, \boldsymbol{x}),$$

where the Dirac delta distribution δ_Γ is used which fulfills

$$\int_{\Omega} f(\boldsymbol{x})\delta_\Gamma(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_{\Gamma} f(\boldsymbol{x}) \, \mathrm{d}o.$$

We also refer to Hysing [72] or Smolianski [97] who used this approach in the simulation of interfacial two-phase flow problems, too. This reinterpretation of surface tension forces as volume forces allows us to easily incorporate them into the variational formulation leading to just another force term on the right-hand side. However, here again, we need a way to approximate the Dirac delta function which requires locating the interface between both phases.

The boundary $\partial\Omega$ is divided into the inflow boundary Γ_{in}, the outflow boundary Γ_{out}, and the fixed boundary Γ_{fixed}. Here, we also have to prescribe boundary conditions. As a simplified model for the capillary pressure, the flow will be driven not only by surface tension effects, but also by a prescribed pressure drop P between the inflow and the outflow boundary. Hence, on Γ_{in} and Γ_{out}, we use the following boundary conditions:

$$\begin{aligned} \boldsymbol{\sigma}(\boldsymbol{v},p)\boldsymbol{n} &= -P\boldsymbol{n} && \text{on } \Gamma_{\text{in}}, \\ \boldsymbol{\sigma}(\boldsymbol{v},p)\boldsymbol{n} &= \boldsymbol{0} && \text{on } \Gamma_{\text{out}}. \end{aligned} \tag{6.4}$$

To allow so-called *wetting effects*, we use free-slip boundary conditions on Γ_{fixed} together with some condition on the tangential stresses instead of no-slip conditions:

$$\boldsymbol{v}\cdot\boldsymbol{n} = 0, \qquad \boldsymbol{t_i}\cdot\boldsymbol{\sigma}(\boldsymbol{v},p)\boldsymbol{n} = 0, \quad i = 1,\dots,d-1, \tag{6.5}$$

where $\boldsymbol{t_i}$, $i = 1,\dots,d-1$ are unit tangential vectors on Γ_{fixed}.

Remark 6.4. These boundary conditions indeed lead to the mean pressure P on Γ_{in} and mean pressure 0 on Γ_{out} if the inflow and outflow boundary are perpendicular to the fixed boundary Γ_{fixed} and the interface between both phases does not cross the inflow or outflow boundary. For simplicity, we present the arguments for the two-dimensional case as depicted in Figure 6.25. However, they are easily transfered to the three-dimensional case.

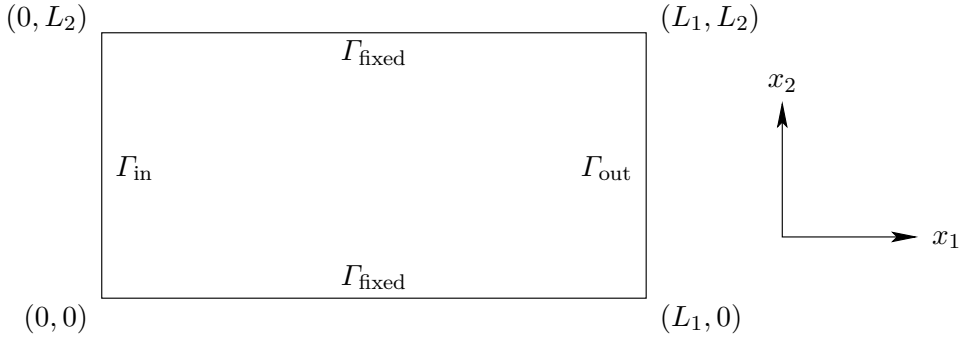


Figure 6.25. Rectangular domain with inflow boundary Γ_{in}, outflow boundary Γ_{out}, and fixed boundary Γ_{fixed}

Since $\boldsymbol{\sigma}(\boldsymbol{v},p)\boldsymbol{n} = -P\boldsymbol{n}$ on Γ_{in}, we have by employing the precise form of $\boldsymbol{\sigma}(\boldsymbol{v},p)$

$$\int_{\Gamma_{\text{in}}} p\boldsymbol{n}\ \mathrm{d}o = \int_{\Gamma_{\text{in}}} P\boldsymbol{n}\ \mathrm{d}o + \int_{\Gamma_{\text{in}}} \mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T)\boldsymbol{n}\ \mathrm{d}o.$$

Using the fact that $\boldsymbol{n} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$ on Γ_{in}, we obtain

$$\int_{\Gamma_{\text{in}}} p\ \mathrm{d}o = P\,|\Gamma_{\text{in}}| + \int_{\Gamma_{\text{in}}} 2\mu\partial_{x_1} v_1\ \mathrm{d}o.$$

Because the interface between both phases does not cross the inflow boundary, μ is constant on Γ_{in}. Employing the incompressibility condition $\nabla \cdot \boldsymbol{v} = 0$ or equivalently $\partial_{x_1} v_1 = -\partial_{x_2} v_2$ yields

$$\begin{aligned}\int_{\Gamma_{\text{in}}} p \, \mathrm{d}o &= P\,|\Gamma_{\text{in}}| - 2\mu \int_0^{L_2} \partial_{x_2} v_2(0, x_2) \, \mathrm{d}x_2 \\ &= P\,|\Gamma_{\text{in}}| - 2\mu v_2(0, L_2) + 2\mu v_2(0, 0) \\ &= P\,|\Gamma_{\text{in}}|\,.\end{aligned}$$

Here, the boundary condition $\boldsymbol{v} \cdot \boldsymbol{n} = 0$ on Γ_{fixed} has been used to conclude $v_2(x_1, 0) = v_2(x_1, L_2) = 0$. Thus, we have

$$\frac{1}{|\Gamma_{\text{in}}|} \int_{\Gamma_{\text{in}}} p \, \mathrm{d}o = P.$$

In a similar way, the mean pressure condition on Γ_{out}

$$\frac{1}{|\Gamma_{\text{out}}|} \int_{\Gamma_{\text{out}}} p \, \mathrm{d}o = 0$$

can be derived.

6.2.2 Nondimensionalization for two-phase flow problems

For the numerical simulation, it is necessary to introduce a nondimensional form of the problem under consideration. Let us recall the equations describing the two-phase flow problem including gravitational and surface tension forces:

$$\begin{aligned}\partial_t(\rho \boldsymbol{v}) + \rho(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + (\boldsymbol{v} \cdot \nabla\rho)\boldsymbol{v} + \nabla p - \nabla \cdot (\mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T)) &= \rho \boldsymbol{f} + \sigma\kappa\delta_\Gamma \boldsymbol{n}_{\boldsymbol{\Gamma}}, \\ \nabla \cdot \boldsymbol{v} &= 0.\end{aligned} \tag{6.6}$$

We choose a characteristic length L, a characteristic velocity U, a characteristic density ρ_{ref}, and a characteristic viscosity μ_{ref}. The scales automatically also define a characteristic time scale as LU^{-1}. With the help of these scales, we define the following nondimensional variables:

$$\hat{\boldsymbol{x}} \coloneqq L^{-1}\boldsymbol{x}, \quad \hat{\boldsymbol{v}} \coloneqq U^{-1}\boldsymbol{v}, \quad \hat{t} \coloneqq L^{-1}Ut, \quad \hat{\rho} \coloneqq \rho_{\text{ref}}^{-1}\rho, \quad \hat{\mu} \coloneqq \mu_{\text{ref}}^{-1}\mu, \quad \hat{p} \coloneqq \rho_{\text{ref}}^{-1}U^{-2}p.$$

Simple calculations show that (6.6) is equivalent to

$$\begin{aligned}\partial_{\hat{t}}(\hat{\rho}\hat{\boldsymbol{v}}) + \hat{\rho}(\hat{\boldsymbol{v}} \cdot \hat{\nabla})\hat{\boldsymbol{v}} + (\hat{\boldsymbol{v}} \cdot \hat{\nabla}\hat{\rho})\hat{\boldsymbol{v}} + \hat{\nabla}\hat{p} - \frac{\mu_{\text{ref}}}{\rho_{\text{ref}} LU}\hat{\nabla} \cdot (\hat{\mu}(\hat{\nabla}\hat{\boldsymbol{v}} + \hat{\nabla}\hat{\boldsymbol{v}}^T)) &= \hat{\rho}\hat{\boldsymbol{f}} + \frac{\sigma}{\rho_{\text{ref}} LU^2}\hat{\kappa}\delta_\Gamma \boldsymbol{n}_{\boldsymbol{\Gamma}}, \\ \hat{\nabla} \cdot \hat{\boldsymbol{v}} &= 0.\end{aligned}$$

Here, we have used the notations

$$\hat{\boldsymbol{f}} \coloneqq \frac{L}{U^2}\boldsymbol{f} \quad \text{and} \quad \hat{\kappa} \coloneqq L^2\kappa.$$

Since $\boldsymbol{f}$ represents gravitational forces, we may write

$$\boldsymbol{f} = g\boldsymbol{e_g}$$

with the gravitational constant g and a unit vector $\boldsymbol{e_g}$ pointing in the direction in which the gravitational forces act. Hence, we have

$$\hat{\boldsymbol{f}} = \frac{gL}{U^2}\boldsymbol{e_g}.$$

If we relabel all variables and again write $\boldsymbol{v}$ instead of $\hat{\boldsymbol{v}}$, for example, the nondimensional form of the two-phase flow problem (6.6) is given by

$$\begin{aligned}\partial_t(\rho\boldsymbol{v}) + \rho(\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla\rho)\boldsymbol{v} + \nabla p - \frac{1}{\mathrm{Re}}\nabla\cdot(\mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T)) &= \frac{\rho}{\mathrm{Fr}}\boldsymbol{e_g} + \frac{1}{\mathrm{We}}\kappa\delta_\Gamma\boldsymbol{n_\Gamma},\\ \nabla\cdot\boldsymbol{v} &= 0.\end{aligned} \tag{6.7}$$

In this form, the Reynolds number Re, the Froude number Fr, and the Weber number We, given by

$$\mathrm{Re} \coloneqq \frac{\rho_{\mathrm{ref}}LU}{\mu_{\mathrm{ref}}}, \quad \mathrm{Fr} \coloneqq \frac{U^2}{gL}, \quad \mathrm{We} \coloneqq \frac{\rho_{\mathrm{ref}}LU^2}{\sigma},$$

respectively, have been introduced. These nondimensional numbers are used to characterize the flow. The Reynolds number Re relates inertial to viscous forces while the Froude number Fr and the Weber number We relate inertial to gravitational effects and inertial to surface tension effects, respectively.

In our case, we choose the characteristic scales as

$$L \coloneqq 10^{-3}\,\mathrm{m}, \quad U \coloneqq 10^{-3}\,\mathrm{m\,s^{-1}}, \quad \rho_{\mathrm{ref}} \coloneqq 10^{3}\,\mathrm{kg\,m^{-3}}, \quad \mu_{\mathrm{ref}} \coloneqq 10^{-3}\,\mathrm{kg\,m^{-1}\,s^{-1}}$$

which leads to a Reynolds number of $\mathrm{Re} = 1$.

6.2.3 Discretization of the model

In this section, we give details on several aspects of the discretization of the two-phase flow model.

Starting point again is a variational formulation of the governing equations. For its derivation, we assume we have a smooth solution and multiply equations (6.7) with appropriate test functions $\boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi)^T$ satisfying

$$\boldsymbol{\psi} \in H_{\mathrm{slip}} \coloneqq \left\{ \boldsymbol{v} \in H^1(\Omega)^d \;\middle|\; \boldsymbol{v}\cdot\boldsymbol{n} = 0 \text{ on } \Gamma_{\mathrm{fixed}} \right\} \quad \text{and} \quad \chi \in L_{\mathrm{slip}} \coloneqq L^2(\Omega).$$

Using integration by parts, we obtain

$$\begin{aligned}(\partial_t(\rho\boldsymbol{v}),\boldsymbol{\psi}) + (\rho(\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla\rho)\boldsymbol{v},\boldsymbol{\psi}) + (\mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T),\nabla\boldsymbol{\psi}) - (p,\nabla\cdot\boldsymbol{\psi})\\ - (\mu(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T)\boldsymbol{n},\boldsymbol{\psi})_{\partial\Omega} + (p\boldsymbol{n},\boldsymbol{\psi})_{\partial\Omega} = \frac{1}{\mathrm{Fr}}(\rho\boldsymbol{e_g},\boldsymbol{\psi}) + \frac{1}{\mathrm{We}}(\kappa\delta_\Gamma\boldsymbol{n_\Gamma},\boldsymbol{\psi})\end{aligned} \tag{6.8}$$

and

$$(\nabla \cdot \boldsymbol{v}, \chi) = 0. \tag{6.9}$$

Recalling the boundary conditions (6.4) on Γ_{in} and Γ_{out} and the definition of the stress tensor $\boldsymbol{\sigma}(\boldsymbol{v}, p) = -p\mathbf{I} + \mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T)$, (6.8) can be modified to

$$\begin{aligned}(\partial_t(\rho \boldsymbol{v}), \boldsymbol{\psi}) + (\rho(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + (\boldsymbol{v} \cdot \nabla\rho)\boldsymbol{v}, \boldsymbol{\psi}) + (\mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T), \nabla \boldsymbol{\psi}) - (p, \nabla \cdot \boldsymbol{\psi}) \\ - (\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{fixed}}} = -(P\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{in}}} + \frac{1}{\text{Fr}}(\rho \boldsymbol{e_g}, \boldsymbol{\psi}) + \frac{1}{\text{We}}(\kappa \delta_\Gamma \boldsymbol{n_\Gamma}, \boldsymbol{\psi})\end{aligned} \tag{6.10}$$

On Γ_{fixed}, we decompose $\boldsymbol{\psi} \in H_{\text{slip}}$ as

$$\boldsymbol{\psi} = (\boldsymbol{\psi} \cdot \boldsymbol{n})\boldsymbol{n} + \sum_{i=1}^{d-1} (\boldsymbol{\psi} \cdot \boldsymbol{t_i})\boldsymbol{t_i} = \sum_{i=1}^{d-1} (\boldsymbol{\psi} \cdot \boldsymbol{t_i})\boldsymbol{t_i}.$$

Inserting this into $(\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{fixed}}}$ and recalling the free-slip boundary condition (6.5) yields

$$\begin{aligned}(\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{fixed}}} &= \sum_{i=1}^{d-1} (\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, (\boldsymbol{\psi} \cdot \boldsymbol{t_i})\boldsymbol{t_i})_{\Gamma_{\text{fixed}}} \\ &= \sum_{i=1}^{d-1} (\boldsymbol{t_i} \cdot \boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi} \cdot \boldsymbol{t_i})_{\Gamma_{\text{fixed}}} \\ &= 0.\end{aligned} \tag{6.11}$$

Hence, the variational formulation (6.10) reduces to

$$\begin{aligned}(\partial_t(\rho \boldsymbol{v}), \boldsymbol{\psi}) + (\rho(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + (\boldsymbol{v} \cdot \nabla\rho)\boldsymbol{v}, \boldsymbol{\psi}) + (\mu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T), \nabla \boldsymbol{\psi}) - (p, \nabla \cdot \boldsymbol{\psi}) \\ = -(P\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{in}}} + \frac{1}{\text{Fr}}(\rho \boldsymbol{e_g}, \boldsymbol{\psi}) + \frac{1}{\text{We}}(\kappa \delta_\Gamma \boldsymbol{n_\Gamma}, \boldsymbol{\psi})\end{aligned} \tag{6.12}$$

Let us now come back to the problem of identifying which phase a point $\boldsymbol{x} \in \Omega$ belongs to. This is done by a level set approach. Level set methods were introduced by Osher and Sethian [90]. There is a wide variety of articles and books coping with this topic, for example, Chang, Hou, Merriman, and Osher [30], Evans and Spruck [50], Groß, Reichelt, and Reusken [62], Hysing [71, 72], Nagrath, Jansen, and Lahey Jr. [83], Olsson and Kreiss [87], Olsson, Kreiss, and Zahedi [88], Smolianski [97], Sussman, Smereka, and Osher [98], Zhao, Chan, Merriman, and Osher [109], as well as the textbook of Osher and Fedkiw [89]. The main idea of the level set approach is to represent the interface $\Gamma(t)$ between both phases by the zero level set of a function $\phi\colon I \times \Omega \to \mathbb{R}$:

$$\Gamma(t) \coloneqq \{\, \boldsymbol{x} \in \Omega \mid \phi(t, \boldsymbol{x}) = 0 \,\}.$$

The level set function is usually initialized as a signed distance function:

$$\phi(0, \boldsymbol{x}) \coloneqq \begin{cases} \operatorname{dist}(\Gamma, \boldsymbol{x}) & \boldsymbol{x} \in \Omega_{\text{liquid}}, \\ 0 & \boldsymbol{x} \in \Gamma, \\ -\operatorname{dist}(\Gamma, \boldsymbol{x}) & \boldsymbol{x} \in \Omega_{\text{gas}}. \end{cases}$$

However, it is sufficient for the level set function to be close to a signed distance function near Γ. In Figure 6.26, a signed distance function corresponding to a circular interface Γ is shown.

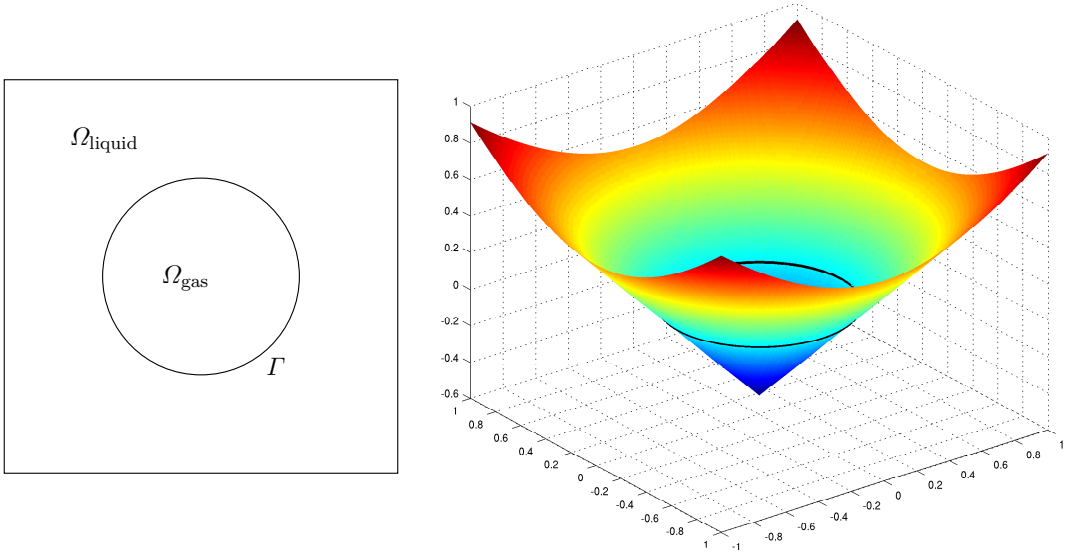


Figure 6.26. Circular interface Γ and corresponding signed distance function

To derive an equation for the level set function ϕ, we first observe that the following equation holds for points $\boldsymbol{x}(t)$ on the moving interface $\Gamma(t)$:

$$\phi(t, \boldsymbol{x}(t)) = 0.$$

Differentiation with respect to t then yields by applying the chain rule:

$$\partial_t \phi + \partial_t \boldsymbol{x} \cdot \nabla \phi = 0.$$

Since the points $\boldsymbol{x} \in \Omega$ are moving with speed $\boldsymbol{v}(\boldsymbol{x})$, we obtain

$$\partial_t \phi + \boldsymbol{v} \cdot \nabla \phi = 0.$$

Using the level set function ϕ, we are able to express the density and viscosity by

$$\begin{aligned} \rho(\boldsymbol{x}) &= \rho_{\text{gas}} + (\rho_{\text{liquid}} - \rho_{\text{gas}}) H(\phi(\boldsymbol{x})), \\ \mu(\boldsymbol{x}) &= \mu_{\text{gas}} + (\mu_{\text{liquid}} - \mu_{\text{gas}}) H(\phi(\boldsymbol{x})), \end{aligned}$$

where the Heaviside function $H : \mathbb{R} \to \mathbb{R}$ with

$$H(s) = \begin{cases} 0 & s < 0, \\ 1 & s > 0 \end{cases}$$

is used and ρ_{liquid}, μ_{liquid} and ρ_{gas}, μ_{gas} denote the constant density and viscosity of the liquid and the gaseous phase, respectively. Additionally, the level set approach allows us to globally define the normal vectors and curvature by

$$\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|} \quad \text{and} \quad \kappa = -\nabla\cdot\boldsymbol{n} = -\nabla\cdot\left(\frac{\nabla\phi}{|\nabla\phi|}\right). \tag{6.13}$$

Remark 6.5. As already pointed out above, the level set approach employed here is rather simple. Usually one has to reinitialize the level set function after some time steps to keep it close to a signed distance function. This is necessary in order to obtain good approximations of the normal vectors and to keep the width of the interface fixed, see below for further details. Fortunately, due to the specific velocity field which develops in this example, we can omit this step, because the level set function does not deviate too much from a signed distance function, at least in the neighborhood of the interface. If we had to apply reinitialization procedures, we would not be able to use the developed a posteriori error estimators because they rely on a coupled variational formulation. However, the established reinitialization methods do not fit into this framework, see, for example, Hysing [72] for an overview of reinitialization schemes.

Furthermore, we do not consider the aspect of mass conservation. We refer to Olsson and Kreiss [87] and Olsson, Kreiss, and Zahedi [88] for a conservative level set method. This method also relies on some sort of reinitialization step. However, this intermediate step allows for a better conservation of mass.

We will now concentrate on the surface tension term and present a method which avoids explicitly evaluating the curvature κ of the interface Γ. This is especially important in the spatially discretized problem because the evaluation of the curvature via (6.13) obviously requires the evaluation of second derivatives which in general will not provide accurate approximations of the continuous curvature. The modification we present here is due to Dziuk [39] and has also been used by many others, for example, Bänsch [7], Ganesan and Tobiska [55], Groß [61], and Hysing [71]. See also the literature cited therein.

Before introducing this approach, we give some definitions from differential geometry.

Definition 6.1 (Tangential Gradient). *Let f be a function which is differentiable in an open neighborhood of Γ. Then the* tangential gradient *of f is defined as*

$$\underline{\nabla} f(\boldsymbol{x}) \coloneqq \nabla f(\boldsymbol{x}) - (\boldsymbol{n}_{\boldsymbol{\Gamma}}(\boldsymbol{x})\cdot\nabla f(\boldsymbol{x}))\boldsymbol{n}_{\boldsymbol{\Gamma}}(\boldsymbol{x}), \quad \boldsymbol{x}\in\Gamma.$$

Definition 6.2 (Laplace-Beltrami Operator). *Let f be a function which is two times differentiable in an open neighborhood of Γ. Then the* Laplace-Beltrami operator *of f is defined as*

$$\underline{\Delta} f(\boldsymbol{x}) \coloneqq \underline{\nabla}\cdot(\underline{\nabla} f(\boldsymbol{x})), \quad \boldsymbol{x}\in\Gamma.$$

Using these definitions, we can state a well-known result from differential geometry:

Lemma 6.1. *Let* $\mathbf{id}_\Gamma : \Gamma \to \Gamma$ *be the identity mapping on* Γ*. There holds*

$$\underline{\Delta}\mathbf{id}_\Gamma = \kappa \boldsymbol{n}_\Gamma.$$

Proof. A proof can be found, for instance, in Gallot, Hulin, and Lafontaine [54]. □

Inserting this relation into the surface tension term above yields

$$(\kappa\delta_\Gamma \boldsymbol{n}_\Gamma, \boldsymbol{\psi}) = \int_\Gamma \kappa \boldsymbol{n}_\Gamma \cdot \boldsymbol{\psi} \, \mathrm{d}o = \int_\Gamma \underline{\Delta}\mathbf{id}_\Gamma \cdot \boldsymbol{\psi} \, \mathrm{d}o.$$

Using integration by parts on Γ (see Gallot, Hulin, and Lafontaine [54]) supplies

$$\begin{aligned} \int_\Gamma \underline{\Delta}\mathbf{id}_\Gamma \cdot \boldsymbol{\psi} \, \mathrm{d}o &= -\int_\Gamma \underline{\nabla}\mathbf{id}_\Gamma \cdot \underline{\nabla}\boldsymbol{\psi} \, \mathrm{d}o + \int_\gamma (\boldsymbol{n}_\gamma \cdot \underline{\nabla})\mathbf{id}_\Gamma \cdot \boldsymbol{\psi} \, \mathrm{d}s \\ &= -\int_\Gamma \underline{\nabla}\mathbf{id}_\Gamma \cdot \underline{\nabla}\boldsymbol{\psi} \, \mathrm{d}o + \int_\gamma \boldsymbol{n}_\gamma \cdot \boldsymbol{\psi} \, \mathrm{d}s, \end{aligned} \tag{6.14}$$

because there holds $(\boldsymbol{n}_\gamma \cdot \underline{\nabla})\mathbf{id}_\Gamma = \boldsymbol{n}_\gamma$ for the intrinsic outer unit normal vector $\boldsymbol{n}_\gamma$ of Γ at $\gamma = \partial\Gamma$, see Figure 6.27.

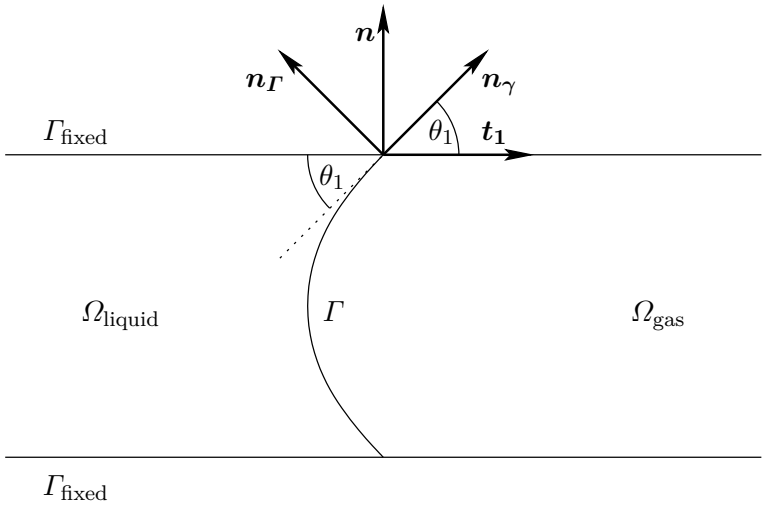


Figure 6.27. Normal and tangential vectors at the interface Γ in two spatial dimensions

Remark 6.6. If Γ possesses a closed shape or Dirichlet boundary conditions are given on Γ_{fixed} and hence $\boldsymbol{\psi} = \mathbf{0}$ on Γ_{fixed}, the last term in (6.14) vanishes.

We recall the decomposition

$$\boldsymbol{\psi} = (\boldsymbol{\psi} \cdot \boldsymbol{n})\boldsymbol{n} + \sum_{i=1}^{d-1} (\boldsymbol{\psi} \cdot \boldsymbol{t_i})\boldsymbol{t_i} \quad \text{on } \Gamma_{\text{fixed}}$$

with the outer unit normal vector $\boldsymbol{n}$ of Γ_{fixed} and unit tangential vectors $\boldsymbol{t_1}, \dots, \boldsymbol{t_{d-1}}$. Testing (6.14) with test functions $\boldsymbol{\psi} \in H_{\text{slip}}$ fulfilling $\boldsymbol{\psi} \cdot \boldsymbol{n} = 0$ on Γ_{fixed} leads to

$$\begin{aligned} \int_\Gamma \underline{\Delta}\mathbf{id}_\Gamma \cdot \boldsymbol{\psi} \, \mathrm{d}o &= -\int_\Gamma \underline{\nabla}\mathbf{id}_\Gamma \cdot \underline{\nabla}\boldsymbol{\psi} \, \mathrm{d}o + \sum_{i=1}^{d-1} \int_\gamma (\boldsymbol{n_\gamma} \cdot \boldsymbol{t_i})(\boldsymbol{\psi} \cdot \boldsymbol{t_i}) \, \mathrm{d}s \\ &= -\int_\Gamma \underline{\nabla}\mathbf{id}_\Gamma \cdot \underline{\nabla}\boldsymbol{\psi} \, \mathrm{d}o + \sum_{i=1}^{d-1} \int_\gamma \cos(\theta_i)(\boldsymbol{\psi} \cdot \boldsymbol{t_i}) \, \mathrm{d}s \end{aligned} \tag{6.15}$$

with the contact angles θ_i, see also Figure 6.27.

The complete variational formulation of the two-phase flow problem hence reads: Find $\boldsymbol{u} = (\boldsymbol{v}, p, \phi)^T \in H_{\text{slip}} \times L_{\text{slip}} \times H^1(\Omega)$ such that $\boldsymbol{v}(0, \boldsymbol{x}) = \boldsymbol{v^0}(\boldsymbol{x})$, $\phi(0, \boldsymbol{x}) = \phi^0(\boldsymbol{x})$, and

$$a(\boldsymbol{u})(\boldsymbol{\varphi}) = -(P\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{in}}} \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi, \xi)^T \in H_{\text{slip}} \times L_{\text{slip}} \times H^1(\Omega), \tag{6.16}$$

where

$$\begin{aligned} a(\boldsymbol{u})(\boldsymbol{\varphi}) \coloneqq\ & (\partial_t(\rho(\phi)\boldsymbol{v}), \boldsymbol{\psi}) + (\rho(\phi)(\boldsymbol{v} \cdot \nabla)\boldsymbol{v} + (\boldsymbol{v} \cdot \nabla\rho(\phi))\boldsymbol{v}, \boldsymbol{\psi}) \\ & + (\mu(\phi)(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T), \nabla\boldsymbol{\psi}) - (p, \nabla \cdot \boldsymbol{\psi}) \\ & - \frac{1}{\text{Fr}}(\rho(\phi)\boldsymbol{e_g}, \boldsymbol{\psi}) + \frac{1}{\text{We}} \int_\Gamma \underline{\nabla}\mathbf{id}_\Gamma \cdot \underline{\nabla}\boldsymbol{\psi} \, \mathrm{d}o - \frac{1}{\text{We}} \sum_{i=1}^{d-1} \cos(\theta_i) \int_\gamma \boldsymbol{\psi} \cdot \boldsymbol{t_i} \, \mathrm{d}s \\ & + (\nabla \cdot \boldsymbol{v}, \chi) + (\partial_t\phi, \xi) + (\boldsymbol{v} \cdot \nabla\phi, \xi). \end{aligned}$$

For the temporal discretization, we choose the fractional-step-θ scheme while the spatial discretization is done using a cG(1) method for all unknowns on dynamic meshes. This selection is reasonable because on the one hand, we want to accurately resolve the interface between the liquid and the gaseous phase over the whole time interval while at the same time keep the number of degrees of freedom as small as possible. On the other hand, the dynamics of the flow do not change much in time and hence an equidistant temporal discretization is sufficient, see Section 6.2.4 for results. Therefore, we do not need a posteriori error estimates for adaptive refinement of the temporal discretization. Furthermore, we do not have to worry about bad approximations of the pressure on dynamic meshes because of the fractional-step-θ scheme used for the temporal discretization, see Chapter 5.

Before introducing the spatial discretization, we note that the semi-linear form $a(\cdot)(\cdot)$ involves integration of discontinuous functions due to the discontinuous density and viscosity. In practice, integration is done by cell-wise applying quadrature rules. However, those quadrature rules rely on the boundedness of higher order derivatives of the integrand. A usual way to overcome this difficulty is to smoothen the Heaviside function H and to replace the discontinuous density and viscosity by their smoothed counterparts. We use the following regularized Heaviside function

$$H_\varepsilon(s) \coloneqq \begin{cases} 0 & s < -\varepsilon, \\ \frac{1}{2}\left(1 + \frac{x}{\varepsilon} + \frac{\sin(\frac{\pi s}{\varepsilon})}{\pi}\right) & |s| \le \varepsilon, \\ 1 & s > \varepsilon \end{cases}$$

and set

$$\rho_\varepsilon(\phi) \coloneqq \rho_{\text{gas}} + (\rho_{\text{liquid}} - \rho_{\text{gas}})H_\varepsilon(\phi) \quad \text{and} \quad \mu_\varepsilon(\phi) \coloneqq \mu_{\text{gas}} + (\mu_{\text{liquid}} - \mu_{\text{gas}})H_\varepsilon(\phi).$$

The regularization parameter should be chosen according to the cell size h. By differentiation of the regularized Heaviside function, we obtain a regularized approximation of the Dirac delta function:

$$\delta_\varepsilon(s) \coloneqq \begin{cases} 0 & |s| > \varepsilon, \\ \frac{1}{2\varepsilon}(1 + \cos(\frac{\pi s}{\varepsilon})) & |s| \leq \varepsilon. \end{cases}$$

Employing the regularized delta function, we are able to replace the integrals over Γ by integrals over Ω via

$$\int_\Gamma f(\boldsymbol{x}) \, \mathrm{d}o \approx \int_\Omega \tilde{f}(\boldsymbol{x})\delta_\varepsilon(\phi(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{x}$$

where $\tilde{f}$ is an extension of f from Γ onto Ω. Hence, the surface tension terms can be approximated by

$$\int_\Gamma \underline{\nabla}\mathbf{id}_{\boldsymbol{\Gamma}} \cdot \underline{\nabla}\boldsymbol{\psi} \, \mathrm{d}o \approx \int_\Omega (\underline{\nabla}\widetilde{\mathbf{id}}_{\boldsymbol{\Gamma}} \cdot \underline{\nabla}\boldsymbol{\psi})\delta_\varepsilon(\phi) \, \mathrm{d}\boldsymbol{x} \quad \text{and}$$
$$\sum_{i=1}^{d-1} \cos(\theta_i) \int_\gamma \boldsymbol{\psi} \cdot \boldsymbol{t_i} \, \mathrm{d}s \approx \sum_{i=1}^{d-1} \cos(\theta_i) \int_{\partial\Omega} (\boldsymbol{\psi} \cdot \boldsymbol{t_i})\delta_\varepsilon(\phi) \, \mathrm{d}o,$$

where $\widetilde{\mathbf{id}}_{\boldsymbol{\Gamma}}$ is the extension of $\mathbf{id}_{\boldsymbol{\Gamma}}$ onto Ω simply given by $\mathbf{id}_{\boldsymbol{\Omega}}$.

This spatial discretization again needs stabilization due to the inf-sup condition which is not fulfilled as well as stabilization of the convective terms. This is again done by the local projection stabilization method. The free-slip boundary conditions on Γ_{fixed} are imposed weakly following the approach presented in Freund and Stenberg [53]. This method is based on a technique introduced by Nitsche [86]. The main idea is to consistently add symmetric terms which vanish on the continuous solution. Recalling (6.5) and the decomposition

$$\boldsymbol{\psi} = (\boldsymbol{\psi} \cdot \boldsymbol{n})\boldsymbol{n} + \sum_{i=1}^{d-1} (\boldsymbol{\psi} \cdot \boldsymbol{t_i})\boldsymbol{t_i},$$

we have

$$\begin{aligned}
(\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{fixed}}} &= (\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, (\boldsymbol{\psi} \cdot \boldsymbol{n})\boldsymbol{n})_{\Gamma_{\text{fixed}}} + \sum_{i=1}^{d-1} (\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, (\boldsymbol{\psi} \cdot \boldsymbol{t_i})\boldsymbol{t_i})_{\Gamma_{\text{fixed}}} \\
&= (\boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi} \cdot \boldsymbol{n})_{\Gamma_{\text{fixed}}} + \sum_{i=1}^{d-1} (\boldsymbol{t_i} \cdot \boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi} \cdot \boldsymbol{t_i})_{\Gamma_{\text{fixed}}} \\
&= (\boldsymbol{n} \cdot \boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{\psi} \cdot \boldsymbol{n})_{\Gamma_{\text{fixed}}}.
\end{aligned}$$

Hence, we conclude that the discrete version of (6.16) may be formulated as

$$a_h(\boldsymbol{u_h})(\boldsymbol{\varphi}) = -(P\boldsymbol{n}, \boldsymbol{\psi})_{\Gamma_{\text{in}}} \quad \forall \boldsymbol{\varphi} = (\boldsymbol{\psi}, \chi, \xi)^T \in V_h^d \times V_h \times V_h, \tag{6.17}$$

where $V_h = \left\{ v_h \in C(\overline{\Omega}) \;\middle|\; v_h\big|_K \in \mathcal{Q}_1(K)\ \forall K \in \mathcal{T}_h \right\}$ and

$$\begin{aligned}
a_h(\boldsymbol{u})(\boldsymbol{\varphi}) \coloneqq\ & (\partial_t(\rho_\varepsilon(\phi)\boldsymbol{v}), \boldsymbol{\psi}) + (\rho_\varepsilon(\phi)(\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla\rho_\varepsilon(\phi))\boldsymbol{v}, \boldsymbol{\psi}) \\
& + (\mu_\varepsilon(\phi)(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T), \nabla\boldsymbol{\psi}) - (p, \nabla\cdot\boldsymbol{\psi}) \\
& - \frac{1}{\mathrm{Fr}}(\rho_\varepsilon(\phi)\boldsymbol{e_g}, \boldsymbol{\psi}) + \frac{1}{\mathrm{We}}(\underline{\nabla}\mathbf{id}_{\boldsymbol{\Omega}}, \delta_\varepsilon(\phi)\underline{\nabla}\boldsymbol{\psi}) - \frac{1}{\mathrm{We}}\sum_{i=1}^{d-1}\cos(\theta_i)(\boldsymbol{\psi}, \delta_\varepsilon(\phi)\boldsymbol{t_i})_{\partial\Omega} \\
& + (\nabla\cdot\boldsymbol{v}, \chi) + (\partial_t\phi, \xi) + (\boldsymbol{v}\cdot\nabla\phi, \xi) \\
& + \sum_{K\in\mathcal{T}_h}\Big\{(\nabla\pi p, \alpha_K\nabla\pi\chi)_K + (\rho_\varepsilon(\phi)(\boldsymbol{v}\cdot\nabla)\boldsymbol{\pi v}, \delta_K^{(\boldsymbol{v})}\rho_\varepsilon(\phi)(\boldsymbol{v}\cdot\nabla)\boldsymbol{\pi\psi})_K \\
& + (\boldsymbol{v}\cdot\nabla\pi\phi, \delta_K^{(\phi)}\boldsymbol{v}\cdot\nabla\pi\xi)_K\Big\} - (\boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}, \boldsymbol{n}\cdot\boldsymbol{\psi})_{\Gamma_{\text{fixed}}} \\
& - (\boldsymbol{n}\cdot\boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\sigma}(\boldsymbol{\psi}, \chi)\boldsymbol{n})_{\Gamma_{\text{fixed}}} + \beta(\boldsymbol{n}\cdot\boldsymbol{v}, \boldsymbol{n}\cdot\boldsymbol{\psi})_{-1,h,\Gamma_{\text{fixed}}}
\end{aligned}$$

with the obvious modifications if dynamic spatial meshes are used and hence the finite element spaces V_h^m do not coincide. Here, $(f, g)_{-1,h,G}$ is defined as

$$(f, g)_{-1,h,G} \coloneqq \sum_{E\subseteq G} h_E^{-1}(f, g)_E,$$

where h_E is the size of the edge or face E and $\beta > 0$ is a constant. The stabilization parameters α_K, $\delta_K^{(\boldsymbol{v})}$, and $\delta_K^{(\phi)}$ are chosen in the same way as in (3.9).

Remark 6.7. In Braack [18], a variant of the local projection stabilization for anisotropic meshes was proposed. However, in our numerical examples we did not observe any considerable difference between the isotropic and the anisotropic variant of the local projection stabilization, neither concerning the behavior of the solver nor in the solution. This can be explained due to the rather small aspect ratio of approximately 1:10.

We refrain from presenting the precise fractional-step-θ formulation of the problem because it becomes quite lengthy. The a posteriori error estimator for the spatial discretization error can be derived in the same way as in Chapter 4 taking the Lagrangian based on the fractional-step-θ time-stepping formulation as basis. Due to the consistent implementation of the free slip-boundary conditions, we obtain a similar expression for the spatial a posteriori error estimator as in Section 4.3. Again, we neglect the h-dependent terms arising due to stabilization and regularization of the Heaviside function. However, since the continuous solution does not vanish on the semi-discrete fractional-step-θ formulation, we are not able to transfer the derivation of the temporal a posteriori error estimator of Chapter 4 to this situation. On the other hand, as already pointed out above, we apply an equidistant discretization in time and hence, this does not matter.

6.2.4 Numerical results

Before we present the results of the simulation of the filling process of a prototypical microdiagnosis chip, we validate in a first step the numerical model, especially the treatment of the surface tension and the inflow and outflow boundary conditions.

Numerical validation

In this first test case, we consider a two-dimensional rectangular domain $\Omega = (0,2) \times (0,1)$, see Figure 6.28. At the beginning, the subdomain $\Omega_{\text{liquid}} = (0, \frac{1}{2}) \times (0,1)$ is filled with liquid, the rest of the domain $\Omega_{\text{gas}} = (\frac{1}{2}, 2) \times (0,1)$ is occupied by the gaseous phase. Both phases are separated by the interface $\Gamma = \{ \frac{1}{2} \} \times (0,1)$. The boundary of the domain $\partial\Omega = \Gamma_{\text{in}} \cup \Gamma_{\text{fixed}} \cup \Gamma_{\text{out}}$ is decomposed as indicated in Figure 6.28. The initial velocity is set to zero, so the whole system is at rest.

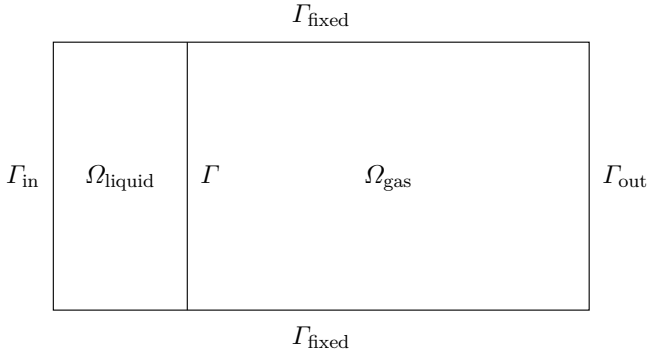


Figure 6.28. Computational domain and initial configuration

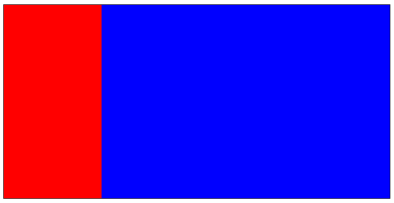
By prescribing a contact angle of $\theta = 45°$, the interface starts to get curved and a flow is developing without setting a non-trivial pressure drop between the inflow and outflow boundary. The flow makes the interface move from the left to the right. In the computation, the dimensionless parameters are set to

$$\rho_{\text{liquid}} = 1, \quad \rho_{\text{gas}} = 10^{-3}, \quad \mu_{\text{liquid}} = 1, \quad \mu_{\text{gas}} = 10^{-1}, \quad \text{We} = 1.$$

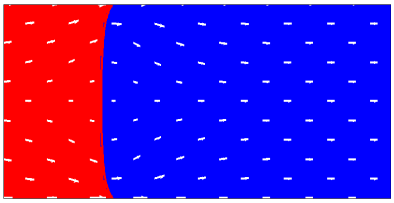
The gravitational forces are set to zero. Since we only aim at verifying the model developed in the previous section, we do not consider adaptivity here. Instead, we use a uniformly refined grid with 8192 cells yielding 33540 degrees of freedom and equidistant time steps of size $k = 0.1$. The parameter in the weak implementation of the free-slip boundary conditions is set to $\beta = 10000$. The results of the computation are shown in Figure 6.29. There, the liquid phase is shown in red while the gaseous phase is depicted in blue. The actual flow direction is also indicated by the arrows.

First of all, we observe that the developed model reproduces the expected results quite well since the prescription of the given contact angle leads to wetting of the lower and upper boundary and induces a flow in the corresponding direction. Furthermore, we also note that the applied inflow and outflow boundary conditions lead to a parallel inflow and outflow as intended.

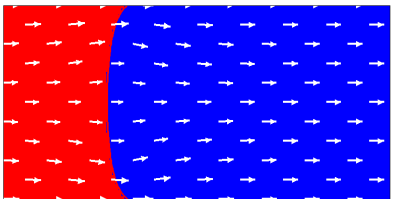
Besides this purely qualitative test case, we intend to verify the developed model also quantitatively. To this end, we consider in the second test case a stationary circular bubble



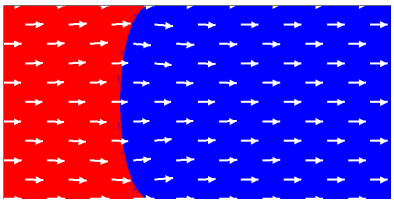
(a) $t = 0.0$



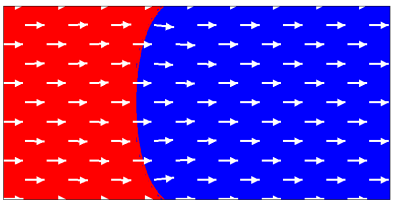
(b) $t = 0.1$



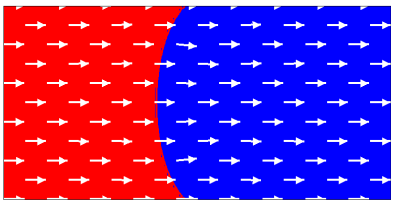
(c) $t = 0.2$



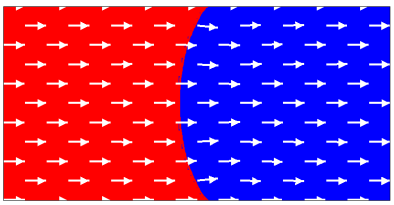
(d) $t = 0.3$



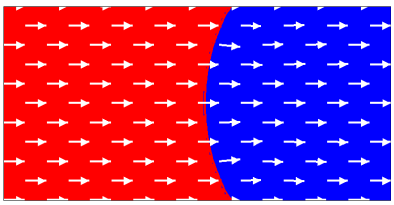
(e) $t = 0.4$



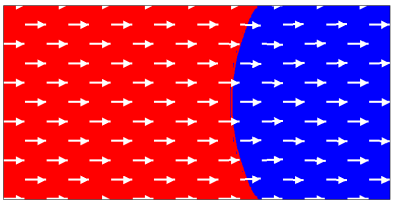
(f) $t = 0.5$



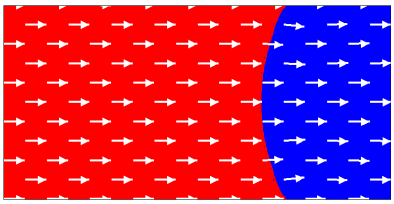
(g) $t = 0.6$



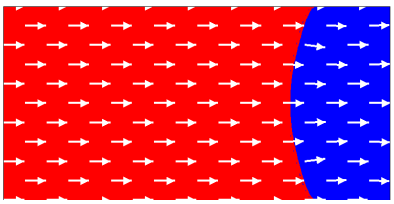
(h) $t = 0.7$



(i) $t = 0.8$



(j) $t = 0.9$



(k) $t = 1.0$

Figure 6.29. First test case: Results of the computation (red: liquid phase, blue: gaseous phase)

at equilibrium. This configuration is taken from Hysing [72]. Due to the Laplace-Young law, the pressure inside the bubble is given by

$$p_{\text{in}} = p_{\text{out}} + \frac{\sigma}{r},$$

where r denotes the radius of the circle. In our configuration, we choose $r = \frac{1}{4}$ for a circle located in the center of the unit square $\Omega = (0,1)^2$. The dimensionless densities are set to 10^4, while the viscosities are set to 1. Since the interface Γ between both phases is closed, there are no contact angles to be prescribed. Again, we do not consider gravitational forces. The Reynolds and Weber numbers are set to 1.

We prescribe free-slip boundary conditions on the whole boundary $\partial\Omega = \Gamma_{\text{fixed}}$. Hence, the pressure is sought in the space $L_{\text{slip}} \coloneqq L^2(\Omega)/\mathbb{R}$ because it is only uniquely determined up to an additive constant.

The quantity of interest is set to

$$J(\boldsymbol{u}) = p(1, \boldsymbol{x}_{\mathbf{gas}}) - p(1, \boldsymbol{x}_{\mathbf{liquid}})$$

with

$$\boldsymbol{x}_{\mathbf{gas}} = (\tfrac{1}{2}, \tfrac{1}{2})^T \quad \text{and} \quad \boldsymbol{x}_{\mathbf{liquid}} = (1, \tfrac{1}{2})^T$$

which corresponds to the difference of the pressure in the liquid and the gaseous phase at final time $T = 1$. The size of the time steps is chosen as $k = 0.1$. Since everything is stationary in this configuration, we keep the spatial meshes fixed in time.

In Table 6.16, we present the results of the computation with adaptive spatial refinement. The notation used therein is the same as in previous parts of this thesis. We again observe the very good quantitative assessment of the discretization error by the developed error estimator.

Table 6.16. Static bubble: Adaptive spatial refinement

N	η_h	$J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})$	I_{eff}
324	$-7.76 \cdot 10^{-01}$	$-2.61 \cdot 10^{+00}$	3.37
652	$3.24 \cdot 10^{-01}$	$3.52 \cdot 10^{-02}$	0.11
996	$5.65 \cdot 10^{-02}$	$7.03 \cdot 10^{-02}$	1.24
2020	$-1.64 \cdot 10^{-02}$	$9.17 \cdot 10^{-03}$	-0.56
5348	$2.12 \cdot 10^{-03}$	$7.81 \cdot 10^{-03}$	3.68
11676	$-2.71 \cdot 10^{-04}$	$-5.54 \cdot 10^{-04}$	2.04
28948	$-6.97 \cdot 10^{-05}$	$-3.07 \cdot 10^{-05}$	0.44
73820	$-2.06 \cdot 10^{-05}$	$-2.07 \cdot 10^{-05}$	1.01
208532	$-7.57 \cdot 10^{-06}$	$-7.88 \cdot 10^{-06}$	1.04

In Figure 6.30, this computation with adaptive spatial refinement is compared to a computation based on uniform refinement of the spatial mesh. In both cases, the temporal discretization is kept fixed with the size of the time steps set to $k = 0.1$. Again, the benefit of applying the quantitative error estimator can be seen: For obtaining an error of $3 \cdot 10^{-5}$,

the computational costs counted in spatial degrees of freedom can be reduced by a factor of 10 when using adaptive refinement instead of uniform refinement.

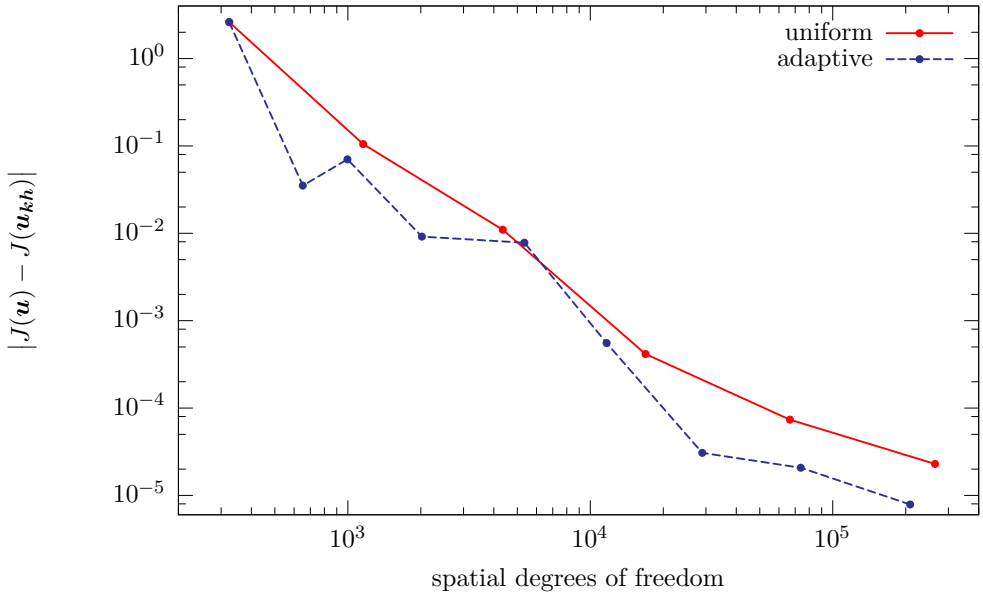


Figure 6.30. Static bubble: Comparison of the error $|J(\boldsymbol{u}) - J(\boldsymbol{u_{kh}})|$ for different refinement strategies

Figure 6.31 shows the distribution of the pressure at final time $T = 1$ and an adaptively refined mesh, corresponding to the last line in Table 6.16. We like to emphasize an additional advantage of the developed error estimator compared to heuristic error indicators: Due to the special structure of the solution, one obviously has to refine the mesh near the interface between both phases. Additionally, it is clear that refinement near the evaluation points $\boldsymbol{x}_{\mathbf{gas}}$ and $\boldsymbol{x}_{\mathbf{liquid}}$ is important. However, balancing both effects can be difficult, especially in situations where the exact solution is not known in advance. The quantitative error estimator on the other hand is able to estimate the influence of errors produced in both regions by incorporating information of the dual variable. As can be seen, the mesh is only slightly refined near the evaluation points, but strongly refined near the interface.

In the next two sections, we present the simulation of the filling process of a prototypical subregion of a lab-on-a-chip. Here, we first consider a two-dimensional simplification before we show the results in the full three-dimensional case.

Two-dimensional simplification

We consider the filling process of the prototypical two-dimensional cut-out of a microdiagnosis chip shown in Figure 6.32. The liquid phase enters the domain through the inlet Γ_{in} which is located at the lower left side of the domain. At the same time, the gaseous

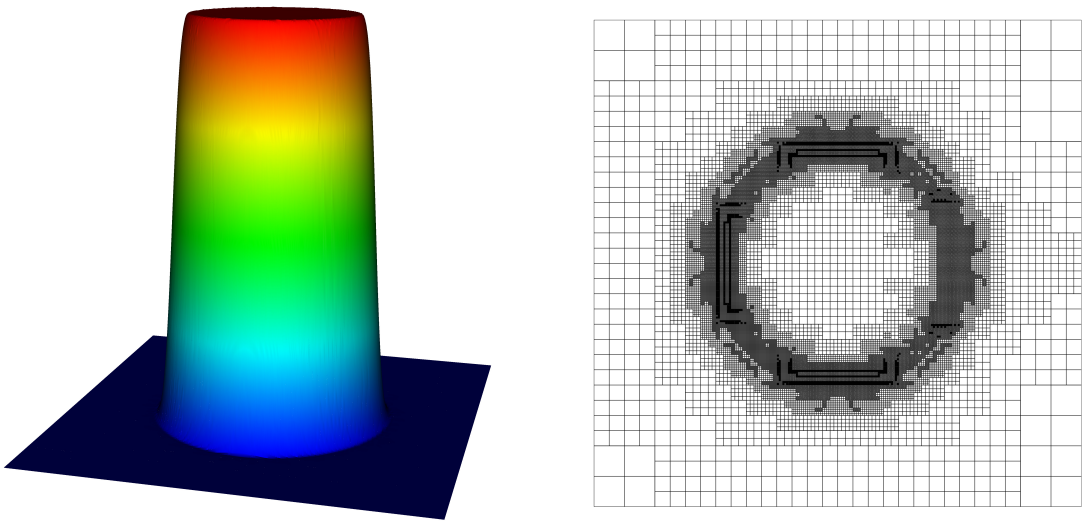


Figure 6.31. Static bubble: Pressure distribution at final time $T = 1$ and adaptively refined mesh

phase escapes through the outlet Γ_{out} situated at the upper left side of the domain. The remaining part of the boundary forms the free-slip boundary Γ_{fixed}.

For the computation, the dimensionless parameters are set to

$$\rho_{\text{liquid}} = 1, \quad \rho_{\text{gas}} = 10^{-3}, \quad \mu_{\text{liquid}} = 1, \quad \mu_{\text{gas}} = 10^{-1}, \quad \text{We} = 10.$$

The mean pressure on the inflow boundary is given by $P = 200$ while the contact angle of the interface is set to $\theta = 45°$. The constant β arising in the weak implementation of the free-slip boundary conditions is chosen as $\beta = 10000$. In this two-dimensional simplification we do not consider gravitational forces.

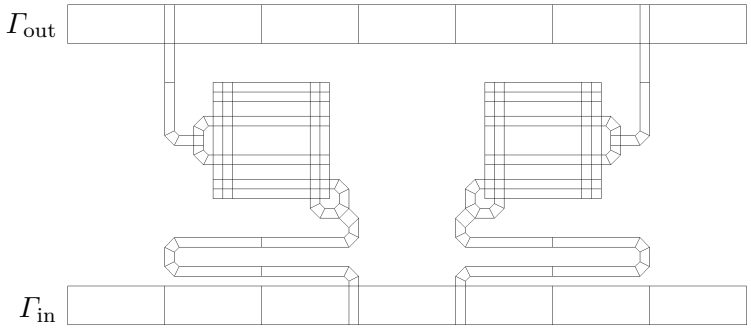


Figure 6.32. Mesh of a prototypical two-dimensional cut-out of a microdiagnosis chip

As quantity of interest, we look at the averaged volume occupied by the liquid phase

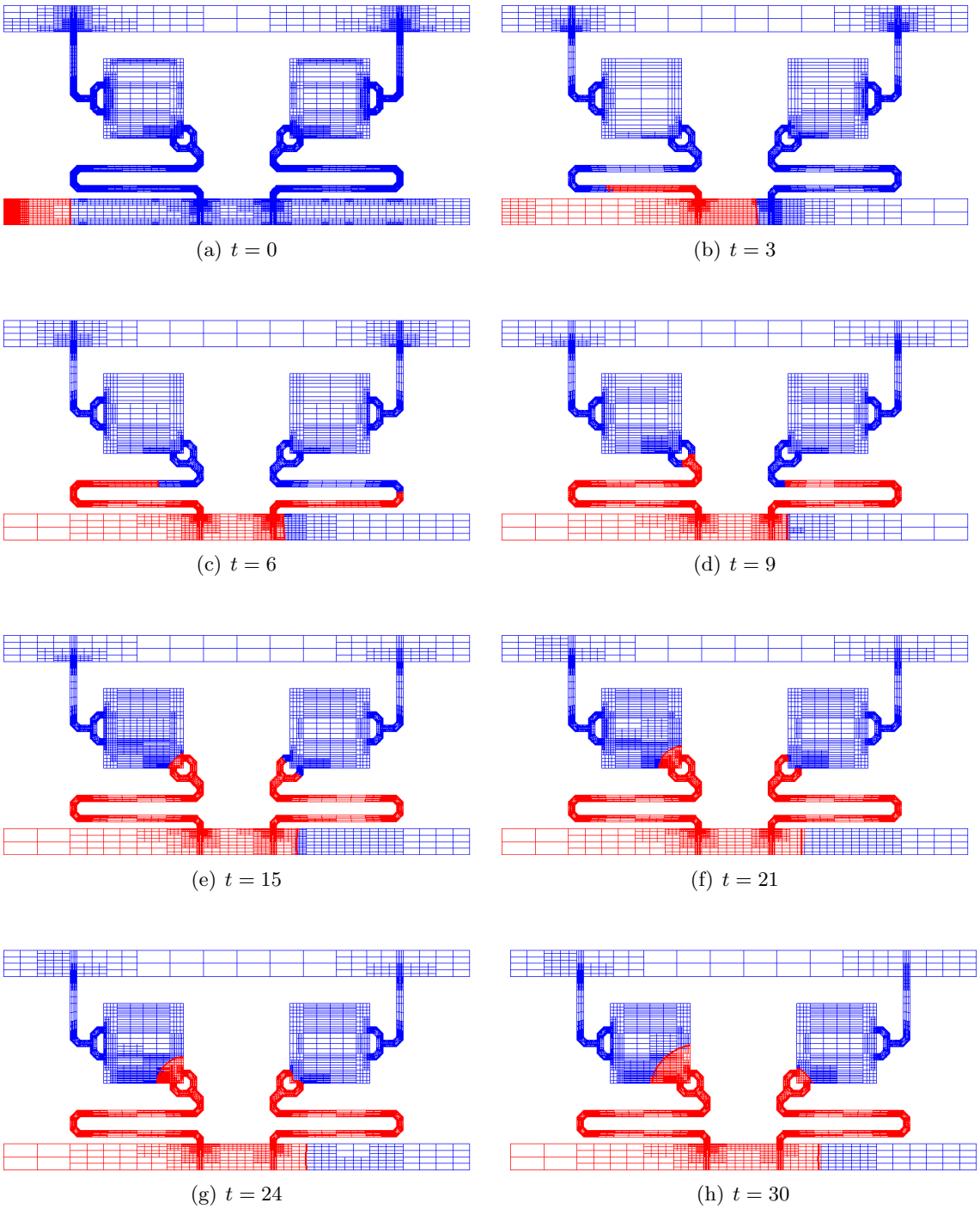
$$J(\boldsymbol{u}) = \int_0^T \int_{\Omega_{\text{liquid}}} \mathrm{d}\boldsymbol{x}\ \mathrm{d}t = \int_0^T \int_{\Omega} H(\phi)\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}t,$$

which should lead to a precise tracking of the interface between the liquid and the gaseous phase. The final time is set to $T = 60$. The size of the time steps is chosen as $k = 0.3$. For the spatial discretization, we use dynamic meshes because this allows us to capture the movement of the interface by adaptive refinement without increasing the degrees of freedom excessively.

In contrast to the two- and three-dimensional benchmark configurations "Laminar Flow Around a Cylinder" discussed in Section 6.1, we cannot give the usual plots on the convergence history here because there is no reference solution given nor are we able to perform computations on meshes fine enough to produce reliable reference values. Instead, this example has to be seen as a feasibility test which allows only for a qualitative discussion of the results.

Figure 6.33 shows the results of the simulation of the filling process by presenting a sequence of filling states at different times. We observe that the meshes used at the beginning of the simulation are stronger refined than those used at the end. This, of course, is meaningful because errors produced at early times have a greater influence on the overall discretization error than those occurring at later times. We also note another phenomenon which might seem surprising at first thought: The application of the developed a posteriori error estimator leads to refinement of cells which are far away from the current interface position. However, a closer look shows that the areas of strong refinement are always located near corners. This helps understanding why refining these areas is important: The propagation speed of the interface in such a two-phase flow setting is mainly determined by the pressure distribution in the whole system. The reentrant corners lead to singularities in the pressure which have to be captured by adaptive refinement of the cells to obtain a good approximation of the overall pressure distribution. Hence, for precisely tracking the interface one must on the one hand refine the cells in the neighborhood of the interface in order to be able to precisely represent the shape of the interface. On the other hand, however, one also has to refine the areas near corners in order to resolve the pressure singularities which mainly determine the propagation of the interface. Balancing both criteria is hardly possible without the information of an a posteriori error estimator. For example, using only information from the level set function—like it is usually done in two-phase flow simulations—would lead to refinement around the interface, but then the propagation speed of the interface might be wrong because the overall flow pattern is not computed correctly.

To substantiate this statement, we repeat the simulation of the filling process, but this time we do not apply the developed a posteriori error estimator to trigger the adaptive refinement of the spatial discretization. Instead, a cell is marked to be refined if the level set function ϕ takes positive as well as negative values on it. Since the interface between both phases is given by the zero level set of ϕ, this corresponds to refining those cells which



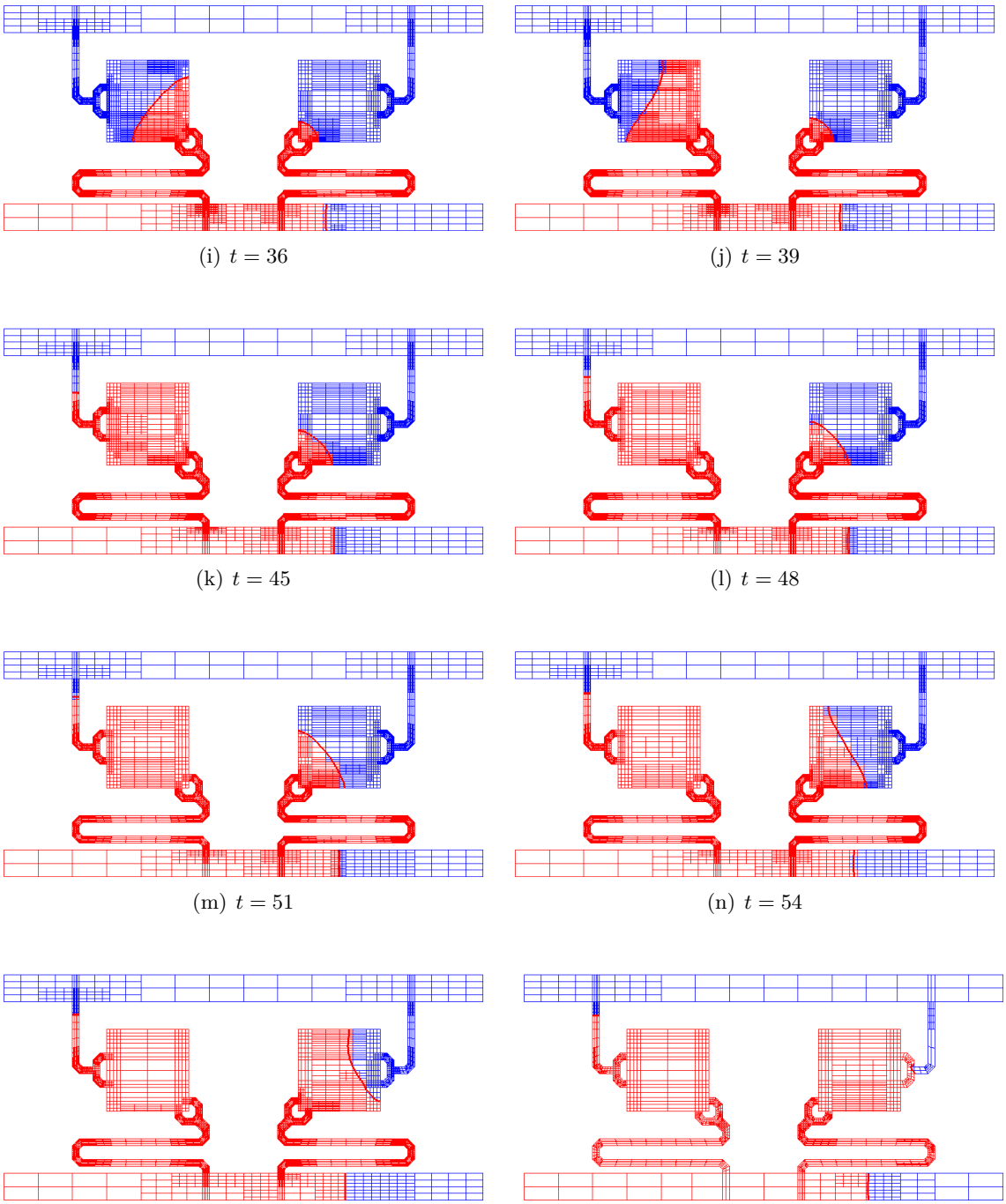
(a) $t = 0$ (b) $t = 3$

(c) $t = 6$ (d) $t = 9$

(e) $t = 15$ (f) $t = 21$

(g) $t = 24$ (h) $t = 30$

Figure 6.33. Filling process of the two-dimensional configuration (red: liquid phase, blue: gaseous phase)



(i) $t = 36$ (j) $t = 39$

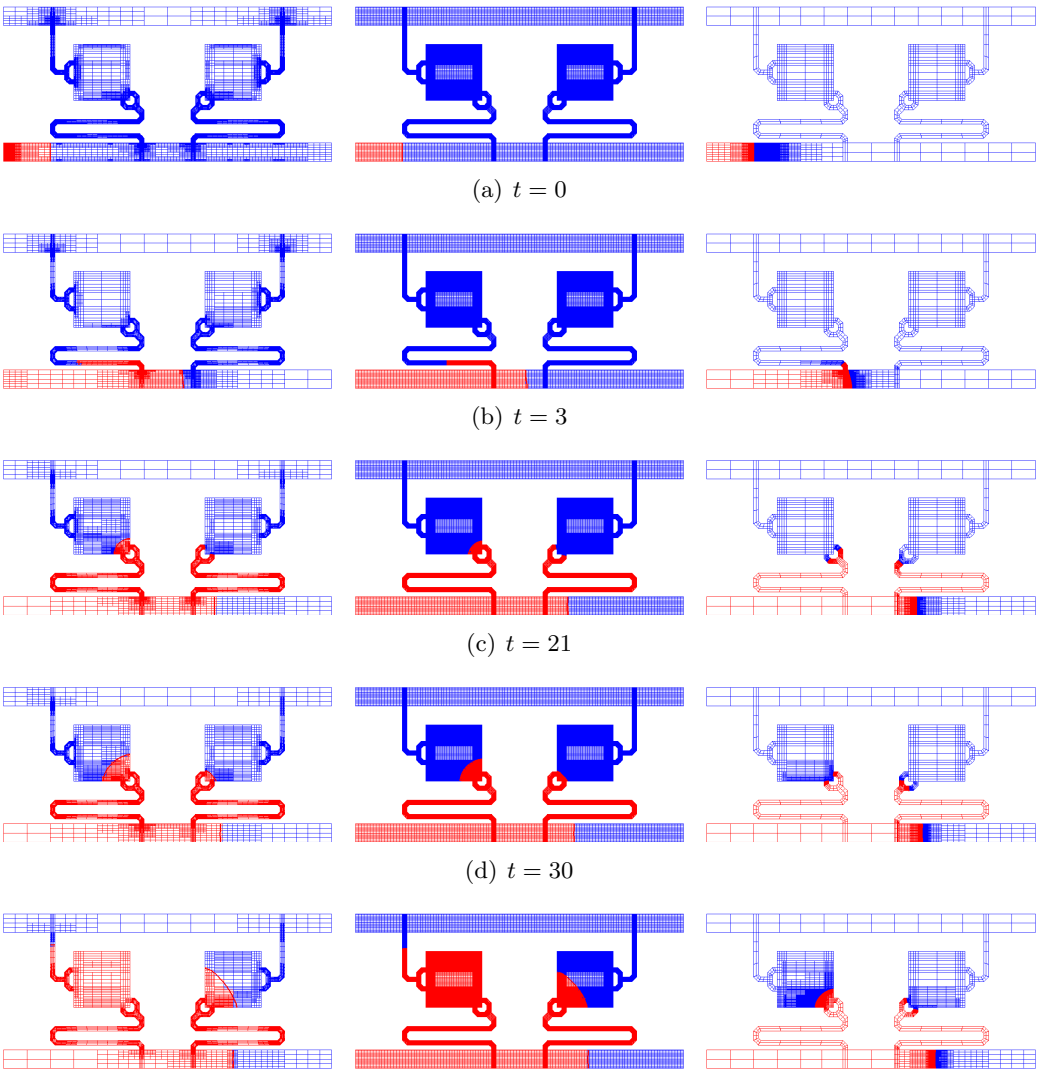
(k) $t = 45$ (l) $t = 48$

(m) $t = 51$ (n) $t = 54$

(o) $t = 57$ (p) $t = 60$

Figure 6.33. Filling process of the two-dimensional configuration (red: liquid phase, blue: gaseous phase) (cont.)

are crossed by the interface. In Figure 6.34, we show a comparison of the filling states computed either with uniform refinement, with adaptive refinement based on the developed a posteriori error estimator or with adaptive refinement based on the information of the level set function. We observe a pretty good agreement of the computations involving uniform and adaptive refinement based on the a posteriori error estimator. On the other hand, the computation with adaptive refinement based on the information of the level set function leads to a completely different propagation speed of the interface between the liquid and the gaseous phase.



(a) $t = 0$

(b) $t = 3$

(c) $t = 21$

(d) $t = 30$

(e) $t = 51$

Figure 6.34. Comparison of the filling states obtained with uniform refinement (middle), adaptive refinement based on the a posteriori error estimator (left) and information of the level set function (right)

Three-dimensional results

In this section, we consider the filling process of a prototypical three-dimensional cut-out of a microdiagnosis chip as it is shown in Figure 6.35. Here, the inlet boundary Γ_{in} is given as the front plane of the right main channel, while the outlet boundary Γ_{out} is represented by the front plane of the left main channel.

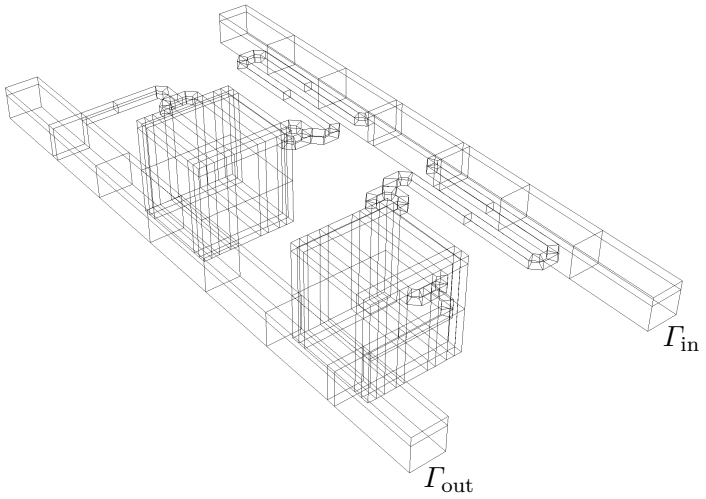


Figure 6.35. Mesh of a prototypical three-dimensional cut-out of a microdiagnosis chip

For the computation, the dimensionless parameters are set to

$$\rho_{\text{liquid}} = 1, \quad \rho_{\text{gas}} = 10^{-3}, \quad \mu_{\text{liquid}} = 1, \quad \mu_{\text{gas}} = 10^{-1}, \quad \text{Fr} = \frac{1}{98.1}.$$

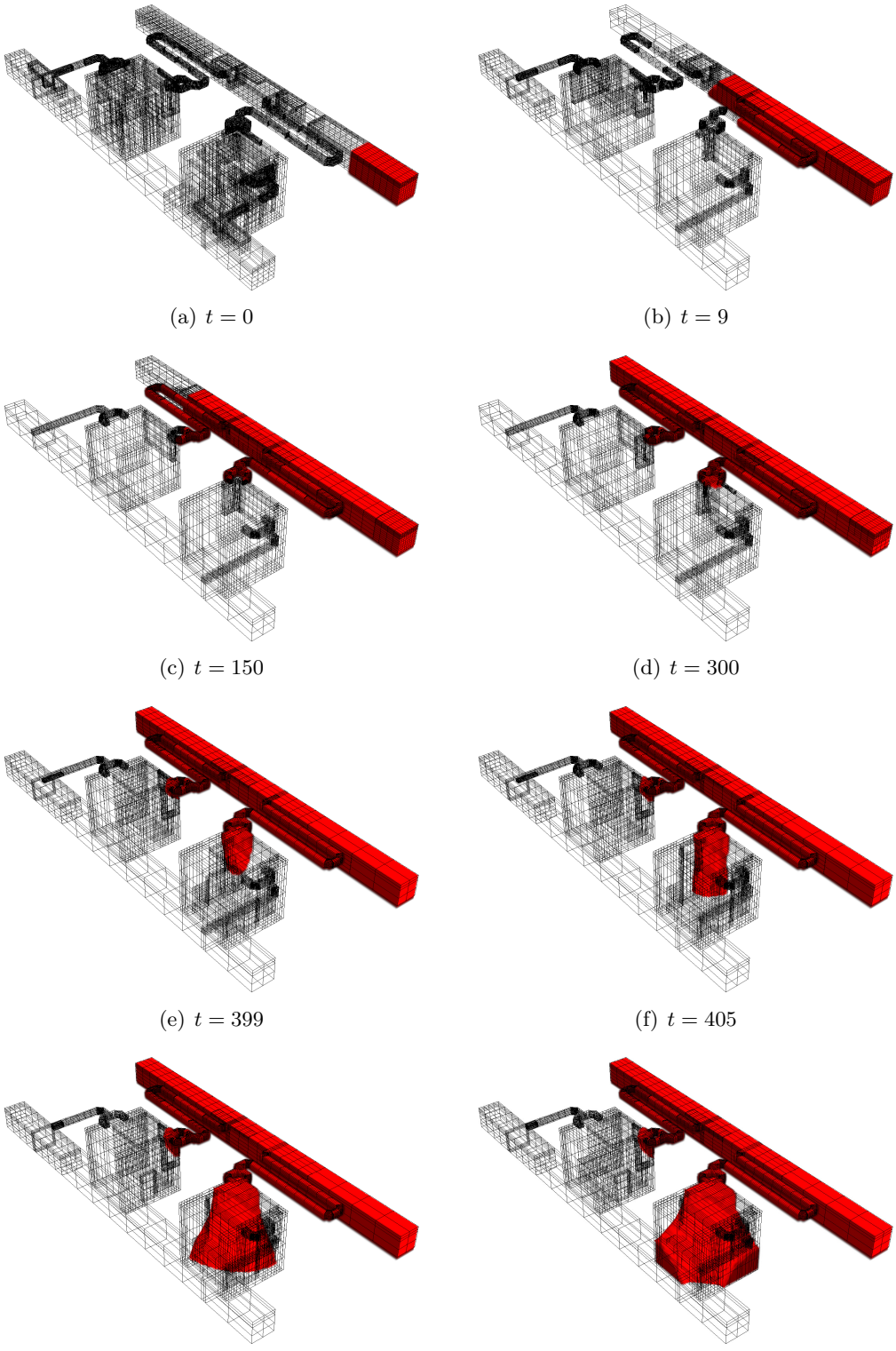
Here, the flow is driven only by the pressure drop between the inflow and outflow boundary where the mean pressure on the inflow boundary is given by $P = 400$. As in the two-dimensional case, the constant β arising in the weak implementation of the free-slip boundary conditions is chosen as $\beta = 10000$.

As quantity of interest, we look again at the averaged volume occupied by the liquid phase

$$J(\boldsymbol{u}) = \int_0^T \int_{\Omega_{\text{liquid}}} \mathrm{d}\boldsymbol{x}\ \mathrm{d}t = \int_0^T \int_{\Omega} H(\phi)\ \mathrm{d}\boldsymbol{x}\ \mathrm{d}t.$$

The final time is set to $T = 600$. The size of the time steps is chosen as $k = 1$. As in the two-dimensional case, dynamic meshes are used for the spatial discretization.

In Figure 6.36, the results of the simulation of the filling process are shown. There, we present a sequence of filling states at different times. We observe that first the liquid phase which enters the domain through the inlet Γ_{in} passes through the main channel. Then, the flow separates and the thin channels are filled. Finally, under the influence of gravity,



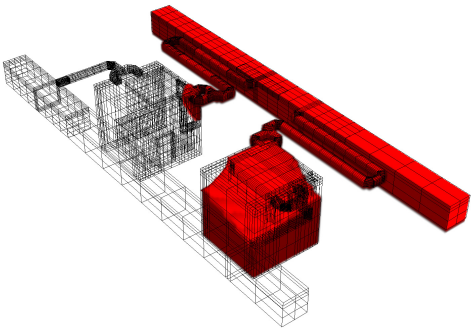
(a) $t = 0$ (b) $t = 9$

(c) $t = 150$ (d) $t = 300$

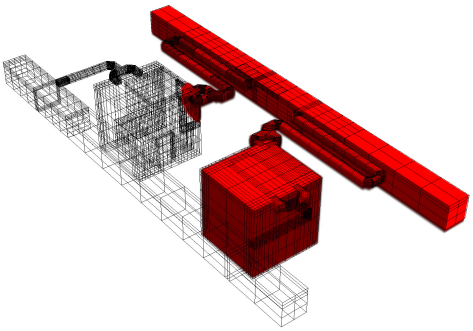
(e) $t = 399$ (f) $t = 405$

(g) $t = 414$ (h) $t = 417$

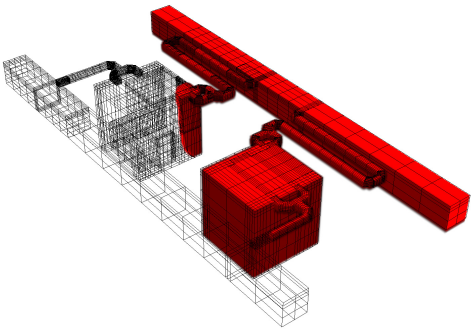
Figure 6.36. Filling process of the three-dimensional configuration



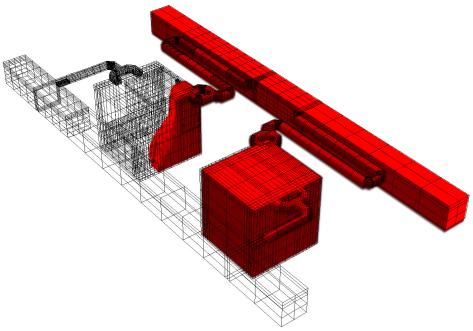
(i) $t = 420$



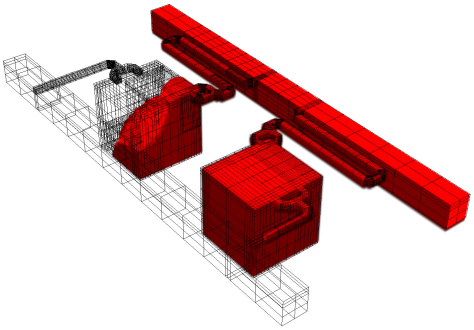
(j) $t = 423$



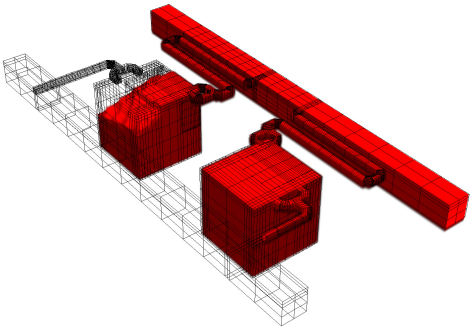
(k) $t = 441$



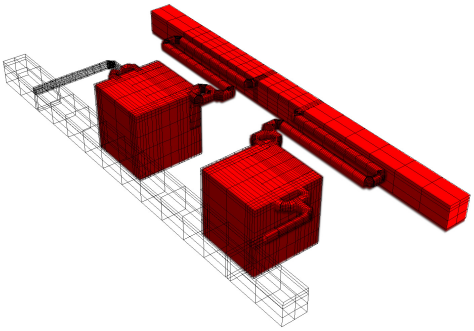
(l) $t = 450$



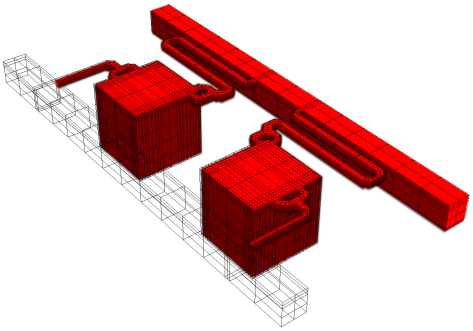
(m) $t = 456$



(n) $t = 459$



(o) $t = 462$



(p) $t = 600$

Figure 6.36. Filling process of the three-dimensional configuration (cont.)

the liquid flows into the reaction chambers and fills them bottom up. This shows that the model developed in Section 6.2.1 is well suited for the simulation of such filling processes.

As in the two-dimensional case, the meshes used at the beginning of the time interval are stronger refined than those used at the end. Since the propagation speed of the interface between the liquid and the gaseous phase is mainly determined by the overall pressure distribution, the zones of strong refinement are not only concentrated to the interface. Instead, the meshes are also refined in the neighborhood of corners. This is necessary to resolve the singularities arising in the pressure. As already pointed out in the two-dimensional case, this is essential in order to obtain a meaningful approximation of the overall pressure distribution.

The last example shows that even for such complex "real-life" problems, the a posteriori error estimator developed in this thesis is a powerful tool in producing efficient discretizations based on dynamic meshes.

7 Conclusion and Outlook

In this thesis, we developed an efficient numerical method for solving nonstationary incompressible flow problems. The proposed method uses Galerkin finite element discretizations in time and space. Based on these discretizations, we derived a posteriori error estimators which are able to separate and quantitatively assess the temporal and spatial discretization errors measured in some quantity of interest. The information from the error estimators was then used within an adaptive algorithm which balances the error contributions of both the temporal and the spatial discretization and keeps both discretization errors equilibrated under further adaptive refinement of the discretizations. To this end, the information from the error estimators was localized to interval-wise or cell-wise contributions. For the proposed adaptive algorithm, an important property is the quantitative reliability of the error estimators as well as their independence of the refinement of the other discretization.

We also introduced the concept of dynamic spatial meshes, i. e., meshes that change in time. We showed that the approximation of the velocity field on such dynamic meshes is satisfactory while the approximation of the pressure deteriorates when changing the spatial mesh. We numerically and theoretically investigated this phenomenon and showed that this error in the pressure approximation occurs if the velocity from the first mesh is not divergence-free with respect to the test functions on the second mesh. We also numerically investigated several attempts to overcome this difficulty where it turned out that using the divergence-free L^2-projection (H-projection) of the velocity from the first mesh onto the second mesh performed best.

The efficiency of the developed numerical method was demonstrated by applying it to academic test cases as well as well-known benchmark configurations. In both cases, we achieved a significant saving considering the required degrees of freedom and hence also in the time needed for computation. We were even able to compute the mean drag-coefficient in the three-dimensional benchmark configuration up to a few percent on a standard personal computer.

Finally, we also considered two-phase flow problems. We developed a model for simulating the filling process of a microdiagnosis chip incorporating surface tension and wetting effects as well as contact angles. For separating both phases, a level set approach was used. We showed that the application of the developed a posteriori error estimator is advantageous compared to simply refining cells near the interface—as it is often done in two-phase flow simulations—because the latter method possibly leads to wrong propagation speeds because the flow pattern away from the interface might not be resolved accurately.

Based on the results achieved in this thesis, the following topics can be regarded as promising future development:

Optimal Control of Nonstationary Flow Problems

In this thesis, we considered only the efficient simulation of nonstationary incompressible flow problems. However, in practice, one might be interested in minimizing specific quantities (for example, the drag coefficient of an obstacle) or in estimating some parameter (like the viscosity, for instance). We regard the combination of the results of this thesis with the concepts derived in Meidner [78] for optimal control problems with parabolic partial differential equations as a promising approach to optimal control problems involving nonstationary flows.

Space-Time-Adaptivity for Two-Phase Flow Problems

We showed how residual based a posteriori error estimation is applicable to two-phase flow problems and what benefit we can get from it. However, in this thesis, the error estimation and adaptive refinement for two-phase flow problems was restricted to the spatial discretization. For setting up efficient adaptive algorithms on the other hand, one has to cover the estimation of the temporal discretization error as well. One important point here is posing a coupled space-time-variational formulation of the governing equations. To this end, it is necessary to replace the level set method by a method which does not require intermediate reinitialization steps like the *initial position set* method developed by Dunne [38] in the context of fluid-structure-interaction problems.

Nonstationary Reactive Flow Problems

In industrial applications of multi-phase flow problems, there are often chemical reactions between some of the phases. The discretization of the governing system of equations leads to extremely large and complex systems of algebraic equations. Hence, to accurately resolve the reactions and the overall flow pattern, the use of adaptive refinement techniques is mandatory. Combining the methods developed in this thesis with those presented in Braack [17] and Richter [93] for stationary reactive flows seems to be a promising way to cope with nonstationary reactive flow problems. Especially in three spatial dimensions, even with adaptive discretizations the computational costs will usually be too high to solve the problems on standard personal computers. Thus, employing parallel computing techniques is a fundamental topic here.

Acknowledgments

First of all, I want to express my gratitude to my supervisor Rolf Rannacher for giving me the opportunity to work on this interesting subject. I would also like to thank him for always having a sympathetic ear, for his enduring support during the last years, and for pointing out the essential objectives of this work. Furthermore, I wish to thank him for the encouragement and the possibility to attend conferences—at home and abroad—in order to present some results of my work, as well as for the help in gaining experience in research and teaching.

I gratefully acknowledge the funding received towards my Ph.D. from the *German Research Foundation (DFG)* through the *Collaborative Research Centre (SFB) 359 "Reactive Flows, Diffusion and Transport"* and from the *Federal Ministry of Education and Research (BMBF)* through the project 03RANBHD of the funding program *"Mathematik für Innovationen in Industrie und Dienstleistungen"*.

I would especially like to thank Dominik Meidner, with whom I have shared my office, for countless fruitful discussions and for always being open to any questions, be it on mathematical topics or other things.

Furthermore, I want to thank Boris Vexler for his collaboration, particularly in paper writing. For many helpful ideas, I also wish to thank Malte Braack, Thomas Dunne, Bärbel Janssen, Stefan Knauf, Thomas Richter, Thomas Wick, and especially Winnifried Wollner.

All numerical computations in this work would not have been possible without the software package *Gascoigne*. Hence, I want to thank the whole team: Roland Becker, Malte Braack, Thomas Dunne, Dominik Meidner, Thomas Richter, Boris Vexler, and Winnifried Wollner. Furthermore, some computations in this thesis required the software package *deal.II*. Therefore, I would like to thank its development team Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat. Special thanks also go to Bärbel Janssen for always offering help with this finite element software package.

I thank the whole Numerical Analysis Group at the University of Heidelberg for always open doors and the nice atmosphere making the time there so enjoyable.

Last but by no means least, I want to thank my parents and Kristin Besier for their never-ending and encouraging support.

List of Tables

List of Figures

List of Algorithms

Bibliography

[1] G. Akrivis, C. Makridakis, and R. H. Nochetto. A posteriori error estimates for the Crank-Nicolson method for parabolic equations. *Math. Comp.*, 75(254), pp. 511–531, 2005.

[2] H. W. Alt. *Lineare Funktionalanalysis*. Springer-Verlag, Berlin Heidelberg New York, third edition, 1999.

[3] J. P. Aubin. Behavior of the error of the approximate solutions of boundary value problems for linear elliptic operators by Galerkin's and finite difference methods. *Ann. Scuola Norm. Sup. Pisa Cl. Sci. (3)*, 21(4), pp. 599–637, 1967.

[4] A. K. Aziz and P. Monk. Continuous finite elements in space and time for the heat equation. *Math. Comp.*, 52(186), pp. 255–274, 1989.

[5] W. Bangerth. *Adaptive Finite-Elemente-Methoden zur Lösung der Wellengleichung mit Anwendung in der Physik der Sonne*. Diploma thesis, Universität Heidelberg, Heidelberg, 1998.

[6] E. Bänsch. An adaptive finite-element strategy for the three-dimensional time-dependent Navier-Stokes equations. *J. Comput. Appl. Math.*, 36(1), pp. 3–28, 1991.

[7] E. Bänsch. Finite element discretization of the Navier-Stokes equations with a free capillary surface. *Numer. Math.*, 88(2), pp. 203–235, 2001.

[8] R. Becker. *Adaptive Finite Elements for Optimal Control Problems*. Habilitation thesis, Universität Heidelberg, 2001.

[9] R. Becker and M. Braack. Multigrid techniques for finite elements on locally refined meshes. *Numer. Linear Algebra Appl.*, 7(6), pp. 363–379, 2000.

[10] R. Becker and M. Braack. A modification of the least-squares stabilization for the Stokes equations. *Calcolo*, 38(4), pp. 173–199, 2001.

[11] R. Becker and M. Braack. A two-level stabilization scheme for the Navier-Stokes equations. In *Numerical Mathematics and Advanced Applications. ENUMATH 2003*, edited by M. Feistauer, V. Dolejší, P. Knobloch, and K. Najzar, pp. 123–130. Springer-Verlag, Heidelberg, 2004.

[12] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: Basic analysis and examples. *East-West J. Numer. Math.*, 4, pp. 237–264, 1996.

[13] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. In *Acta Numerica 2001*, edited by A. Iserles, volume 10, pp. 1–102. Cambridge University Press, Cambridge, 2001.

[14] R. Becker and B. Vexler. A posteriori error estimation for finite element discretization of parameter identification problems. *Numerische Mathematik*, 96(3), pp. 435–459, 2004.

[15] M. Behr and T. E. Tezduyar. Finite element solution strategies for large-scale flow simulations. *Comput. Methods Appl. Mech. Engrg.*, 112(1–4), pp. 3–24, 1994.

[16] M. Berggren, R. Glowinski, and J. L. Lions. A computational approach to controllability issues for flow-related models. (I): Pointwise control of the viscous Burgers equation. *Int. J. Comput. Fluid Dyn.*, 7(3), pp. 237–252, 1996.

[17] M. Braack. *An Adaptive Finite Element Method for Reactive Flow Problems*. Ph.D. thesis, Universität Heidelberg, Heidelberg, 1998.

[18] M. Braack. A stabilized finite element scheme for the Navier-Stokes equations on quadrilateral anisotropic meshes. *M2AN Math. Model. Numer. Anal.*, 42(6), pp. 903–924, 2008.

[19] M. Braack, E. Burman, V. John, and G. Lube. Stabilized finite element methods for the generalized Oseen equations. *Comput. Methods Appl. Mech. Engrg.*, 196(4–6), pp. 853–866, 2007.

[20] M. Braack and A. Ern. A posteriori control of modeling errors and discretization errors. *Multiscale Model. Simul.*, 1(2), pp. 221–238, 2003.

[21] M. Braack and T. Richter. Solutions of 3D Navier–Stokes benchmark problems with adaptive finite elements. *Comput. & Fluids*, 35(4), pp. 372–392, 2006.

[22] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, 100(2), pp. 335–354, 1992.

[23] D. Braess. *Finite Elemente. Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*. Springer-Verlag, Berlin Heidelberg New York, second edition, 1997.

[24] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, third edition, 2008.

[25] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York Berlin Heidelberg, 1991.

[26] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1–3), pp. 199–259, 1982.

[27] G. F. Carey and J. T. Oden. *Finite Elements. Volume III. Computational Aspects.* The Texas Finite Element Series. Prentice-Hall, Inc., Englewood Cliffs, 1984.

[28] C. Carstensen and R. Verfürth. Edge residuals dominate a posteriori error estimates for low order finite element methods. *SIAM J. Numer. Anal.*, 36(5), pp. 1571–1587, 1999.

[29] L. Cattabriga. Su un problema al contorno relativo al sistema di equazioni di Stokes. *Rend. Sem. Mat. Univ. Padova*, 31, pp. 308–340, 1961.

[30] Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher. A level set formulation of eulerian interface capturing methods for incompressible fluid flows. *J. Comput. Phys.*, 124(2), pp. 449–464, 1996.

[31] A. J. Chorin and J. E. Marsden. *A Mathematical Introduction to Fluid Mechanics*, volume 4 of *Texts in Applied Mathematics*. Springer-Verlag, New-York Berlin Heidelberg, third edition, 2000.

[32] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland Publishing Company, Amsterdam New York Oxford, first edition, 1987.

[33] P. Clément. Approximation by finite element functions using local regularization. *RAIRO Anal. Numér.*, 9(R-2), pp. 77–84, 1975.

[34] M. Dauge. Stationary Stokes and Navier-Stokes systems on two- or three-dimensional domains with corners. Part I: Linearized equations. *SIAM J. Math. Anal.*, 20(1), pp. 74–97, 1989.

[35] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5. Evolution Problems I.* Springer-Verlag, Berlin Heidelberg, 2000.

[36] deal.II. A Finite Element Differential Equations Analysis Library. URL `http://www.dealii.org/`.

[37] M. Delfour, W. Hager, and F. Trochu. Discontinuous Galerkin methods for ordinary differential equations. *Math. Comp.*, 36(154), pp. 455–473, 1981.

[38] T. Dunne. *Adaptive Finite Element Approximation of Fluid-Structure Interaction Based on Eulerian and Arbitrary Lagrangian-Eulerian Variational Formulations.* Ph.D. thesis, Universität Heidelberg, Heidelberg, 2007.

[39] G. Dziuk. An algorithm for evolutionary surfaces. *Numer. Math.*, 58(6), pp. 603–611, 1991.

[40] D. Erickson. Towards numerical prototyping of labs-on-chip: Modeling for integrated microfluidic devices. *Microfluid. Nanofluid.*, 1(4), pp. 301–318, 2005.

[41] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. Introduction to adaptive methods for differential equations. In *Acta Numerica 1995*, edited by A. Iserles, volume 4, pp. 105–158. Cambridge University Press, Cambridge, 1995.

[42] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational Differential Equations*. Cambridge University Press, 1996.

[43] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems. I: A linear model problem. *SIAM J. Numer. Anal.*, 28(1), pp. 43–77, 1991.

[44] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems. II: Optimal error estimates in $L_\infty L_2$ and $L_\infty L_\infty$. *SIAM J. Numer. Anal.*, 32(3), pp. 706–740, 1995.

[45] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems. IV: Nonlinear problems. *SIAM J. Numer. Anal.*, 32(6), pp. 1729–1749, 1995.

[46] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems. V: Long-time integration. *SIAM J. Numer. Anal.*, 32(6), pp. 1750–1763, 1995.

[47] K. Eriksson, C. Johnson, and S. Larsson. Adaptive finite element methods for parabolic problems. VI: Analytic semigroups. *SIAM J. Numer. Anal.*, 35(4), pp. 1315–1325, 1998.

[48] K. Eriksson, C. Johnson, and V. Thomée. Time discretization of parabolic problems by the discontinuous Galerkin method. *M2AN Math. Model. Numer. Anal.*, 19(4), pp. 611–643, 1985.

[49] D. Estep and D. French. Global error control for the continuous Galerkin finite element method for ordinary differential equations. *M2AN Math. Model. Numer. Anal.*, 28(7), pp. 815–852, 1994.

[50] L. C. Evans and J. Spruck. Motion of level sets by mean curvature. II. *Trans. Amer. Math. Soc.*, 330(1), pp. 321–332, 1992.

[51] M. Feistauer. *Mathematical Methods in Fluid Dynamics*, volume 67 of *Pitman Monographs and Surveys in Pure and Applied Mathematics*. Longman Scientific & Technical, Harlow, first edition, 1993.

[52] L. P. Franca and S. L. Frey. Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 99(2–3), pp. 209–233, 1992.

[53] J. Freund and R. Stenberg. On weakly imposed boundary conditions for second order problems. In *Proceedings of the International Conference on Finite Elements in Fluids. New Trends and Applications*, edited by M. M. Cecchi, K. Morgan, J. Périaux, B. Schrefler, and O. C. Zienkiewicz, pp. 327–336. Dipartimento di Matematica Pura ed Applicata, Università di Padova, Venezia, 1995.

[54] S. Gallot, D. Hulin, and J. Lafontaine. *Riemannian Geometry*. Springer-Verlag, Berlin Heidelberg New York, third edition, 2004.

[55] S. Ganesan and L. Tobiska. Finite element simulation of a droplet impinging a horizontal surface. In *ALGORITMY 2005. Proceedings of the 17th Conference on Scientific Computing*, edited by A. Handlovičová, Z. Krivá, K. Mikula, and D. Ševčovič, pp. 1–11. Slovak University of Technology, Bratislava, 2005.

[56] Gascoigne. High Performance Adaptive Finite Element Toolkit. URL `http://www.gascoigne.uni-hd.de/`.

[57] M. Giles, M. G. Larson, J. M. Levenstam, and E. Süli. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow. Technical Report NA-97/06, Oxford University Computing Laboratory, Oxford, 1997.

[58] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations. Theory and Algorithms*, volume 5 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin Heidelberg New York Tokyo, 1986.

[59] A. Griewank. Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optim. Methods Softw.*, 1(1), pp. 35–54, 1992.

[60] P. Grisvard. *Elliptic Problems in Nonsmooth Domains*, volume 24 of *Monographs and Studies in Mathematics*. Pitman Advanced Publishing Program, Boston, 1985.

[61] S. Gross. *Numerical Methods for Three-Dimensional Incompressble Two-Phase Flow Problems*. Ph.D. thesis, RWTH Aachen University, Aachen, 2008.

[62] S. Gross, V. Reichelt, and A. Reusken. A finite element based level set method for two-phase incompressible flows. *Comput. Vis. Sci.*, 9(4), pp. 239–257, 2006.

[63] W. Hackbusch. *Iterative Lösung großer schwachbesetzter Gleichungssysteme*, volume 69 of *Leitfäden der angewandten Mathematik und Mechanik*. Teubner, Stuttgart, second edition, 1993.

[64] R. Hartmann. *A-posteriori Fehlerschätzung und adaptive Schrittweiten- und Ortsgittersteuerung bei Galerkin-Verfahren für die Wärmeleitungsgleichung*. Diploma thesis, Universität Heidelberg, Heidelberg, 1998.

[65] J. G. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier-Stokes problem. I: Regularity of solutions and second-order error estimates for spatial discretization. *SIAM J. Numer. Anal.*, 19(2), pp. 275–311, 1982.

[66] J. G. Heywood and R. Rannacher. Finite element approximation of the nonstationary Navier-Stokes problem. Part IV: Error analysis for second-order time discretization. *SIAM J. Numer. Anal.*, 27(2), pp. 353–384, 1990.

[67] J. G. Heywood, R. Rannacher, and S. Turek. Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations. *Internat. J. Numer. Methods Fluids*, 22(5), pp. 325–352, 1996.

[68] J. Hoffman. Efficient computation of mean drag for the subcritical flow past a circular cylinder using General Galerkin G2. *Internat. J. Numer. Methods Fluids*, 2008. To appear.

[69] P. Hood and C. Taylor. Navier-Stokes equations using mixed interpolation. In *Finite Element Methods in Flow Problems*, edited by J. T. Oden, O. C. Zienkiewicz, R. H. Gallagher, and C. Taylor, pp. 121–132. University of Alabama, Huntsville, 1974.

[70] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Comput. Methods Appl. Mech. Engrg.*, 59(1), pp. 85–99, 1986.

[71] S.-R. Hysing. A new implicit surface tension implementation for interfacial flows. *Internat. J. Numer. Methods Fluids*, 51(6), pp. 659–672, 2006.

[72] S.-R. Hysing. *Numerical Simulation of Immiscible Fluids with FEM Level Set Techniques*. Ph.D. thesis, Universität Dortmund, Dortmund, 2007.

[73] V. John. Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder. *Internat. J. Numer. Methods Fluids*, 44(7), pp. 777–788, 2004.

[74] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press, Cambridge, 1987.

[75] C. Johnson. Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations. *SIAM J. Numer. Anal.*, 25(4), pp. 908–926, 1988.

[76] R. B. Kellogg and J. E. Osborn. A regularity results for the Stokes problem in a convex polygon. *J. Funct. Anal.*, 21(4), pp. 397–431, 1976.

[77] M. Luskin and R. Rannacher. On the smoothing property of the Crank-Nicolson scheme. *Appl. Anal.*, 14(2), pp. 117–135, 1982.

[78] D. Meidner. *Adaptive Space-Time Finite Element Methods for Optimization Problems Governed by Nonlinear Parabolic Systems*. Ph.D. thesis, Universität Heidelberg, Heidelberg, 2007.

[79] S. Mittal, A. Ratner, D. Hastreiter, and T. E. Tezduyar. Space-time finite element computation of incompressible flows with emphasis on flows involving oscillating cylinders. *International Video Journal of Engineering Research*, 1(1), pp. 83–96, 1991.

[80] S. Mittal and T. E. Tezduyar. Notes on the stabilized space-time finite-element formulation of unsteady incompressible flows. *Comput. Phys. Comm.*, 73(1–3), pp. 93–112, 1992.

[81] S. Müller-Urbaniak. Eine Analyse des Zwischenschritt-θ-Verfahrens zur Lösung der instationären Navier-Stokes-Gleichungen. Preprint 94-01, Universität Heidelberg, Heidelberg, 1994.

[82] G. Nabh. *On High Order Methods for the Stationary Incompressible Navier-Stokes Equations*. Ph.D. thesis, Universität Heidelberg, Heidelberg, 1998.

[83] S. Nagrath, K. E. Jansen, and R. T. Lahey Jr. Computation of incompressible bubble dynamics with a stabilized finite element level set method. *Comput. Methods Appl. Mech. Engrg.*, 194(42-44), pp. 4565–4587, 2005.

[84] D. N'dri, A. Garon, and A. Fortin. Incompressible Navier-Stokes computations with stable and stabilized space-time formulations: A comparative study. *Comm. Numer. Methods Engrg.*, 18(7), pp. 495–512, 2002.

[85] J. Nitsche. Ein Kriterium für die Quasi-Optimalität des Ritzschen Verfahrens. *Numer. Math.*, 11, pp. 346–348, 1968.

[86] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ. Hamburg*, 36(1), pp. 9–15, 1971.

[87] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *J. Comput. Phys.*, 210(1), pp. 225–246, 2005.

[88] E. Olsson, G. Kreiss, and S. Zahedi. A conservative level set method for two phase flow II. *J. Comput. Phys.*, 225(1), pp. 785–807, 2007.

[89] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, Berlin, Heidelberg, 2003.

[90] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1), pp. 12–40, 1988.

[91] R. Rannacher. On the stabilization of the Crank-Nicolson scheme for long time calculations. Preprint, Unversität des Saarlandes, Saarbrücken, 1986.

[92] T. Richter. *Funktionalorientierte Gitteroptimierung bei der Finite-Elemente-Approximation elliptischer Differentialgleichungen*. Diploma thesis, Universität Heidelberg, Heidelberg, 2001.

[93] T. Richter. *Parallel Multigrid Method for Adaptive Finite Elements with Application to 3D Flow Problems*. Ph.D. thesis, Universität Heidelberg, Heidelberg, 2005.

[94] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, second edition, 2003.

[95] M. Schäfer and S. Turek. Benchmark computations of laminar flow around a cylinder. In *Flow Simulation with High-Performance Computer II*, edited by E. H. Hirschel, volume 52 of *Notes on Numerical Fluid Mechanics*, pp. 547–566. Vieweg, Braunschweig Wiesbaden, 1996.

[96] M. Schmich and B. Vexler. Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations. *SIAM J. Sci. Comput.*, 30(1), pp. 369–393, 2008.

[97] A. Smolianski. *Numerical Modeling of Two-Fluid Interfacial Flows.* Ph.D. thesis, University of Jyväskylä, Jyväskylä, 2001.

[98] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.*, 114(1), pp. 146–159, 1994.

[99] R. Temam. *Navier-Stokes Equations: Theory and Numerical Analysis.* AMS Chelsea Publishing, Providence, Rhode Island, 2001.

[100] V. Thomée. *Galerkin Finite Element Methods for Parabolic Problems*, volume 25 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin Heidelberg, 1997.

[101] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. *J. Comput. Appl. Math.*, 50(1–3), pp. 67–83, 1994.

[102] R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques.* Wiley-Teubner Series Advances in Numerical Mathematics. Wiley-Teubner, New York Stuttgart, 1996.

[103] VisIt. Visualization Tool. URL `http://www.llnl.gov/visit/`.

[104] A. Walther and A. Griewank. Advantages of binomial checkpointing for memory-reduced adjoint calculations. In *Numerical Mathematics and Advanced Applications. ENUMATH 2003*, edited by M. Feistauer, V. Dolejší, P. Knobloch, and K. Najzar, pp. 834–843. Springer-Verlag, Heidelberg, 2004.

[105] P. Wesseling. *An Introduction to Multigrid Methods.* Edwards, Philadelphia, 2004.

[106] G. Wittum. On the robustness of ILU smoothing. *SIAM J. Sci. Stat. Comput.*, 10(4), pp. 699–717, 1989.

[107] G. Wittum. *Filternde Zerlegungen. Schnelle Löser für große Gleichungssysteme.* Teubner-Skripten zur Numerik. Teubner, Stuttgart, 1992.

[108] J. Wloka. *Partielle Differentialgleichungen.* B. G. Teubner, Stuttgart, 1982.

[109] H.-K. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127(1), pp. 179–195, 1996.

[110] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *Internat. J. Numer. Methods Engrg.*, 33(7), pp. 1331–1364, 1992.

[111] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity. *Internat. J. Numer. Methods Engrg.*, 33(7), pp. 1365–1382, 1992.