Adaptive Discontinuous Galerkin Methods for Variational Inequalities with Applications to Phase Field Models

Dissertation

zur Erlangung des Grades eines Doktors der Naturwissenschaften
am Fachbereich Mathematik und Informatik der Freien Universität Berlin

vorgelegt von

Jes Lasse Hinrichsen-Bischoff

Berlin 2022

Betreuer:	Prof. Dr. Carsten Gräser	(Freie Universität Berlin)
Erstgutachter:	Prof. Dr. Carsten Gräser	(Freie Universität Berlin)
Zweitgutachter:	Prof. Dr. Oliver Sander	(Technische Universität Dresden)

Tag der Disputation:	06.01.2023

Selbstständigkeitserklärung

Name:	Hinrichsen-Bischoff
Vorname:	Jes Lasse

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.
Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

Datum: ________________ Unterschrift: ____________________________

(Jes Lasse Hinrichsen-Bischoff)

Contents

1. Introduction

Many phenomena in scientific applications are modeled with partial differential equations. In the modern, functional-analytic approach, a weak formulation is used to find solutions in a variational setting [54]. Many (stationary) problems can be modeled by finding a function u in some suitable function space H which solves the variational problem

$$a(u, v) = \ell(v) \qquad \forall v \in H. \tag{1.1}$$

That is, for a given bilinear form $a(\cdot, \cdot)$, we try to find a $u \in H$ such that $a(u, \cdot) \in H'$ resolves to the same linear functional as the data $\ell \in H'$. For suitable a, the existence and uniqueness of a solution in a Hilbert space H is guaranteed through the Lax–Milgram theorem [54].

In this thesis, we look at the more general framework of variational *in*equalities. Variational inequalities are used to model a wide range of physical phenomena, such as for example porous media flow [13], contact problems [73] and many more, see also [91] and the references therein. The mathematical formulation is similar to the linear case (1.1). Again, we search for a function $u \in H$, which solves the following inequality:

$$a(u, v - u) + j(v) - j(u) \geq \ell(v - u) \qquad \forall v \in H. \tag{1.2}$$

Here, j is a suitable functional depending on the particular model. Choosing $j \equiv 0$ gives us again the linear problem (1.1), hence generalizing the variational equality case. Under suitable assumptions on a and j, we have again existence and uniqueness of solutions in a Hilbert space H via the Lions–Stampacchia theorem [82](cf. Theorem 2.13). Choosing $j = \chi_K$ as the indicator functional for a convex set $K \subset H$ gives us the important class of (so called) variational inequalities of the first kind. In particular, if K is chosen such that it contains all functions from H which do not violate given obstacle functions pointwise, we speak of an obstacle problem. These obstacle problems can be used, e. g., to model the extension of a membrane under force constrained by a physical obstacle. In a broader spectrum, obstacle problems also arise as subproblems in (semi-)discretized problems for other models. If, for instance, one considers a phase field modeled through the Allen–Cahn equation with an obstacle potential, the stationary problems arising after suitable time discretization are elliptic obstacle problems. Since phase field models are used in a great number of applications, e. g. in material science (see e. g. [105] for an overview), mechanics (e. g. for the simulation of brittle fracture, see [3] for an overview), image segmentation [87] and many more, the efficient numerical treatment of these is of practical importance. However, many of the variational inequalities arising admit a central problem: Their solution might be of very limited regularity even for smooth data [74]. This complicates the numerical analysis and makes efficient numerical treatment hard. It is not obvious how to discretize the problem such that higher order convergence is obtained. Note that, e. g. a discretization of the obstacle problem by piecewise quadratic finite element functions in $\mathcal{P}^2$ only leads to a convergence order of $\mathcal{O}(h^{1.5-\varepsilon})$ [38] instead of $\mathcal{O}(h^2)$ as one might expect from the linear case. Hence, we cannot expect better numerical efficiency solely by using higher order finite element functions. We can however use approaches which try to resolve the different local behaviors of the solution adequately. To this end, we can try to refine the finite element space accordingly using an adaptive approach. An important observation is that for many solutions to variational inequalities, there are indeed (typically large) regions where the solution is locally much smoother than the global regularity implies. For example, the parts which reduce the regularity of the solution to an obstacle problem are usually at the (lower dimensional) free boundary, where the PDE-governed and the obstacle constrained parts of the solution meet. The central idea of this thesis is to apply an *hp*-adaptive approach which tries to exploit the smoother parts with higher order functions (the "p" part of *hp*-adaptivity highlights the order parameter p), while the less regular parts have to be resolved with a finer grid width (denoted with "h"). Since *hp*-adaptivity is very complicated to achieve in a standard (that is, continuous) finite element setting, we opt for the discontinuous Galerkin method. There, functions are only defined on their respective grid element and no inter-element continuity is required. The latter can be enforced, e. g., through a penalty approach [11, 113] which penalizes jumps in the function and therefore generates solutions which are almost continuous. For an overview of methods to apply discontinuous Galerkin (DG) methods to linear problems, we refer to [12]. For variational inequalities, some research has been conducted before. A priori estimates for DG discretizations with piecewise linear and piecewise quadratic functions were derived in [47, 110, 111]. None of these, however, exploit the smoother parts with *hp*-methods. In this thesis, one of the central results is that we prove convergence and error estimates for DG discretizations (using a particular discretization of the admissible set $\mathcal{K}$) of general ansatz orders for the obstacle problem. While in general, the convergence rate of a naive discretization cannot exceed the $\mathcal{O}(h^{1.5})$ bound, we show that for a carefully chosen discretization which acknowledges the free boundary and in particular the nonsmooth regions, one can indeed obtain higher order convergence rates. These are numerically verified in the numerical experiments at the end of the thesis. Finally, we explain how to use a similar approach for (the more general) variational inequalities of the second kind (see equation (1.2)). Proving convergence and error estimates for these can be subject of future research.

The higher order a priori proof relies on a suitable discretization which is locally fine enough with respect to the (unknown) solution. Since in a computation, we do not have the required information beforehand, we use a posteriori error estimates to set up an adaptive algorithm which should identify the aforementioned smooth and nonsmooth regions reliably. Due to their versatility and inherent simplicity, we chose hierarchical error estimates. For (continuous) finite element discretizations, these have been considered in various publications before, see e. g. [46, 75, 76, 77]. In [15], a hierarchical error estimator has been proposed for the DG discretization of the obstacle problem. There, however, the full variational inequality is solved in a bigger space, which may be too expensive for large spaces. Building on the experiences from the aforementioned papers, we propose a preconditioned incremental problem which is computationally cheaper due to localization. For a linear model problem, we prove that the solution of the preconditioned problem is equivalent to the hierarchical error estimate.

Besides providing a suitable discretization (and an adaptive algorithm to obtain it), we have to clarify how to solve the arising algebraic problems. For linear problems, a geometric multigrid approach (see, e. g. [68]) is known to perform well for continuous finite element discretizations (given that a suitable grid hierarchy is available). Similarly, multigrid methods for DG discretizations have been considered [34, 59]. For the discrete variational inequalities we are facing here, however, other methods have to be used to solve the arising nonlinear problems. Many methods have been proposed to solve these such that the solver converges (cf. Chapter 4 and the references there), some of these are of multigrid type (see [64]). One particularly promising approach is the *Truncated Nonsmooth Newton Multigrid* (TNNMG)[61, 64, 67] method, which amends nonlinear smoothers with extra corrections obtained from Newton steps in suitable subspaces. Since it is built on the notion of convexity rather than differentiability [61], it can solve nonsmooth problems in a robust way. In our DG setting, it proved to be a valuable tool to solve the algebraic systems, as well. Since the convergence proof is built around the fact that the nonlinear smoother reduces energy, we are required to use such a smoother, as e. g. a nonlinear Gauss–Seidel method. In a parallel setting, however, it is not obvious how to construct such a smoother. We therefore developed a new parallel nonlinear smoother, which is a nonlinear variant of the so-called ℓ_1-smoother [14], and prove its energy reduction property (and consequently its fitness as a nonlinear smoother in the TNNMG algorithm).

In summary, the thesis explains the full process of solving variational inequalities using adaptive discontinuous Galerkin methods, including the discretization of the problem, the (parallel) solution of the algebraic systems and finally adaptively modifying the discretization to suit the particular problem. Besides the description of the algorithms and methods and practical implementation, we contribute new theoretical results in all of these areas which support the use of the methods.

The individual chapters are outlined as follows: In Chapter 2, we introduce some notation and explain the problems we are dealing with. In particular, we will introduce variational inequalities in more detail and describe Allen–Cahn phase field models. After we know how the continuous problems look like, we will investigate the discretization in Chapter 3. For the time-dependent phase field models, we will briefly discuss time discretizations. For the stationary problems, we show the basic ideas of discontinuous Galerkin finite element spaces and explain a particular method of applying them, namely the Symmetric Interior Penalty (SIPG) method [11]. Afterwards, we will propose a discretization of the obstacle problem and show some a priori estimates as explained before. Finally, we will propose the extension of the ideas from the obstacle problem to more general variational inequalities. Once the discretization is chosen, we need to solve the arising algebraic problems, which is described in Chapter 4. There, the TNNMG method is presented along with a multigrid algorithm for the linear systems which arise as an intermediate step of the TNNMG algorithm. Afterwards, we show how a nonlinear smoother can be used even in a parallel setting. Chapter 5 deals with the question of finding a *good* discretization by applying an *hp*-adaptive algorithm. Most of the chapter is dedicated to finding a suitable error estimator. Once we have such, we explain how elements can be chosen for marking. Since in an *hp* setting it has to be decided which kind of refinement should happen on a marked element (grid refinement (h) or higher polynomial order (p)), we explain some heuristics at the end of the chapter. Before we test everything with numerical experiments, we briefly explain some characteristics and features of our implementation in Chapter 6. In particular, we show how we exploit the blocked structure of the finite element basis functions in a DG space. Finally, in Chapter 7, we test the algorithms and methods proposed in the preceding chapters in several numerical experiments. After verifying our theoretical findings on a set of obstacle problems and Allen–Cahn phase field models, we also apply the algorithms to an image segmentation problem using the Ambrosio–Tortorelli functional near the end of this chapter.

1.1. Acknowledgments

I would like to take the opportunity to thank the many people who helped preparing this thesis in one way or another. Most prominently, there is Prof. Dr. Carsten Gräser who supervised this thesis. He had been a great mentor in the last years, shining with deep knowledge in many mathematical and technical areas and the ability to clearly communicate his ideas with me. His endless patience and curiosity in my work has helped me tremendously and is also valued by many who have had the pleasure of working with him. Moreover, I want to thank Prof. Dr. Ralf Kornhuber for giving me the opportunity to work with him. Over the years, he has been a great teacher whose insights and experience about both mathematics and life in general was invaluable to me. The other members of Ralf's and Carsten's working groups have been good colleagues and even friends for me. This thesis would not have been possible without the constant support through family and friends. These are the people that keep me going. Last, but not least, I want to thank my wife Sandra Bischoff for going the whole way from my undergraduate studies to this thesis with me. I would not have been able to achieve all this without her.

In loving memory of Hannah Hinrichsen.

2. Variational Inequalities and Phase Field Models

2.1. Mathematical Preliminaries and Notation

In the following, we will very briefly introduce some notation and assumptions that should guide the reader through the following chapters. Most of these are well-known and established but are stated to avoid ambiguities. For more in-depth explanations we refer to standard texts on functional analysis and partial differential equations, such as e. g. [1, 54, 115].

2.1.1. Domain

For the remainder of this thesis, we assume the domain Ω to be an open, connected subset of $\mathbb{R}^N$ where the spacial dimension N (usually $N = 2$ or $N = 3$) has to be understood in the local context, if relevant. To ease the further analysis, we assume that Ω satisfies a *uniform cone condition*, i. e.

Assumption 2.1. *We assume that there is a finite cone C such that the following condition holds: For every $x \in \overline{\Omega}$ there is a neighborhood U_x and a cone C_x with vertex x such that C_x is congruent to C, and it holds*

$$z \in \overline{\Omega} \cap U_x \implies z + C_x \subset \Omega,$$

see also [115] or [1] for more detailed expositions.

2.1.2. Function Spaces

In the following, we roughly follow [115] and [1].

Let $\Omega \subset \mathbb{R}^N$ be a domain with the assumptions introduced in the preceding paragraph.

For a multiindex $s \in \mathbb{N}^N$, we write $|s| = s_1 + \cdots + s_N$ and

$$D^s = \frac{\partial^{|s|}}{\partial x_1^{s_1} \ldots \partial x_N^{s_N}}.$$

We call the space of real-valued functions $f : \Omega \to \mathbb{R}$ with bounded and continuous derivatives $D^s f$, $|s| \leq k$ (up to k-th order) $C^k(\Omega)$ and equip it with the norm

$$\|f\|_{C^k(\Omega)} = \sup_{|s| \leq k, x \in \Omega} |D^s f(x)|.$$

Accordingly, for the set of functions that are continuous up to the boundary of the domain, we reserve the symbol $C^k(\overline{\Omega})$ and the norm

$$\|f\|_{C^k(\overline{\Omega})} = \max_{|s|\leq k, x\in\Omega} |D^s f(x)|, \quad f \in C^k(\overline{\Omega}).$$

The subspace of $C^k(\overline{\Omega})$ which has Hölder continuous derivatives with constant λ up to degree k will be denoted by $C^{k,\lambda}(\overline{\Omega})$ and can be normed via

$$\|f\|_{C^{k,\lambda}(\overline{\Omega})} = \|f\|_{C^k(\overline{\Omega})} + \max_{|s|\leq k} \sup_{x\neq y} \frac{|D^s f(x) - D^s f(y)|}{|x-y|^\lambda},$$

cf. [1].

Let $\mathcal{L}^N$ be the N-dimensional Lebesgue measure on $\mathbb{R}^N$. For $1 \leq p < \infty$, we denote the space of Lebesgue-measurable functions f such that

$$\int_\Omega |f(x)|^p \,\mathrm{d}\mathcal{L}^N(x) < \infty$$

by $L^p(\Omega)$ and equip it with the usual norm

$$\|f\|_{L^p(\Omega)} = \left(\int |f(x)|^p \,\mathrm{d}\mathcal{L}^N(x)\right)^{1/p}.$$

For convenience, we will usually write "dx" instead of "d$\mathcal{L}^N$". The space $L^2(\Omega)$ is a separable Hilbert space [115] with the following scalar product:

$$(v,w)_{L^2\Omega} = \int_\Omega vw \,\mathrm{d}x, \quad v,w \in L^2(\Omega).$$

Consequently, we have

$$\|v\|_{L^2(\Omega)} = \sqrt{(v,v)_{L^2(\Omega)}}.$$

We may frequently omit the subscript and write $(v,w) = (v,w)_{L^2(\Omega)}$. Sometimes, when a particular domain U is to be highlighted, we might write

$$(v,w)_U = \int_U vw \,\mathrm{d}x.$$

When integrating over a $(N-1)$ manifold embedded in $\mathbb{R}^N$ (usually the boundary of an N-dimensional open set), we will denote the surface measure (i. e. the $(N-1)$-dimensional Hausdorff measure $\mathrm{d}\mathcal{H}^{N-1}$) by d$S$.

For $p = \infty$, we say that $L^\infty(\Omega)$ is the set of real-valued functions that are bounded up to sets of measure zero and define the norm

$$\|f\|_{L^\infty(\Omega)} = \operatorname{ess\,sup}(f) = \sup_{\substack{x\in\Omega\setminus A,\\ \mathcal{L}^N(A)=0}} |f(x)|.$$

Finally, we define the set of locally integrable functions by $L^1_{\text{loc}}(\Omega)$, i. e. those functions f defined a. e. on Ω such that $f \in L^1(K)$ for every open K such that $K \subset\subset \Omega$ (i. e. K's closure is compact in Ω).

To define weak derivatives, we first introduce the space of *test functions*,

$$\mathcal{D}(\Omega) = C_0^\infty(\Omega),$$

i. e. the set of infinitely often continuously differentiable functions whose support is compact in Ω. Note that $\mathcal{D}(\Omega)$ is not a normable space but can be equipped with a suitable locally convex topology [1].

The dual space $\mathcal{D}'(\Omega)$ (equipped with the weak-star topology) is called the space of *distributions.*

Let $u \in L^1_{\text{loc}}(\Omega)$. We say u has a weak (or distributional) derivative (corresponding to a multiindex s) if there is a $v_s \in L^1_{\text{loc}}(\Omega)$ such that

$$\int_\Omega u D^s \phi \, \mathrm{d}x = (-1)^{|s|} \int_\Omega v_s \phi \, \mathrm{d}x \quad \forall \phi \in \mathcal{D}(\Omega).$$

If u has classical derivatives, these coincide with the weak derivatives and we have $v_s = D^s u$ [1]. Motivated by this fact, we will abuse notation and describe the weak derivative also by $D^s u$ if it is clear from the context.

Now we can finally define Sobolev spaces. Roughly speaking, these are the functions which have weak derivatives in $L^p(\Omega)$. More precisely, for $m \in \mathbb{N}_0$ and $1 \leq p \leq \infty$,

$$W^{m,p}(\Omega) = \left\{ v \in L^p(\Omega) : D^s v \in L^p(\Omega), |s| \leq m \right\}.$$

This space is equipped with the norm

$$\|v\|_{W^{m,p}(\Omega)} = \left(\sum_{|s| \leq m} \|D^s v\|^p_{L^p\Omega} \right)^{1/p}.$$

The special case $p = 2$ deserves more attention. We write

$$W^{m,2}(\Omega) = H^m(\Omega).$$

This is consistent with the usual definition of $H^m(\Omega)$ spaces due to the uniform cone condition we assumed in the previous section [115]. The space $H^m(\Omega)$ can be equipped with the scalar product

$$(v, w)_{H^k(\Omega)} = \sum_{|s| \leq m} (D^s v, D^s w), \quad v, w \in H^m(\Omega), \tag{2.1}$$

using again the L^2 scalar product. Clearly, we have $H^0(\Omega) = L^2(\Omega)$.

As notational convenience, we will frequently write

$$\|v\|_m = \|v\|_{H^m(\Omega)}, \quad v \in H^m(\Omega).$$

The L^2-norm of the weak derivative $D^s v$ is only a half norm on $H^m(\Omega)$. Conventionally, we will write

$$|v|_s = \sqrt{(D^s v, D^s v)}. \tag{2.2}$$

If both the order m of the Sobolev norm and the domain U is to be highlighted, we may use two subscripts and write

$$\|v\|_{m,U} := \|v\|_{H^m(U)}.$$

The analog notation shall hold for half norms (2.2).

For fractional order Sobolev spaces $W^{s,p}$ with non-integer s, there are several ways to define them which prove to be largely equivalent [1] in many cases. Following [89], we state one definition using Gagliardo seminorms.

Definition 2.2. *Let $s \in (0,1)$. The fractional Sobolev space $W^{s,p}(\Omega)$ $(1 \le p \le \infty)$ is defined by*

$$W^{s,p}(\Omega) = \left\{ v \in L^p(\Omega) : \frac{|u(x) - u(y)|}{|x-y|^{n/p+s}} \in L^p(\Omega \times \Omega) \right\}.$$

This space can be equipped with the norm

$$\|v\|^p_{W^{s,p}(\Omega)} = \|v\|^p_{L^p(\Omega)} + \int_\Omega \int_\Omega \frac{|u(x) - u(y)|^p}{|x-y|^{n+sp}} \, \mathrm{d}x \, \mathrm{d}y.$$

This space is in a sense an intermediate space between L^p and $W^{1,p}$ [89]. For non-integer s greater than one, we split s into an integer part m and a non-integer part $\sigma \in (0,1)$,

$$s = \underbrace{\lfloor s \rfloor}_{=:m} + \underbrace{(s - \lfloor s \rfloor)}_{=:\sigma}.$$

Using this splitting, we have

$$W^{s,p}(\Omega) = \left\{ v \in W^{m,p}(\Omega) : D^\alpha v \in W^{\sigma,p}(\Omega), \text{for any } \alpha \text{ s.t. } |\alpha| = m \right\}.$$

Consequently, this space can be normed with

$$\|v\|^p_{W^{s,p}(\Omega)} = \|v\|^p_{W^{m,p}(\Omega)} + \int_\Omega \int_\Omega \frac{|u(x) - u(y)|^p}{|x-y|^{n+\sigma p}} \, \mathrm{d}x \, \mathrm{d}y.$$

Remark 2.3. One might add a factor $\sigma(1-\sigma)$ in front of the of the double integral above to obtain the usual norm if $\sigma \nearrow 1$ or $\sigma \searrow 0$ [89].

2.2. Variational Inequalities of the First Kind

For this chapter, we consider a Hilbert space H. Let $(\cdot,\cdot) : H \times H \to \mathbb{R}$ and $\langle \cdot,\cdot \rangle : H' \times H \to \mathbb{R}$ denote H's inner product and dual pairing, respectively.

To state the problem, consider a continuous bilinear form $a(\cdot,\cdot)$ on H. We assume that $a(\cdot,\cdot)$ is coercive, i. e. there is a $\alpha > 0$ such that

$$\alpha\|x\|^2 \leq a(x,x) \quad \forall x \in H.$$

Moreover, let $l \in H'$.

Problem 2.4. *Let $\mathcal{K} \subseteq H$ be a closed, convex set (not necessarily bounded). Find $u \in \mathcal{K}$ such that*

$$a(u, v-u) \geq l(v-u) \quad \forall v \in \mathcal{K}. \tag{2.3}$$

Equation (2.3) is called a *variational inequality of the first kind.*

Remark 2.5. For $\mathcal{K} = H$, Problem 2.4 reduces to the linear problem of finding $u \in H$ such that

$$a(u,v) = l(v) \quad \forall v \in H.$$

By the Lax–Milgram theorem, this problem possesses a unique solution.

Theorem 2.6. *Problem 2.4 possesses a unique solution $u \in H$.*

Moreover, the problem is well-posed in the sense that the dependence on l is Lipschitz: Let $l_1, l_2 \in H'$ and $u_1, u_2 \in H$ the respective solutions to (2.3). *Then, it holds*

$$\|u_1 - u_2\|_H \leq 1/\alpha \|l_1 - l_2\|_{H'}.$$

Proof. See the Lions–Stampacchia theorem [82]. □

To illustrate Problem 2.4, we consider the following application.

2.2.1. The Obstacle Problem

The elliptic obstacle problem will serve as an example for variational inequalities of the first kind. It is both relevant on its own (e. g. modeling the extension of a membrane perturbed by some obstacle) and as part of the numerical solution process of a time-discretized parabolic problem such as e. g. the Allen–Cahn equation, see sections 2.4 and 3.1.

For an open, connected domain $\Omega \subset \mathbb{R}^d$, consider an appropriate subset of $L^2(\Omega)$, e. g. $H = H^1(\Omega)$ (possibly restricted to functions that satisfy given boundary conditions). We choose lower and upper obstacle functions $\underline{\psi}, \overline{\psi} \in H$ with $\underline{\psi} \leq \overline{\psi}$ almost everywhere. Then, the set

$$\mathcal{K} = \left\{ v \in H : \underline{\psi} \leq v \leq \overline{\psi} \text{ a.e.} \right\} \tag{2.4}$$

is convex and closed.

Remark 2.7. We are free to drop one of the obstacles, such that we consider only a single obstacle ψ which can be a lower or upper obstacle. Informally, this can also be seen by setting $\underline{\psi} \equiv -\infty$ or $\overline{\psi} \equiv \infty$.

Example 2.7.1. A prototypical example would be now to consider the Poisson equation

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

In its weak form, this reads

$$u \in H_0^1(\Omega)\colon\ (\nabla u, \nabla v) = \langle f, v\rangle \quad \forall v \in H_0^1(\Omega). \tag{2.5}$$

More generally, we will consider second order elliptic equations of the form

$$Lu = -\operatorname{div}(A\nabla u) + \alpha u = f, \tag{2.6}$$

where $A(x) \in \mathbb{R}^{d\times d}$ is symmetric positive definite for all x, there is a $\theta > 0$ such that $\lambda_{\min}(A(x)) \geq \theta$ for all x and finally $\alpha \geq 0$. For simplicity, we also assume $A_{ij} \in C^1(\overline{\Omega})$, though $A_{ij} \in L^\infty(\Omega)$ might suffice in many cases [54].

This gives rise to a bilinear form

$$a(v, w) = (A\nabla v, \nabla w) + \alpha\,(v, w)\,.$$

and a functional

$$l(v) = \langle f, v\rangle.$$

Coercivity and boundedness of $a(\cdot,\cdot)$ are standard results to be found in any textbook on partial differential equations, see e. g. [54].

Theorem 2.6 now gives us that there is a unique $u \in \mathcal{K}$ which solves

Problem 2.8 (Obstacle Problem). *Find* $u \in \mathcal{K}$ *such that*

$$a(u, v-u) \geq \langle f, v-u\rangle \quad \forall v \in \mathcal{K}. \tag{2.7}$$

Over the years several, several regularity results have been established, see e. g. [74, 96]. For example, we have for the unilateral lower obstacle case:

Theorem 2.9. *Assume* Ω *has a smooth boundary. Let* $a(\cdot,\cdot) = (\nabla\cdot, \nabla\cdot)$ *and* $\overline{\psi} \equiv \infty$ *(i. e. no upper obstacle). For* $1 < p < \infty$*, assume* $f \in L^p(\Omega)$ *and* $\max(-\Delta\underline{\psi} - f, 0) \in L^p(\Omega)$*. Then, the solution* u *of the obstacle problem* (2.7) *has the property*

$$u \in W^{2,p}(\Omega) \cap C^{1,\lambda}(\overline{\Omega})$$

with $\lambda = 1 - N/p$*.*

Proof. See Theorem 2.3 in [74]. □

However, simple counter examples exists which show that in general $u \notin C^2(\Omega)$ (and also $u \notin H^3(\Omega)$) even for very smooth data [96]. Thus, even this very simple example of a variational inequality involves some limitations for both the analytical and the numerical treatments. Another result by Brezis [37] states that under more severe assumptions on the problem data and the domain, one has

$$u \in W^{s,p}(\Omega), \quad 1 < p < \infty, \quad s < 2 + 1/p, \tag{2.8}$$

see [37, 112].

For our results about a priori error estimates of our discretization schemes, we will not discuss the requirements on the problem data in detail but rather assume a certain smoothness of the solution. We will, however, respect the fact that the obstacle problem's solution is in general not too smooth and in particular we will not assume any smoothness that goes beyond (2.8).

2.3. Variational Inequalities of the Second Kind

Picking up the notation from Section 2.2, we will introduce variational inequalities of the second kind. Again, let $a(\cdot,\cdot) : H \times H \to \mathbb{R}$ be a continuous, coercive bilinear form and $l \in H'$ be a continuous functional.

Definition 2.10. *A functional $j : X \to \mathbb{R}$ is said to be* lower semicontinuous *if for any sequence $\{x_k\}_k \subset X$ with $x_k \to x^* \in X$, we have*

$$\liminf j(x_k) \geq j(x^*).$$

Moreover, we say j is proper *if $j(x) > -\infty$ for all $x \in X$ and $\exists \tilde{x} \in X$ such that $j(\tilde{x}) < \infty$.*

Problem 2.11. *Assume $j : H \to \mathbb{R}$ is a convex, lower semicontinuous and proper functional. Find $u \in H$ such that*

$$a(u, v-u) + j(v) - j(u) \geq l(v-u) \quad \forall v \in H. \tag{2.9}$$

The inequality (2.9) *is called a* variational inequality of the second kind.

Remark 2.12. Variational inequalities of the first kind can be viewed as a special case of variational inequalities of the second kind by choosing the indicator functional $j = \chi_{\mathcal{K}}$, where

$$\chi_{\mathcal{K}}(v) = \begin{cases} 0 & \text{if } v \in \mathcal{K}, \\ \infty & \text{else.} \end{cases}$$

Analogously to the case for variational inequalities of the first kind, we can state existence and uniqueness of solutions:

Theorem 2.13. *Problem 2.11 possesses a unique solution $u \in H$.*

Proof. See again Lions and Stampacchia [82]. □

2.4. Allen–Cahn Phase Field Models

In the following, we will consider a simple phase field model introduced by Allen and Cahn [2]. It describes the separation of phases driven by a gradient flow with respect to the space $L^2(\Omega)$ of (a scaled version of) the so-called Ginzburg–Landau energy [56],

$$\mathcal{E}(u) = \int_\Omega \frac{\varepsilon}{2}|\nabla u|^2 + \frac{1}{\varepsilon}F(u)\,\mathrm{d}x, \tag{2.10}$$

with F being a (possibly nonsmooth) double-well potential to be defined later.

We will consider a binary phase field, where a solution u assumes values between -1 (Phase A) and 1 (Phase B). The regions where u is neither -1 nor 1, often called *interface*, is usually of width $\mathcal{O}(\varepsilon)$ and its evolution is governed by the mean curvature flow of the zero level set of a given initial function u_0 [43]. Indeed, it can be shown that the zero level set of the Allen–Cahn solution u converges to the mean curvature flow for $\varepsilon \to 0$ for an appropriate potential F [43, 98].

Mathematically, the Allen–Cahn equation is closely related to the heat equation, perturbed by a nonlinear potential driving the function values to one of the stable phases A or B.

In the last decades, the problem has been extensively discussed both from an analytic and a numerical point of view. For the problem statement, we will follow the notation of [56]. The *Allen–Cahn* equation reads as follows:

Problem 2.14. *Let $\Omega \subset \mathbb{R}^d$ be a bounded, convex domain with polygonal boundary and let $T > 0$ be a finite time horizon. Find $u \in L^2([0,T], H^1(\Omega))$, $u_t \in L^2([0,T], H^1(\Omega))$ such that*

$$\varepsilon u_t - \varepsilon \Delta u + \frac{1}{\varepsilon} f(u) = 0 \qquad \textit{in } \Omega \times (0,T), \tag{2.11}$$

$$u = u_0 \qquad \textit{in } \Omega \times \{0\}, \tag{2.12}$$

$$\frac{\partial u}{\partial n} = 0 \qquad \textit{on } \partial\Omega \times (0,T), \tag{2.13}$$

for some initial state $u_0 \in \mathcal{G} = \left\{ v \in L^\infty(\Omega) : |v| \leq 1 \right\}$.

We assume F to be a double well potential with global minima at ± 1 and $\varepsilon > 0$ is a parameter corresponding to the width of the interface between the two phases.

Some often used potentials are

- the quartic potential $F(\xi) = \frac{1}{4}\left(\xi^2 - 1\right)^2$,
- the logarithmic potential
$$F(\xi) = \chi_{[-1,1]}(\xi) + \frac{\theta}{2}\left[(1+\xi)\ln(1+\xi) + (1-\xi)\ln(1-\xi)\right] - \frac{\theta_c}{2}\xi^2$$
with $\theta < \theta_c$,
- the obstacle potential $F(\xi) = \frac{1}{2}(1 - \xi^2) + \chi_{[-1,1]}(\xi)$.

All of these potentials have convex and concave parts (creating the double well structure of the potential). For the further analysis, it is convenient to rewrite these parts explicitly:

$$F(\xi) = \Phi(\xi) - \frac{\alpha}{2}\xi^2. \tag{2.14}$$

Here, $-\frac{\alpha}{2}\xi^2$ (with α being 1 for the quartic and obstacle potential and θ_c for the logarithmic potential) is the concave smooth part of the function while Φ is the convex, potentially nonsmooth part. To simplify notation, we will assume $\alpha = 1$ in the following. For smooth Φ, the function f in (2.11) is the derivative $f(\xi) = F'(\xi) = \Phi'(\xi) - \xi$. If the convex part Φ is nonsmooth, we have to consider the subdifferential of Φ.

Definition 2.15. *Let X be a normed vector space. For $g : X \to \mathbb{R}$ convex, we say $x' \in X'$ is a* subgradient *of g at x if*

$$g(z) \geq g(x) + \langle x', z - x\rangle \quad \forall z \in X.$$

The subdifferential $\partial g : X \to 2^{X'}$ *is defined through*

$$x \mapsto \left\{x' \in X' : x' \textit{ is subgradient of } g \textit{ at } x\right\}.$$

Clearly, if g is differentiable at x, we have $\partial g(x) = \{g'(x)\}$.

For the Allen–Cahn equation, this means if the convex part Φ might be nonsmooth, we say $f(u) = \partial\Phi(u) - u$. Since the subdifferential is set-valued, this renders (2.11) an inclusion of the form

$$\varepsilon u_t - \varepsilon\Delta u + \frac{1}{\varepsilon} f(u) \ni 0 \quad \text{in } \Omega \times (0, T).$$

Using the convex-concave splitting, we can also rewrite the Ginzburg–Landau energy accordingly:

$$\mathcal{E}(u) = \varepsilon\mathcal{E}_0(u) + \frac{1}{\varepsilon}\phi(u) - \frac{1}{\varepsilon}\|u\|^2,$$

with

$$\mathcal{E}_0(u) = \int_\Omega |\nabla u|^2 \,\mathrm{d}x, \quad \phi(u) = \int_\Omega \Phi(u)\,\mathrm{d}x.$$

It can be shown that (2.11) can be rewritten as a parabolic variational inequality of the second kind, see e. g. [57]. Thus, we replace Problem 2.14 by the following parabolic variational inequality of the second kind:

Problem 2.16. *Let $H = W^1([0, T], H^1(\Omega))$. Find $u \in H$ such that*

$$\varepsilon\,(u_t, v - u) + \varepsilon\,\big(\nabla u, \nabla(v - u)\big) - \frac{1}{\varepsilon}(u, v - u) + \frac{1}{\varepsilon}(\phi(v) - \phi(u)) \geq 0 \quad \forall v \in H. \tag{2.15}$$

For the special case of the obstacle potential, i. e. Φ being $\chi_{[-1,1]}$, we have that Problem 2.16 can be translated into a variational inequality of the first kind [74], namely

Problem 2.17. *Let $H = W^1([0, T], H^1(\Omega))$. Find $u \in H$ such that*

$$\varepsilon\,(u_t, v - u) + \varepsilon\,\big(\nabla u, \nabla(v - u)\big) - \frac{1}{\varepsilon}(u, v - u) \geq 0 \quad \forall v \in \mathcal{K}, \tag{2.16}$$

where $\mathcal{K} = \{v \in H : |v(\cdot, t)| \leq 1$ a. e. in Ω and for all $t\}$.

We cite the following result about existence and uniqueness of the Allen–Cahn equation.

Theorem 2.18. *For $u_0 \in \big\{v \in L^\infty(\Omega) : |v| \leq 1 \text{ a. e.}\big\}$, Problem 2.16 (and consequently Problem 2.17 as a special case) has a unique solution.*

Proof. See [36] for the theoretical foundation and [100] for an extended discussion.

The special case of the obstacle potential was also discussed in [41]. □

Having the Allen–Cahn equation written as parabolic variational inequality will allow us to treat it in the same framework as in the stationary case once an appropriate time discretization has been chosen. Indeed, Problem 2.16 and Problem 2.17 have the same structure as problems 2.11 and 2.4, respectively. Thus many of the methods for elliptic problems developed in this thesis carry over and can be applied to phase-field evolutions.

3. Discretization

We are interested in having approximate solutions to the problems introduced in Chapter 2. The natural question which arises is how to turn the continuous (and therefore generally infinite dimensional) problems into problems of finite dimension which can be handled by a numerical algorithm. This questions has to be answered both for the time discretization (if the given problem is not stationary) and for the space discretization. Both questions shall be answered in this chapter. Since these are very broad questions for which (depending on the problem) many answers may exist, we focus on a small set of methods which we will discuss extensively in this thesis. We start out by suggesting discretizations in time for the Allen–Cahn equation and continue by introducing discontinuous Galerkin methods with a special focus on the Symmetric Interior Penalty Galerkin (SIPG) method. The latter will be first explained for a linear elliptic model problem and later be applied to variational inequalities.

For the specific case of the obstacle problem, we will derive some new a priori estimates in Section 3.3 which generalize existing results such as [111]. These are particularly interesting because they indicate a way how higher order convergence rates can be obtained despite the limited regularity of the underlying problem.

3.1. Time Discretization of the Allen–Cahn Equation

For the numerical approximation of the Allen–Cahn equation (or more general, for time-dependent partial differential equations), we are left with two possibilities. First, one could discretize in space (using e. g. a finite element method on a fixed grid) and discretize in time afterwards. This method ("method of lines") has the severe drawback that one has to decide for a spatial discretization once and for all, effectively eliminating the possibility of applying adaptively refined spatial discretizations in each time step. Since the solution of a time-dependent PDE typically evolves with time, different regions of the domain might have to be more accurately resolved. This is of course not possible if the spatial discretization is fixed for all time steps.

Hence, we opt for an alternative way by choosing a time-discretization first. This will lead to elliptic problems in each timestep which subsequently can (maybe adaptively) be discretized in space ("Rothe's method" [97]), see also [27] for an extended discussion.

We recall that the Allen–Cahn equation can be written as a variational inequality of the second kind, see Problem 2.16. Thus, our problem reads (given a suitable initial function u_0)

$$\varepsilon\left(u_t, v-u\right)+\varepsilon\big(\nabla u, \nabla(v-u)\big)-\frac{1}{\varepsilon}(u, v-u)+\frac{1}{\varepsilon}(\phi(v)-\phi(u)) \geq 0 \quad \forall v \in H.$$

In the following, we will briefly discuss two first order time integration schemes for the Allen–Cahn equation, namely fully implicit and semi-implicit Euler methods.

3.1.1. Fully Implicit Euler Scheme

For a given timestep size $\tau = T/M > 0$, $M \in \mathbb{N}$, and the approximation from a previous time step u^m, we can apply the implicit Euler method and get the stationary variational inequality of finding $u^{m+1} \in H$ such that

$$\begin{aligned}\frac{\varepsilon}{\tau}\left(u^{m+1} - u^m, v - u^{m+1}\right) + \varepsilon\left(\nabla u^{m+1}, \nabla(v - u^{m+1})\right)\\ -\frac{1}{\varepsilon}(u^{m+1}, v - u^{m+1}) + \frac{1}{\varepsilon}(\phi(v) - \phi(u^{m+1})) \geq 0 \quad \forall v \in H.\end{aligned}$$

Rearranging terms, we get

$$\begin{aligned}\left(\frac{\varepsilon}{\tau} - \frac{1}{\varepsilon}\right)\left(u^{m+1}, v - u^{m+1}\right) + \varepsilon\left(\nabla u^{m+1}, \nabla(v - u^{m+1})\right)\\ +\frac{1}{\varepsilon}(\phi(v) - \phi(u^{m+1})) \geq \frac{\varepsilon}{\tau}\left(u^m, v - u^{m+1}\right) \quad \forall v \in H.\end{aligned} \tag{3.1}$$

Since existence and uniqueness results for solutions of variational inequalities usually require the bilinear form to be coercive, we deduce that the factor in front of the mass term should not be negative. Therefore, a restriction on the length τ of each timestep, namely

$$\tau < \varepsilon^2$$

naturally emerges. Indeed, we have

Theorem 3.1. *Under the condition $\tau < \varepsilon^2$, the implicit Euler method* (3.1) *is stable, i. e.*

$$\frac{\varepsilon}{2\tau}\sum_{m=1}^{M}\|u^m - u^{m-1}\|_0^2 + \frac{\varepsilon}{2}\sum_{m=1}^{M}\|\nabla u^m - \nabla u^{m-1}\|_0^2 \leq \mathcal{E}(u^0),$$

where $\mathcal{E}$ is the Ginzburg–Landau energy defined in (2.10)*.*

Proof. This is a special case of [65, Theorem 3.3]. The obstacle potential case was also discussed in [26]. □

Consequently, the choice of the implicit Euler method is on the one hand reasonable as it is stable, on the other hand we have a restriction on the time steps that scales quadratically with the (presumably already small) interface width ε.

Remark 3.2. In [56], error estimates for an implicit Euler discretization of the strong form of the Allen–Cahn equation have been derived. Of particular interest is that the authors succeed to circumvent the typical exponential term arising from application of the Grönwall lemma which would erode any practical applicability of the result. Instead, they are only left with a term scaling like $1/\varepsilon$ in low polynomial order.

Under several (rather complex) regularity assumptions, they show $O(\tau)$ error estimates for the time-discretized equation. The convergence rate, however, will decrease for a less regular u_0.

To the best of the author's knowledge, there are no further results known for general potentials. There are, however, some results for the quartic potential (see e. g. [103]). Since we are mostly concerned with the nonsmooth potential case, we will not review these here.

3.1.2. Semi-implicit Euler Scheme

As a remedy, semi-implicit schemes have been introduced (see, e. g. [43]) which discretize the concave part of the double-well potential explicitly while applying the implicit scheme to the rest of the variational inequality.

This reads

$$\begin{aligned}&\frac{\varepsilon}{\tau}\left(u^{m+1}-u^m, v-u^{m+1}\right)+\varepsilon\left(\nabla u^{m+1}, \nabla(v-u^{m+1})\right)\\&-\frac{1}{\varepsilon}(u^m, v-u^{m+1})+\frac{1}{\varepsilon}(\phi(v)-\phi(u^{m+1}))\geq 0 \quad \forall v\in H.\end{aligned}$$

Note that we now have (the *old* state) u^m in the $-\frac{1}{\varepsilon}(\,\cdot\,, v-u^{m+1})$ term. Again, rearranging terms gives

$$\begin{aligned}&\frac{\varepsilon}{\tau}\left(u^{m+1}, v-u^{m+1}\right)+\varepsilon\left(\nabla u^{m+1}, \nabla(v-u^{m+1})\right)\\+&\frac{1}{\varepsilon}(\phi(v)-\phi(u^{m+1}))\geq\left(\frac{\varepsilon}{\tau}+\frac{1}{\varepsilon}\right)\left(u^m, v-u^{m+1}\right) \quad \forall v\in H.\end{aligned} \tag{3.2}$$

In this case, no negative terms in front of the bilinear forms can appear no matter how large τ is. This promising fact manifests itself in the following theorem, see [43] and [65]:

Theorem 3.3. *The semi-implicit Euler method* (3.2) *is unconditionally stable, i. e.*

$$\left(\frac{\varepsilon}{\tau}+\frac{1}{2\varepsilon}\right)\sum_{m=1}^{M}\|u^m-u^{m-1}\|_0^2+\frac{\varepsilon}{2}\sum_{m=1}^{M}\|\nabla u^m-\nabla u^{m-1}\|_0^2\leq\mathcal{E}(u^0),$$

where $\mathcal{E}$ is again the Ginzburg–Landau energy defined in (2.10)*.*

Proof. This is a special case of [65, Theorem 3.5]. □

While this theoretical result looks very promising, it was observed in practical computations that the semi-implicit scheme severely underestimates the speed of the evolution for large timestep sizes. Even for choices of τ that fulfill the restriction for implicit schemes, namely $\tau<\varepsilon^2$, the time discretized problem evolves too slowly [23, 65].

3.2. Spatial Discretization – Discontinuous Galerkin

Discontinuous Galerkin methods are used since the 1970s [12]. While first used for solving hyberbolic problems, they were adopted in the numerical solution of elliptic and parabolic problems soon. The variational formulation of many problems allowed to commit what is called a "variational crime" (see, e. g. [35]) by dropping the continuity requirement of the ansatz functions in the finite element spaces. Each basis function in a discontinuous Galerkin finite element space has only support in a single element of a given grid $\mathcal{T}$. In particular, it is not required that the elements of these spaces are continuous across element boundaries.

Remark 3.4. In this thesis, we will call the $(N-1)$-dimensional intersections of two elements *faces* (as opposed to e. g. edges in 2D) independently of N.

To ensure that numerical solutions to the discretized problems are continuous (up to a certain point) and the bilinear forms arising from discretizing elliptic problems still fulfill requirements such as stability, certain modifications have to be made. Several methods have been proposed which emphasize different properties to varying degree. For an overview, see [12]. There, many of these approaches are put into a joined framework that allows for a unified analysis.

In this thesis, we will use the symmetric variant of Interior Penalty Discontinuous Galerkin (SIPG)[11]. The main idea is that discontinuities across element faces are penalized by adding extra terms in the bilinear form. We emphasize that other DG methods that give symmetric bilinear forms and are stable could be employed as well.

In recent years, discontinuous Galerkin methods got a renewed interest due to technical aspects on high performance hardware. The discontinuous structure of the ansatz spaces offers several possibilities for parallelization and matrix-free computation, which have been shown to be superior to classical matrix based approaches in certain applications [81, 88]. In particular, very large problems can become feasible with these approaches. Moreover, easy handling of non-conforming grids and varying polynomial degrees make discontinuous Galerkin methods particularly suited for *hp*-adaptive computations, see also Chapter 5.

3.2.1. The Discontinuous Finite Element Space

For the rest of this work, let Ω be an open, connected domain in $\mathbb{R}^N$, where N is typically 2 or 3, see also the assumptions in Section 2.1. We assume that Ω has a polygonal boundary.

Definition 3.5. *We say the set $\mathcal{T}$ is a* grid*, if it is a partition of Ω consisting of open disjoint elements K that cover Ω. The elements K in $\mathcal{T}$ are required to be (diffeomorphic) affine images of either the reference cube $(0,1)^N$ or the N-simplex. For every element $K \in \mathcal{T}$, let $F_K\colon\ \colon \hat{K} \to \overline{K}$ denote the diffeomorphism which maps the reference element to $\overline{K}$.*

We explicitly do not require that $\mathcal{T}$ is conforming, i. e. in particular hanging nodes are allowed.

Definition 3.6. *For a given grid $\mathcal{T}$, we denote that set of $(N-1)$-dimensional intersections between grid elements (the* inner faces*) by Γ_h, i. e.*

$$\Gamma_h = \left\{ \overline{K_0} \cap \overline{K_1} : K_0, K_1 \in \mathcal{T}, \dim(\overline{K_0} \cap \overline{K_1}) = N-1 \right\}.$$

The set of boundary faces *is called Γ_b, i. e.*

$$\Gamma_b = \left\{ e = \partial\Omega \cap \partial K : \dim(e) = N-1, K \in \mathcal{T} \right\},$$

Now, the union of these is the set of all faces, defined by

$$\Gamma := \Gamma_b \dot{\cup} \Gamma_h. \tag{3.3}$$

Finally, we define for a given element K

$$\Gamma_K = \{ e \in \Gamma : e \subset \partial K \}$$

the set of faces belonging to K.

To avoid potential corner cases, we introduce the following condition [35]:

Definition 3.7. *A family of partitions is* regular *or* non-degenerate *if there are positive numbers θ and ρ such that for every K, it holds*

1. *Every angle of K is greater or equal to θ,*
2. *for every face $e \in \Gamma$ with $e \subset \partial K$, we have*

$$|\partial K| / |e| \leq \rho.$$

In particular, the second condition implies that the number of faces corresponding to a single grid element is uniformly bounded.

Assumption 3.8. *All grids considered in this thesis are assumed to be* regular*. In particular, if a sequence of grids are considered (say due to refinement), the same constants θ and ρ hold for all grids.*

For a given grid, we can now proceed to define functions defined on grid entities and construct the discontinuous Galerkin finite element spaces.

Definition 3.9. *Let $K \in \mathcal{T}$ be an element generated as by an affine transformation of the reference* cube *$\hat{K}$. $\mathcal{Q}^k(K)$ is the space of tensor-product polynomials of degree k on $K \in \mathcal{T}$, i. e. we have*

$$\mathcal{Q}^k(K) = \operatorname{span} \left\{ v : \overline{K} \to \mathbb{R} : v(x) = \prod_{i=1}^{N} v_i \left((F_K^{-1}(x))_i \right), v_i \in \mathbb{P}^k([0,1]) \right\},$$

where $\mathbb{P}^k([0,1])$ is the set of all polynomials on the unit interval of degree at most k.

For K being the affine image of the N-simplex (e. g. a triangle), we have the set of complete polynomials

$$\mathcal{P}^k(K) = \left\{ p : \overline{K} \to \mathbb{R} : p(x) = \sum_{\substack{i \in \{0,\dots,k\}^N \\ |i| \leq k}} c_i (F_K^{-1}(x))^i, c_i \in \mathbb{R} \right\},$$

where $|i| = \sum_{k=1}^N i_k$ and $y^i = \prod_{k=1}^N y_k^{i_k}$ for $y \in \mathbb{R}^N$.

To ease the notation we introduce the following symbol denoting the type of polynomial space depending on the respective reference element of each grid element:

Definition 3.10. *For each $K \in \mathcal{T}$ and its reference element $\hat{K} = F_K^{-1}(K)$, we define*

$$\mathsf{P}^k(K) = \begin{cases} \mathcal{Q}^k(K) & \text{if } \hat{K} \text{ is a cube,} \\ \mathcal{P}^k(K) & \text{if } \hat{K} \text{ is a simplex.} \end{cases}$$

We can now define the abstract DG space.

Definition 3.11. *For a given function $p \colon \mathcal{T} \to \mathbb{N}$ that assigns every element in $\mathcal{T}$ a polynomial degree, the discontinuous Galerkin space $V_{\mathcal{T}}^p$ consists of all $L^2(\Omega)$ functions that are piecewise polynomials of degree $p(K)$:*

$$V_{\mathcal{T}}^p = \left\{ v \in L^2(\Omega) : v|_K \in \mathsf{P}^{p(K)}(K),\ K \in \mathcal{T} \right\}. \tag{3.4}$$

Due to the discontinuous nature of the functions in $V_{\mathcal{T}}^p$, they may not be part of the classical Sobolev spaces $W^{m,p}(\Omega)$ but rather of a broken Sobolev space:

Definition 3.12. *For $m \geq 0$ and $1 \leq p \leq \infty$, we define the* broken Sobolev *space*

$$W^{m,p}(\mathcal{T}) = \left\{ v \in L^2(\Omega) : v|_K \in W^{m,p}(K) \right\} \tag{3.5}$$

equipped with the norm

$$\|v\|_{W^{m,p}(\mathcal{T})} = \left(\sum_{K \in \mathcal{T}} \|v\|^p_{W^{m,p}(K)} \right)^{1/p},$$

see also Section 2.1.2 for the definition of (unbroken) Sobolev spaces.

In particular, differential operators on these spaces are to be understood piecewise and not in the sense of distributions [11].

One can see that while the space $V_{\mathcal{T}}^p$ is conceptually very close to the classic finite element space $\mathsf{P}^k(\mathcal{T})$, no global constrains such as inter-element continuity are imposed. This fact also directly influences the construction of the basis functions of $V_{\mathcal{T}}^p$. While not directly necessary for general DG methods, we require the local basis functions to be of nodal kind, i. e. the coefficients of each function $v \in \mathsf{P}^k(K)$ should be uniquely defined by the nodal values $v(x_i)$ in a set of nodes $\{x_i\}$ to be chosen:

Definition 3.13. *For $K \in \mathcal{T}$, and a set of $n_K := \dim(\mathsf{P}^{p(K)}(K))$ pairwise distinct nodes $\mathbb{X}_K = \left\{x_1, \ldots, x_{n_K}\right\} \subset K$, we say a basis $\left\{\phi_i^{\mathbb{X}_K} : i = 1, \ldots, n_K\right\}$ of $\mathsf{P}^{p(K)}(K)$ is* nodal*, if*

$$\phi_i^{\mathbb{X}_K}(x_j) = \begin{cases} 1 & \textit{if } i = j, \\ 0 & \textit{else,} \end{cases}.$$

Since the interpolation in n_K nodes produces a unique polynomial, we have that the choice of nodes $\mathbb{X}_K$ (called *Lagrange nodes*) induces a local basis. For notational convenience, we may later suppress $\phi_i^{\mathbb{X}_K}$'s dependence on $\mathbb{X}_K$ if the particular choice of nodes has been made clear from the context.

Remark 3.14. For $\mathsf{P}^k(K) = \mathcal{Q}^k(K)$, the set of nodes is usually generated by the tensor product of a set of nodes on the real line. These might be equidistant or rather based on a quadrature rule such as Gauss–Lobatto. For example, if the $\hat{\mathbb{X}}_K$ is the set of Gauss-Lobatto nodes in $[0, 1]$ of size $k + 1$, one has

$$\mathbb{X}^K = F_K(\hat{\mathbb{X}}_K \times \ldots \hat{\mathbb{X}}_K).$$

The corresponding local basis would be the tensor product of the scalar Lagrange polynomials defined by the scalar nodes $\hat{\mathbb{X}}_K$ composed with F_K^{-1}.

While the construction of local basis functions is similar to those for continuous finite element spaces, we do not explicitly require that there are any Lagrange nodes on the element boundaries.

Assume for now that for every $K \in \mathcal{T}$ a corresponding node set $\mathbb{X}^K$ of appropriate (with respect to the reference elements and $p(K)$) size has been chosen. Using the construction of the local basis functions, we can easily generalize this concept to a global basis of our DG space $V_{\mathcal{T}}^p$.

Definition 3.15. *Given an element $K \in \mathcal{T}$ and a local nodal basis $\left\{\phi_i^{\mathbb{X}_K}\right\}$, we can define the* global basis function

$$\varphi_i^K(x) = \begin{cases} \phi_i^{\mathbb{X}_K}(x) & \textit{if } x \in \overline{K}, \\ 0 & \textit{else,} \end{cases}, \qquad i = 0, \ldots, \dim\left(\mathsf{P}^{p(K)}(K)\right). \tag{3.6}$$

It is clear from the definition that each basis function has support only on the closure of the element K it corresponds to. In fact, DG functions can only be evaluated with respect to a given element K. The downside to this approach is that a global finite element function is only well-defined if $x \in K$ for a single $K \in \mathcal{T}$. For $x \in e$ for some inner face $e \in \Gamma_h$, we have values from all adjacent elements. Since the inner faces form a set of Lebesgue-measure zero, we do not need to define function values there in the variational setting. However, it will be necessary to introduce the following notions of *jumps* and *averages* across adjacent elements.

Definition 3.16. *For any point x in a given inner face $\overline{K_0} \cap \overline{K_1} \in \Gamma_h$, we denote the unit normal vector* pointing from K_0 into K_1 *by $\mathbf{n}_0(x)$ and analogously the unit normal pointing from K_1 into K_0 by $\mathbf{n}_1(x)$. In particular, this implies $\mathbf{n}_0 = -\mathbf{n}_1$.*

If only a single element K is considered, we call the outward pointing normal $\mathbf{n}_K(x)$ for $x \in \partial K$.

Similarly, for x on the domain's boundary (in particular on a face $e \in \Gamma_b$), we denote the unit normal pointing outside Ω by $\mathbf{n}(x)$.

As usual, we may drop the dependence on x if it is clear from the context where the normal is evaluated.

Definition 3.17. *Let v be piecewise continuous, i. e. $v \in C^0(\mathcal{T})$. For a face $e = \overline{K_0} \cap \overline{K_1} \in \Gamma_h$, we define the* jump *on e by*

$$[\![v]\!] = v|_{K_0}\mathbf{n}_0 + v|_{K_1}\mathbf{n}_1. \tag{3.7}$$

The average *across the face e is defined by*

$$\{v\} = \frac{1}{2}\left(v|_{K_0} + v|_{K_1}\right). \tag{3.8}$$

For notional convenience, we also define the $[\![\cdot]\!]$ and $\{\cdot\}$ operators for boundary faces $e \in \Gamma_b$ with $e \subset \partial K$:

$$\begin{aligned}\{v\} &= v|_K,\\ [\![v]\!] &= v|_K\mathbf{n}.\end{aligned}$$

Similarly, for vector-valued functions $\mathbf{v} \in (C^0(\mathcal{T}))^N$, we define the jump *and* average *across an interior face $\overline{K_0} \cap \overline{K_1}$ by*

$$\begin{aligned}[\![\mathbf{v}]\!] &= \mathbf{v}|_{K_0} \cdot \mathbf{n}_0 + \mathbf{v}|_{K_1} \cdot \mathbf{n}_1,\\ \{\mathbf{v}\} &= \frac{1}{2}\left(\mathbf{v}|_{K_0} + \mathbf{v}|_{K_1}\right),\end{aligned}$$

where "$\cdot$" denotes the Euclidian scalar product. For a boundary face $e \in \Gamma_b$, $e \in \partial K$, we say

$$\{\mathbf{v}\} = \mathbf{v}|_K.$$

Remark 3.18. The particular definition for the jumps and averages given in Definition 3.17 frees us from assigning an artifical ordering between the elements since the definitions are invariant under permutations of K_0's and K_1's indices.

Other authors might assume an ordering and use a definition that does not involve the outer normal.

Before we introduce a specific DG scheme, we will define the interpolation operator Π_h.

Definition 3.19. *For (piecewise) continuous functions, we define the interpolation operator $\Pi_h : C^0(\mathcal{T}) \to V^p_{\mathcal{T}}$ by locally interpolating in the chosen node set, i. e. we have that for every $K \in \mathcal{T}$,*

$$\Pi_h v|_K \in \mathsf{P}^{p(K)}(K), \quad v \in C^0(\mathcal{T})$$

is the unique tensor-product polynomial such that

$$\Pi_h v|_K(x_i) = v(x_i) \quad \forall x_i \in \mathbb{X}_K.$$

3.2.2. Symmetric Interior Penalty DG

The Interior Penalty Discontinuous Galerkin method was introduced for elliptic problems (in a nonsymmetric formulation) by Mary Wheeler in [113] and analyzed in greater detail in Douglas Arnold's PhD thesis[10] and the resulting paper [11]. There, also the symmetric variant is considered.

Relying on Nitsche's approach to enforce boundary data [90], Arnold establishes a method to use functions that are of the form (3.4) to solve parabolic problems numerically.

As a motivation, we will sketch the derivation of the Symmetric Interior Penalty Method (SIPG) by considering the following simple model problem:

Problem 3.20. *Find $u \in C^2(\Omega) \cap C^0(\overline{\Omega})$ such that*

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

for suitable f and g.

For a function $u_0 \in H^1(\Omega)$ which equals g on the boundary, the weak formulation on the ("unbroken") Sobolev space is [31] to find $u \in H^1(\Omega)$ such that

$$(\nabla u, \nabla v) = (f, v) \quad \forall v \in H_0^1(\Omega), \tag{3.9}$$

$$u - u_0 \in H_0^1(\Omega). \tag{3.10}$$

For these classical Sobolev spaces, one directly translates the $(\nabla u, \nabla v)$ into the symmetric bilinear form

$$a(u, v) = (\nabla u, \nabla v)\,.$$

When we assume the solution u to Problem 3.20 to be reasonably smooth, say at least $u \in H^2(\Omega)$, we can derive a weak formulation tested with broken Sobolev functions. First, we state the following integration by parts formula, see also [11].

Lemma 3.21. *Let $\phi \in H^1(\mathcal{T})^N$ and $\psi \in H^1(\mathcal{T})$. Then, by applying integration by parts, one has*

$$\begin{aligned} \int_\Omega -\nabla \cdot \phi\psi \,\mathrm{d}x &= \sum_{K\in\mathcal{T}} \int_K -\nabla \cdot \phi\psi \,\mathrm{d}x \\ &= \sum_{K\in\mathcal{T}} \int_K \phi\nabla\psi \,\mathrm{d}x - \int_{\partial K} (\phi \cdot \mathbf{n}_K)\,\psi \,\mathrm{d}S \\ &= (\phi, \nabla\psi) - \int_{\partial\Omega} (\phi \cdot \mathbf{n})\,\psi \,\mathrm{d}S - \sum_{e\in\Gamma_h} \int_e \{\phi\} [\![\psi]\!] + \{\psi\} [\![\phi]\!] \,\mathrm{d}S. \end{aligned} \tag{3.11}$$

Proof. As the bulk parts $\int_K \phi\nabla\psi \,\mathrm{d}x$ are obvious, we only consider the lower dimensional face integrals

$$\int_{\partial K} (\phi \cdot \mathbf{n}_K)\,\psi \,\mathrm{d}S.$$

By construction of the grid, the boundary of any element $K \in \mathcal{T}$ can be decomposed into faces, i. e.

$$\int_{\partial K} (\phi \cdot \mathbf{n}_K)\, \psi \,\mathrm{d}S = \sum_{e \in \Gamma_K} \int_e (\phi \cdot \mathbf{n}_K)\, \psi \,\mathrm{d}S.$$

If we sum over all integrals on the boundary faces, we get

$$\sum_{e \in \Gamma_b} \int_e (\phi \cdot \mathbf{n})\, \psi \,\mathrm{d}S = \int_{\partial\Omega} (\phi \cdot \mathbf{n})\, \psi \,\mathrm{d}S,$$

which is the second term in (3.11).

Now we consider the inner faces. As we sum over all elements $K \in \mathcal{T}$, we will visit each face $e \in \Gamma_h$ twice, i. e. for $e = K_0 \cap K_1 \in \Gamma_h$, we have a summand

$$\int_e \left(\phi|_{K_0} \cdot \mathbf{n}_0\right) \psi|_{K_0} \,\mathrm{d}S + \int_e \left(\phi|_{K_1} \cdot \mathbf{n}_1\right) \phi|_{K_1} \,\mathrm{d}S. \tag{3.12}$$

Using $\mathbf{n}_1 = -\mathbf{n}_0$, expanding all terms in $\int_e \{\phi\} [\![\psi]\!] + \{\psi\} [\![\phi]\!] \,\mathrm{d}S$ and noticing how cross terms cancel each other out, we can rewrite (3.12) by:

$$\int_e \left(\phi|_{K_0} \cdot \mathbf{n}_0\right) \psi|_{K_0} \,\mathrm{d}S + \int_e \left(\phi|_{K_1} \cdot (-\mathbf{n}_0)\right) \phi|_{K_1} \,\mathrm{d}S = \int_e \{\phi\} [\![\psi]\!] + \{\psi\} [\![\phi]\!] \,\mathrm{d}S.$$

Summing over all $e \in \Gamma_h$, we get the last term in (3.11). □

Having established the partial integration formula (3.11), we can apply it to the solution u of the model problem. Since it is assumed that u is reasonable smooth, i. e. $u \in H^2(\Omega)$, we know that $[\![\nabla u]\!] = 0$ on every interior face. Thus, (3.11) with $\phi = \nabla u$ and $\psi = v \in H^1(\mathcal{T})$ reads

$$\int_\Omega (-\nabla \cdot \nabla u) v \,\mathrm{d}x = (\nabla u, \nabla v) - \int_{\partial\Omega} (\nabla u \cdot \mathbf{n})\, v \,\mathrm{d}S - \sum_{e \in \Gamma_h} \int_e \{\nabla u\} [\![v]\!] \,\mathrm{d}S.$$

First, to symmetrize the expression (3.11), we use the fact that $[\![u]\!] = 0$ on every face $e \in \Gamma_h$ and get

$$-(\Delta u, v) = (\nabla u, \nabla v) - \int_{\partial\Omega} \nabla u \cdot \mathbf{n} v \,\mathrm{d}S - \sum_{e \in \Gamma_h} \int_e \{\nabla u\} [\![v]\!] + \{\nabla v\} [\![u]\!] \,\mathrm{d}S \tag{3.13}$$

Remark 3.22. One could have as well changed the sign in front of the extra term $\int_e \{\nabla v\} [\![u]\!] \,\mathrm{d}S$. This would then have lead to the Nonsymmetric Interior Penalty DG variant, which is also common.

So far, a bilinear form based on (3.13) would not be coercive and a numerical scheme based on this equation would not have a unique solution. In particular, the numerical scheme would have no incentive to drive towards an (at least approximatively) continuous solution. Therefore one can introduce another term, again using $[\![u]\!] = 0$. For a given penalty factor $\sigma > 0$ (or rather, a given family of penalty factors $\{\sigma_e\}_e$, $\sigma_e > 0$), whose particular choice will be discussed later, we add the penalty term

$$\frac{\sigma_e}{|e|} \int_e [\![u]\!][\![v]\!] \, \mathrm{d}S$$

for every face $e \in \Gamma_h$. This gives

$$\begin{aligned} -(\Delta u, v) &= (\nabla u, \nabla v) - \langle \nabla u, v \rangle_{\partial\Omega} \\ &\sum_{e \in \Gamma_h} - \int_e \{\nabla u\} [\![v]\!] + \{\nabla v\} [\![u]\!] \, \mathrm{d}S + \frac{\sigma}{|e|} \int_e [\![u]\!][\![v]\!] \, \mathrm{d}S. \end{aligned} \tag{3.14}$$

Of particular elegance is the fact that with the notation of averages and jumps on boundary faces introduced in Definition 3.17, we can introduce a weak enforcement of Dirichlet data in a way Nitsche [90] introduced in 1971. This allows us to drop the boundary data requirement on the test space, i. e. we consider test functions $v \in H^1(\mathcal{T})$.

In Nitsche's approach for the model Problem 3.20, one can drop the boundary requirements on the ansatz space by introducing extra terms to the bilinear form and the right hand side. Using the notation $\langle v, w \rangle_M = \int_M vw \, \mathrm{d}S$ for a $(N-1)$-dimensional manifold M and the $(N-1)$-dimensional Hausdorff-measure S, Nitsche formulated a problem which is equivalent to

$$(\nabla u, \nabla v) - \left\langle u, \frac{\partial v}{\partial \mathbf{n}} \right\rangle_{\partial\Omega} - \left\langle v, \frac{\partial u}{\partial \mathbf{n}} \right\rangle_{\partial\Omega} + \mu \langle u, v \rangle_{\partial\Omega} = (f, v) + \left\langle \frac{\partial v}{\partial \mathbf{n}} + \mu v, g \right\rangle_{\partial\Omega}, \tag{3.15}$$

where $\mu > 0$ is a penalty factor to be chosen. Consider for now a continuous FE space with piecewise linear functions, i. e. $\mathcal{P}^1(\Omega)$ or $\mathcal{Q}^1(\Omega)$. Nitsche showed that if μ scales as η/h for a suitable large constant η, one gets optimal convergence properties in both the L^2- and H^1-norm [90].

If we apply the notions $[\![v]\!] = v\mathbf{n}$ and $\{v\} = v$ on boundary faces as introduced in Definition 3.17 and use $u = g$ on $\partial\Omega$, we can rewrite (3.15) for a given finite element space $V^p_{\mathcal{T}}$ defined on $\mathcal{T}$ by

$$\begin{aligned} \sum_{K \in \mathcal{T}} (\nabla u, \nabla v)_K + \sum_{e \in \Gamma_b} \left(- \int_e \left[\{\nabla v\} [\![u]\!] + \{\nabla u\} [\![v]\!] \right] \mathrm{d}S + \mu \int_e [\![u]\!][\![v]\!] \, \mathrm{d}S \right) \\ = (f, v) + \sum_{e \in \Gamma_b} \int_e (\partial_{\mathbf{n}} v + \mu v) g \, \mathrm{d}S. \end{aligned} \tag{3.16}$$

If one defines $\mu = \frac{\sigma}{|e|}$, the boundary face terms have exactly the same form as for the inner faces in (3.14). Hence, we can state the final form of the symmetric interior penalty method by defining the bilinear form

$$\begin{aligned} a_h(v, w) = \sum_{K \in \mathcal{T}} (\nabla v, \nabla w)_K + \sum_{e \in \Gamma} - \int_e \{\nabla v\} [\![w]\!] + \{\nabla w\} [\![v]\!] \, \mathrm{d}S \\ + \frac{\sigma}{|e|} \int_e [\![v]\!][\![w]\!] \, \mathrm{d}S. \end{aligned} \tag{3.17}$$

Picking up the boundary terms from the Nitsche method, we also define the right hand side

$$F_h(v) = (f, v) + \sum_{e \in \Gamma_b} \int_e \left(-\partial_{\mathbf{n}} v + \frac{\sigma}{|e|} v \right) g \, \mathrm{d}S. \tag{3.18}$$

We will frequently write $F_h v$ instead of $F_h(v)$ to emphasize the linearity of the functional.

Remark 3.23. Neumann boundary data can be included in a similar way, see e. g. [94] (but beware of the misplaced sign, see their erratum). Assume Γ_D to be the subset of the boundary faces Γ_b where Dirichlet data g_D is imposed and Γ_N to be the part where Neumann data $\frac{\partial u}{\partial \mathbf{n}} = g_N$. Then, we have the right hand side

$$F_h v = (f, v) + \sum_{e \in \Gamma_D} \int_e \left(-\partial_{\mathbf{n}} v + \frac{\sigma}{|e|} v \right) g_D \, \mathrm{d}S + \sum_{e \in \Gamma_N} \int_e v g_N \, \mathrm{d}S \tag{3.19}$$

and the bilinear form

$$\begin{aligned} a_h(v, w) = \sum_{K \in \mathcal{T}} (\nabla v, \nabla w)_K + \sum_{e \in \Gamma_h \cup \Gamma_D} \Bigg(&- \int_e \left[\{\nabla v\} [\![w]\!] + \{\nabla w\} [\![v]\!] \right] \mathrm{d}S \\ &+ \frac{\sigma}{|e|} \int_e [\![v]\!][\![w]\!] \, \mathrm{d}S \Bigg). \end{aligned} \tag{3.20}$$

Note how the boundary faces with Neumann boundary conditions are left out in the DG bilinear form.

Remark 3.24. For simplicity, we treated σ as a constant in our derivation. In practice, however, one can (and should!) let σ depend on the current face, i. e. σ is a function $e \mapsto \sigma(e) := \sigma_e > 0$. This is particularly important if non uniform ansatz degrees are used. We may emphasize this notion by writing σ_e if appropriate.

While most of the cited authors in this section assume a constant penalty parameter for simplicity, most arguments and estimates are face-local and can be generalized to the face-dependent penalty function.

Having derived the bilinear form (3.17) and right hand side (3.18), we can formulate the discretized version of Problem 3.20.

Problem 3.25. *For a given penalty parameter (function) $\sigma_e > 0$, find $u_h \in V_{\mathcal{T}}^p$ such that*

$$a_h(u_h, v) = F_h v \quad \forall v \in V_{\mathcal{T}}^p. \tag{3.21}$$

We have constructed the method such that a_h contains (besides the usual bulk terms) all the boundary integrals arising from partial integration when testing with functions from a broken Sobolev space. Moreover, by assuming that $u \in H^2(\Omega)$, we were able to add additional terms to create a symmetric bilinear form. These extra terms, however, vanish when u is used as an ansatz function since u has no jumps on the faces by assumption. Hence the solution u to the continuous problem satisfies equation (3.21), i. e. $a_h(u, v) = F_h v$ for all $v \in V_{\mathcal{T}}^p$ by construction. Therefore, we call the discretization *consistent*.

Boundedness and Coercivity

To show that the former problem is well-posed, we have to show boundedness and coercivity of the bilinear form in suitable norms. The particular choices of DG specific norms vary in the literature. On $V_{\mathcal{T}}^{p}$, the different norms should be equivalent. Here, we will select the norms as used in [35].

For the boundedness, we consider a norm (depending on the mesh and σ) that is an upper bound to the energy product $a_h(v, v)$:

$$\|v\|_h^2 = |v|_1^2 + \sum_{e \in \Gamma} \left(\frac{|e|}{\sigma_e} \int_e \{\nabla v\}^2 \,\mathrm{d}S + \frac{2\sigma_e}{|e|} \int_e [\![v]\!]^2 \,\mathrm{d}S \right). \tag{3.22}$$

We can bound $a_h(\cdot, \cdot)$ in this norm:

Lemma 3.26. *For all $v, w \in H^1(\Omega) + V_{\mathcal{T}}^{p}$, it holds*

$$a_h(v, w) \leq \|v\|_h \|w\|_h .$$

Proof. Apply Young's inequality, see also [35, Exercise 10.x.32]. □

Remark 3.27. For other choices of the norm $\|\cdot\|_h$, one gets an upper bound

$$a_h(v, w) \leq C_b \|v\|_h \|w\|_h \tag{3.23}$$

with a constant $C_b > 0$. With our choice of $\|\cdot\|_h$, it holds $C_b = 1$ [35]. To be more independent of the particular choice, we will use (3.23) when referring to the boundedness of the bilinear form.

To derive coercivity of the bilinear form, we need another norm, namely

$$|\!|\!|v|\!|\!|^2 = |v|_1^2 + \sum_{e \in \Gamma} \sigma_e \int_e [\![v]\!]^2 \,\mathrm{d}S. \tag{3.24}$$

As mentioned before, the penalty parameter might depend on the particular face it is evaluated on.

Remark 3.28. To simplify notation, we suppress the dependence of the norm on the grid and penalty parameters where possible. If we need to make the dependence explicit, we will do so by appending the corresponding discrete space as a subscript, e. g. $|\!|\!|\cdot|\!|\!|_{V_{\mathcal{T}}^{p}}$.

Without loss of generality, assume $|e| \leq 2$ for all faces $e \in \Gamma$. Obviously, we have

$$|\!|\!|v|\!|\!| \leq \|v\|_h \quad \forall v \in H^1(\Omega) + V_{\mathcal{T}}^{p}.$$

On the other hand, we have the equivalence *on the discrete space* by the following inequality:

Lemma 3.29. *There is a $C > 0$ depending only on the grid quality (i. e. ρ and θ from Definition 3.7), such that*

$$\|v\|_h \leq C(1 + \sigma^{-1}) |\!|\!|v|\!|\!| \quad \forall v \in V_{\mathcal{T}}^{p}.$$

Proof. See [35, Lemma 10.5.15]. □

Equipped with this norm, we may state the following stability relation:

Lemma 3.30. *There is a $\sigma_0 > 0$ depending on the grid quality and the ansatz degree p such that*

$$a_h(v, v) \geq \frac{1}{2} |||v|||^2 \quad \forall v \in V_{\mathcal{T}}^p \tag{3.25}$$

provided the penalty parameter σ was chosen larger than σ_0 on every face $e \in \Gamma_h$.

Proof. For piecewise linear elements, see [35, Lemma 10.5.19]. The more general case is discussed e. g. in [53] and the references therein. □

Note that the coercivity only holds on the discrete space but not on the full energy space $H^1(\Omega) + V_{\mathcal{T}}^p$. While this is sufficient to guarantee the existence of unique discrete solutions, it is a source of difficulty for the numerical analysis.

Remark 3.31. Using the equivalence of $\|\cdot\|_h$ and $|||\cdot|||$ on the discrete space, we have that there is a constant $C_s > 0$ such that

$$a_h(v, v) \geq C_s \|v\|_h^2 \quad \forall v \in V_{\mathcal{T}}^p. \tag{3.26}$$

Another issue is the practical choice of the penalty parameter. As Lemma 3.30 shows, it needs to exceed a certain threshold. This particular threshold, however, is often not known. For a grid consisting of simplices, the penalty parameter can be estimated using results from [53]. These estimates can be applied for each intersection of two elements separately and depend on the smallest angle of the involved elements as well as the polynomial degree employed in the discrete space for the respective elements.

In general it is expected that σ should asymptotically scale as p^2, therefore some authors also write

$$\sigma = \mu p^2 \tag{3.27}$$

for an appropriate constant μ which only depends on the grid quality.

For the rest of this work, we tacitly assume that σ is of form (3.27) and is large enough in the sense of Lemma 3.30.

Remark 3.32. While in our derivation we only considered a model problem of the form $-\Delta u = f$, one can easily generalize this to equations of the form (2.6).

The bilinear form (3.17) then reads

$$\begin{aligned} a_h(v, w) = &\sum_{K \in \mathcal{T}} \left[(A\nabla v, \nabla w)_K + \alpha(v, w)_K \right] \\ &+ \sum_{e \in \Gamma_h \cup \Gamma_D} - \int_e \{A\nabla v\} [\![w]\!] + \{A\nabla w\} [\![v]\!] \,\mathrm{d}S \\ &+ \frac{\sigma}{|e|} \int_e [\![v]\!] [\![w]\!] \,\mathrm{d}S. \end{aligned} \tag{3.28}$$

The coercivity and boundedness can also be shown for $a_h(\cdot, \cdot)$ being defined as in (3.28). In particular, (3.17) is a special case of (3.28) where $A = \mathrm{Id}$ and $\alpha = 0$. Thus, in the following, we will assume $a_h(\cdot, \cdot)$ being defined as in (3.28).

Interpolation Estimate

Consider a DG space $V_{\mathcal{T}}^p$ with $p(K) \equiv p$ constant for all elements. Recall from Definition 3.19 that Π_h interpolates a given function locally on each element. Given the local approximation property

$$|u - \Pi_h u|_{s,K} \lesssim h^{p+1-s} |u|_{p+1,K}, \tag{3.29}$$

(see also the Bramble–Hilbert lemma [42]), we get the following interpolation error:

Lemma 3.33. *Let $u \in H^q(\mathcal{T})$ with $q \geq p+1$. Then, it holds*

$$\|u - \Pi_h u\|_h \lesssim h^p |u|_{p+1}. \tag{3.30}$$

Proof. Use the arguments from [11] and adapt to the different norm. See also [35]. □

3.3. Discretization of the Obstacle Problem

As an example for the use of (possibly high order) DG methods for variational inequalities of the first kind, we will discuss the discretization of the obstacle problem as stated in Section 2.2.1 and prove some a priori error estimates.

While several error estimates for the finite element discretization of the obstacle problem are known, we contribute several new and important results. First, we show that using DG finite elements *of arbitrary order* will lead to a convergent scheme under mild assumptions on the solution. While it might sound obvious that minimizing a convex functional in a larger space leads also to a convergent solution, we have to take into account that the functional *depends* on the current discretization: First, we have that a penalty parameter has to be chosen such that it approximately scales with p^2, thus changing the underlying quadratic energy with increasing polynomial degree. Second, the way we restrict the admissible set depends on the choice of the basis, see below, thus we minimize over different sets for varying polynomial degree. In fact, our numerical examples (see Chapter 7) indicate that a higher polynomial degree does not always imply a lower discretization error.

Further, we show that for problems which have somewhat smoother solutions, one can expect convergence rates of order $\mathcal{O}(h^{1.5-\varepsilon})$ if using finite elements of order 2 or higher. While a similar result for quadratic DG elements was obtained in [111], our result is substantially improved in the sense that we allow for arbitrary order polynomials (as long as their order is 2 or higher) and, more importantly, we do so under more realistic assumptions. More precisely, in [111] it was assumed that the solution of the obstacle problem is in $H^3(\Omega)$ which is in general not true. Indeed, one has $u \notin H^3$ even for simple counterexamples. In our case, on the other hand, we only require $u \in H^{2.5-\varepsilon}(\Omega)$, which is feasible under certain conditions, cf. (2.8).

In addition to these results that rely on (quasi-)uniform grids, we also prove a result which shows that one can exploit the fact that the solution of the obstacle problem has locally higher regularity in regions which do not intersect the free boundary. Assuming a grid which is locally refined such that the free boundary is finer resolved than the rest of the domain (where the solution is assumed to be smoother), we show that one can actually gain convergence orders higher than 1.5 for higher order ansatz functions in these regions. Assuming the grid elements near the free boundary are smaller by an appropriate number of orders, a global high order convergence could be expected.

As a reminder, we will once more state the obstacle problem as given in Problem 2.8: Let L be a second order differential operator of the form (2.6), i. e.

$$Lu = -\operatorname{div}(A\nabla u) + \alpha u,$$

with smooth coefficients $A_{ij}(x) \in C^1(\overline{\Omega})$ and $\alpha \geq 0$. This gives rise to a bilinear form

$$a(v,w) = (A\nabla v, \nabla w) + \alpha\,(v,w)\,.$$

For a given functional l, we seek to solve the obstacle problem of finding $u \in \mathcal{K}$ such that

$$a(u, v-u) \geq l(v-u) \quad v \in \mathcal{K},$$

where the convex set $\mathcal{K}$ is generated by two obstacle functions $\underline{\psi}$ and $\overline{\psi}$,

$$\mathcal{K} = \left\{ v \in H : \underline{\psi} \leq v \leq \overline{\psi} \text{ a.e.} \right\},$$

H being a closed affine subspace of $H^1(\Omega)$ that obeys some boundary conditions.

When discretizing Problem 2.8 with DG finite elements, a central aspect is how to discretize the admissible set $\mathcal{K}$. For piecewise linear finite elements ($\mathcal{P}^1$ or $\mathcal{Q}^1$), this is usually done by interpolating the obstacle function in the finite element space and controlling the obstacle conditions in the interpolation nodes. As the ansatz functions are linear between these nodes, it is obvious that any function that does not violate the obstacle in these nodes will also not violate the interpolated obstacle function. For any function of polynomial degree greater than one, however, this may not hold anymore. This puts additional burden on the analysis as it invalidates some commonly used arguments. Nevertheless, we still discretize the set $\mathcal{K}$ by controlling in the interpolation points.

In the following, we assume that a certain grid $\mathcal{T}$ and local polynomial degrees $p : \mathcal{T} \to \mathbb{N}$ have been chosen such that our finite element space $V_{\mathcal{T}}^p$ is well defined. Moreover, we assume that the element-local node sets $\{\mathbb{X}_K\}_K$ were chosen such that $\mathbb{X}_K$ are based on a proper quadrature rule. More precisely, we require the weights of such a quadrature rule to be positive and the rule to be able to integrate polynomials of order at least $p(K)$ exactly:

Assumption 3.34. *For each K, we require that the Lagrange nodes $\mathbb{X}_K$ are such that the induced basis functions (see Definition 3.13) have positive integral, i. e.*

$$w_i^K := \int_K \phi_i^{\mathbb{X}_K}\,\mathrm{d}x > 0 \quad \forall i \in \left\{0, \ldots, \dim\left(\mathsf{P}^{p(K)}(K)\right)\right\}. \tag{3.31}$$

For a polynomial $v \in \mathsf{P}^{p(K)}(K)$, we have

$$\int_K v\,\mathrm{d}x = \sum_i v(x_i) w_i^K, \tag{3.32}$$

where $x_i \in \mathbb{X}_K$ are the Lagrange nodes on the element K.

Assumption 3.34 shows why we suggested to take tensor-products of quadrature rules such as Gauss–Lobatto when using $\mathcal{Q}^k$ elements. For finite element functions on simplex-based elements, the choice of suitable Lagrange nodes and the construction of appropriate local basis functions might be more delicate and will not be discussed here. The usage of Fekete points in a triangle [108] might lead to an approach that is similar to the construction of using Gauss–Lobatto nodes for $\mathcal{Q}^k$. In [108] points that satisfy our requirements for moderate polynomial orders are computationally approximated.

Definition 3.35. *For given obstacle functions $\underline{\psi}, \overline{\psi}$, we define the discrete admissible set by*

$$\mathcal{K}_{hp} = \left\{ v \in V : \underline{\psi}|_K(x) \leq v|_K(x) \leq \overline{\psi}|_K(x) \quad \forall K \in \mathcal{T}\, \forall x \in \mathbb{X}_K \right\}. \tag{3.33}$$

We do not require any specific boundary conditions for the set $\mathcal{K}_{hp}$. These will be weakly enforced by the bilinear form as indicated in Section 3.2.2.

Remark 3.36. Note that since the node sets $\mathbb{X}_K$ are not necessarily subsets of each other for different p, in general we do *not* have $\mathcal{K}_{hp} \subset \mathcal{K}_{h\tilde{p}}$ with $p \neq \tilde{p}$.

After having found a suitable discrete representation of $\mathcal{K}$ and defining the DG bilinear form $a_h(\cdot,\cdot)$ and functional F_h as in (3.28) and (3.19) respectively, we can state the discrete problem corresponding to Problem 2.8:

Problem 3.37. *Find $u_h \in \mathcal{K}_{hp}$ such that*

$$a_h(u_h, v - u_h) \geq F_h(v - u_h) \quad \forall v \in \mathcal{K}_{hp}. \tag{3.34}$$

Theorem 3.38. *Problem 3.37 has a unique solution $u_h \in \mathcal{K}_{hp}$.*

Proof. As for the continuous case, this is implied by the Lions–Stampacchia theorem [82]. □

By construction of the Lagrange basis, the obstacle condition as suggested in Definition 3.35 is particularly easy to verify by comparing the coefficients of the finite element functions:

Consider the interpolated obstacle function with coefficients $\underline{\psi}_i^K$, i. e.

$$\Pi_h \underline{\psi} = \sum_K \sum_i \underline{\psi}_i^K \varphi_i^K$$

and analogously $\overline{\psi}_i^K$ for $\Pi_h \overline{\psi}$. For a given finite element function $v = \sum_K \sum_i v_i^K \varphi_i^K \in V_{\mathcal{T}}^p$, we have

$$v \in \mathcal{K}_{hp} \iff \underline{\psi}_i^K \leq v_i^K \leq \overline{\psi}_i^K \quad \forall K \in \mathcal{T}\, \forall i \in \{0, \ldots, p\}^N .$$

As explained before, the given construction cannot ensure that the interpolated obstacle is not violated by a function $v \in \mathcal{K}_{hp}$, i. e.

$$v \in \mathcal{K}_{hp} \not\Rightarrow \underline{\psi} \leq v \text{ a. e. in } \Omega,$$
$$v \in \mathcal{K}_{hp} \not\Rightarrow v \leq \overline{\psi} \text{ a. e. in } \Omega.$$

We can, however, derive the following weaker result:

Lemma 3.39. *Let v be in $\mathcal{K}_{hp}$. Then it holds for every element $K \in \mathcal{T}$*

$$\int_K \Pi_h \underline{\psi} - v \, \mathrm{d}x \leq 0, \tag{3.35}$$
$$\int_K \Pi_h \overline{\psi} - v \, \mathrm{d}x \geq 0. \tag{3.36}$$

Proof. We will only show the first part, i. e. (3.35), as the second equation's proof is performed in the same manner.

Since $v \in \mathcal{K}_{hp} \subset V_{\mathcal{T}}^p$, we have $(\Pi_h \underline{\psi} - v)\Big|_K \in \mathsf{P}^p(K)$. By assumption on $\mathbb{X}_K$, the chosen quadrature rule has positive weights $\left\{w_i^K\right\}_i$ and can integrate polynomials of degree p exactly. Hence, we have

$$\int_K \Pi_h \underline{\psi} - v \, \mathrm{d}x = \sum_i (\Pi_h \underline{\psi} - v)\Big|_K (x_i) w_i^K \leq 0$$

since $(\Pi_h \underline{\psi} - v)\Big|_K (x_i) \leq 0$ for every node $x_i \in \mathbb{X}_K$ by definition of $\mathcal{K}_{hp}$. □

3.3.1. A Priori Error Estimates

In this chapter, we will discuss the discretization error $\|u - u_h\|_h$. A priori error estimates for finite element discretizations of the obstacle problem have been derived before by various authors. Among others, Brezzi et al.[38] derived $\mathcal{O}(h)$ convergence for piecewise linear finite elements. In the same article, $\mathcal{O}(h^{1.5-\varepsilon})$ convergence was proven for piecewise quadratic ansatz functions where the obstacle condition is controlled in the midpoints of the edges. The same result under weaker assumptions is stated in [112]. Similar estimates for piecewise linear and piecewise quadratic DG function spaces have been derived in [111]. The argument there, however, suffers from a regularity assumption on u, namely $u \in H^3(\Omega)$, which is not true in general. A convergence result for a p–FEM discretization where the obstacle condition is also controlled in Gauss–Lobatto nodes has been derived in [78]. There, the convergence is proven when increasing the polynomial degree p without altering the mesh width.

We will now derive several error estimates for our DG discretizations of Problem 3.37 which are not limited to linear or quadratic elements. At first, we generalize the results from [111] in the sense that we show that we obtain $\mathcal{O}(h)$ convergence for general P^k (possibly higher order) DG finite elements. Also, we show that if the continuous solution has certain smoothness properties, we get $\mathcal{O}(h^{1.5-\varepsilon})$ convergence for P^k elements with $k \geq 2$. Afterwards, we proceed to investigate whether we can get better than that:

Due to the limited regularity (see e. g. the monograph [74] for an extended discussion) of the continuous solution u, we cannot expect to get better global convergence rates than $\mathcal{O}(h^{1.5-\varepsilon})$, which raises the question whether the use of higher order ansatz functions is appropriate at all. Our motivation is that for sufficiently smooth data, u might be locally smoother than $H^{2.5}$ for those subregions of Ω that are not adjacent to the free boundary. For regions in the interior of $\Omega^l \subset \Omega$ where the obstacle is active, that is $u|_{\Omega^l} \equiv \underline{\psi}|_{\Omega^l}$, we obviously have that u has the same regularity as $\underline{\psi}$ (or as $\overline{\psi}$ in Ω^u respectively). Similarly, in the interior of $\Omega^+ \subset \Omega$ where u is strictly greater than $\underline{\psi}$ and strictly less than $\overline{\psi}$, we have that u's smoothness is controlled by the underlying linear PDE. Depending on the problem data, the error estimate for these regions might be of higher order than the global $\mathcal{O}(h^{1.5})$ bound. Since the nonsmooth regions near the free boundary will in general only converge with lower order, we will split our error estimates between these regions. In the nonsmooth regions, the mesh width is bounded by h_F while on the smoother parts a mesh width of h_C is used. To achieve a method of at least asymptotically higher order, h_F must be significantly smaller than h_C.

For the proof of convergence, we split our grid into subsets which are induced by the contact and non-contact areas of the domain Ω:

Definition 3.40. *We define the* contact region *and the regions where the solution is equal to the lower or upper obstacle, respectively, by*

$$\begin{aligned} \Omega^+ &= \left\{ x \in \Omega \mid \underline{\psi}(x) < u(x) < \overline{\psi}(x) \right\}, \\ \Omega^l &= \left\{ x \in \Omega \mid u(x) = \underline{\psi}(x) \right\}, \\ \Omega^u &= \left\{ x \in \Omega \mid u(x) = \overline{\psi}(x) \right\}. \end{aligned}$$

Similarly, for the grid $\mathcal{T}$, we define the subsets

$$\begin{aligned} \mathcal{T}^l &= \left\{ K \in \mathcal{T} \mid K \subset \Omega^l \right\}, \\ \mathcal{T}^u &= \left\{ K \in \mathcal{T} \mid K \subset \Omega^u \right\}, \\ \mathcal{T}^+ &= \left\{ K \in \mathcal{T} \mid K \subset \Omega^+ \right\}, \\ \mathcal{T}^b &= \mathcal{T} \setminus \left(\mathcal{T}^l \cup \mathcal{T}^u \cup \mathcal{T}^+ \right). \end{aligned}$$

Moreover, we have the following complementarity conditions for the solution $u \in H^2(\Omega)$ of the variational inequality (2.3), see e. g. [74],

$$Lu = f \quad \text{a.e. in } \Omega^+, \tag{3.37}$$

$$Lu \geq f \quad \text{a.e. in } \Omega^l, \tag{3.38}$$

$$Lu \leq f \quad \text{a.e. in } \Omega^u. \tag{3.39}$$

Later in the proofs, it will be convenient to have the following definition:

Definition 3.41. *For an element $K \in \mathcal{T}$, we denote the local $L^2(K)$-projection into the space of constant functions by*

$$\begin{aligned} P_K^0 &: L^1(K) \to \mathsf{P}^0(K), \\ P_K^0 v &\equiv \frac{1}{|K|}\int_K v \, \mathrm{d}x, \quad v \in L^1(K). \end{aligned}$$

The remainder term will be defined by

$$R_K^0 v = v - P_K^0 v.$$

Corollary 3.42. *Using the notation from Definition 3.41, we get from Lemma 3.39 the following (pointwise) inequalities*

$$\begin{aligned} P_K^0(\Pi_h \underline{\psi} - u_h) &\leq 0, \\ P_K^0(\Pi_h \overline{\psi} - u_h) &\geq 0. \end{aligned}$$

A similar variant of the following statement was also derived in [111], however only for $L = -\Delta$:

Proposition 3.43. *With $u \in H^2(\Omega)$ and $u_h \in V_{\mathcal{T}}^p$ being the solutions to* (2.3) *and* (3.34)*, respectively, we have*

$$a_h\left(u - u_h, \Pi_h u - u_h\right) \leq \sum_{K \in \mathcal{T}} -\int_K \left(f - Lu\right)\left(\Pi_h u - u_h\right). \tag{3.40}$$

Proof. Since $u \in H^2(\Omega)$, we have $[\![u]\!] = 0$ and $\{u\} = u$ on interior faces. Also, it holds that $\{\nabla u\} = \nabla u$ and $[\![\nabla u]\!] = 0$ on interior faces since $\nabla u \in H^1(\Omega)$, see e. g. [93, Proposition 3.2.1]. Moreover, $u = g_D$ on Γ_D and $\partial_{\mathbf{n}} u = g_N$ on Γ_N.

Defining $F_{\mathrm{IP}}^D(v) = \sum_{e \in \Gamma_D} \int_e (-\partial_{\mathbf{n}} v + \frac{\sigma}{|e|} v) g_D \, \mathrm{d}S$ and $F_{\mathrm{IP}}^N(v) = \sum_{e \in \Gamma_N} \int_e v g_N \, \mathrm{d}S$, we have $F_h(v) = (f, v) + F_{\mathrm{IP}}^D(v) + F_{\mathrm{IP}}^N(v)$, cf. equation (3.19). Then, we have

$$\begin{aligned}
a_h\left(u, \Pi_h u - u_h\right) &= \sum_{K\in\mathcal{T}} \int_K A\nabla u \cdot \nabla(\Pi_h u - u_h)\,\mathrm{d}x + \alpha \int_\Omega u\left(\Pi_h u - u_h\right)\mathrm{d}x \\
&\quad - \sum_{e\in\Gamma_h\cup\Gamma_D} \int_e A\nabla u [\![\Pi_h u - u_h]\!]\,\mathrm{d}S + F_{\mathrm{IP}}^D(\Pi_h u - u_h) \\
&= - \sum_{K\in\mathcal{T}} \int_K \operatorname{div}\left(A\nabla u\right)\left[\Pi_h u - u_h\right]\mathrm{d}x + \alpha \int_\Omega u\left(\Pi_h u - u_h\right)\mathrm{d}x \\
&\quad + \sum_{K\in\mathcal{T}} \int_{\partial K} A\nabla u\left(\Pi_h u - u_h\right)\mathbf{n}_K\,\mathrm{d}S \\
&\quad - \sum_{e\in\Gamma_h\cup\Gamma_D} \int_e A\nabla u [\![\Pi_h u - u_h]\!]\,\mathrm{d}S + F_{\mathrm{IP}}^D\left(\Pi_h u - u_h\right) \\
&= - \int_\Omega \operatorname{div}(A\nabla u)\left[\Pi_h u - u_h\right]\mathrm{d}x + \alpha \int_\Omega u\left(\Pi_h u - u_h\right)\mathrm{d}x \\
&\quad + F_{\mathrm{IP}}^D\left(\Pi_h u - u_h\right) + F_{\mathrm{IP}}^N\left(\Pi_h u - u_h\right) \\
&= \int_\Omega Lu\left(\Pi_h u - u_h\right)\mathrm{d}x + F_{\mathrm{IP}}^D\left(\Pi_h u - u_h\right) + F_{\mathrm{IP}}^N\left(\Pi_h u - u_h\right).
\end{aligned}$$

Since u_h solves Problem 3.37 and $\Pi_h u \in \mathcal{K}_{hp}$, we have

$$a_h\left(u_h, \Pi_h u - u_h\right) \geq \left(f, \Pi_h u - u_h\right) + F_{\mathrm{IP}}^D\left(\Pi_h u - u_h\right) + F_{\mathrm{IP}}^N\left(\Pi_h u - u_h\right).$$

Combining both, we get (3.40). □

As a first result, we will prove a result that shows that a DG discretization of the obstacle problem as stated in Problem 3.37 converges with at rate at least $\mathcal{O}(h)$ when reducing the mesh size. While a similar result is known for piecewise linear ansatz functions [111], we want to make sure that higher order ansatz functions do not derail the convergence. This is not obvious since as remarked earlier, controlling the obstacle condition only in the Lagrange nodes might lead to solutions which do not obey the obstacle condition everywhere (not even for the interpolated obstacle).

As customary, we let C denote a generic constant whose precise value might change from line to line.

Theorem 3.44. *Assume $\underline{\psi}, \overline{\psi} \in H^2(\Omega)$. Let $u \in H^2(\Omega)$ be the solution to Problem 2.4 and u_h be the solution to the discrete Problem 3.37. Then, we have*

$$\left\|u - u_h\right\|_h \leq Ch. \tag{3.41}$$

Proof. The proof follows in parts the proofs for continuous [38, 112] and discontinuous [111] quadratic elements. Moreover, we will adopt some arguments made for the double obstacle case as in [110].

By the triangle inequality, we have

$$\left\|u - u_h\right\|_h \leq \left\|u - \Pi_h u\right\|_h + \left\|\Pi_h u - u_h\right\|_h. \tag{3.42}$$

Using the coercivity of the discrete bilinear form for *discrete* functions, we have

$$\begin{aligned} C_s\|\Pi_h u - u_h\|_h^2 &\leq a_h\left(\Pi_h u - u_h, \Pi_h u - u_h\right) \\ &= a_h\left(\Pi_h u - u, \Pi_h u - u_h\right) + a_h\left(u - u_h, \Pi_h u - u_h\right). \end{aligned} \tag{3.43}$$

By Young's inequality, it holds

$$\begin{aligned} a_h\left(\Pi_h u - u, \Pi_h u - u_h\right) &\leq C_b\|\Pi_h u - u\|_h\|\Pi_h u - u_h\|_h \\ &\leq \frac{C_s}{2}\|\Pi_h u - u_h\|_h^2 + \frac{C_b^2}{2C_s}\|\Pi_h u - u\|_h^2. \end{aligned}$$

Inserting into (3.43) and rearranging terms, we get

$$\|\Pi_h u - u_h\|_h^2 \leq \frac{C_b^2}{C_s^2}\|\Pi_h u - u\|_h^2 + \frac{2}{C_s}\underbrace{a_h\left(u - u_h, \Pi_h u - u_h\right)}_{=:T_1}. \tag{3.44}$$

It remains to estimate the term $T_1 = a_h\left(u - u_h, \Pi_h u - u_h\right)$, for which we have

$$T_1 \leq \sum_{K\in\mathcal{T}}\int_K -\left(f - Lu\right)\left(\Pi_h u - u_h\right)\mathrm{d}x, \tag{3.45}$$

by Proposition 3.43.

Combining everything we have so far by inserting (3.45) into (3.44) and using the result in (3.42), we can deduce the following inequality:

$$\begin{aligned} \|u - u_h\|_h \leq& \left(1 + \frac{C_b}{C_s}\right)\|\Pi_h u - u\|_h \\ &+ \left(\frac{2}{C_s}\sum_{K\in\mathcal{T}}\int_K -(f - Lu)(\Pi_h u - u_h)\,\mathrm{d}x\right)^{1/2}. \end{aligned} \tag{3.46}$$

The first term on the right hand side can be estimated by the interpolation result, i. e. we know $\|\Pi_h u - u\|_h \lesssim h|u|_1$ (or even better if higher order polynomials are used).

It remains to handle the latter term, namely

$$\left(\frac{2}{C_s}\underbrace{\sum_{K\in\mathcal{T}}\int_K -(f - Lu)(\Pi_h u - u_h)\,\mathrm{d}x}_{=:T_2}\right)^{1/2}.$$

This term would vanish if we would consider the linear case without any (active) obstacles, as $f - Lu = 0$ would hold.

By the complementarity condition (3.37), we only need to consider those elements that are in $\mathcal{T}^l$, $\mathcal{T}^u$ or $\mathcal{T}^b$. For convenience, we define $w := -\left(f - Lu\right)$. Using this notation, we rewrite (3.45) as

$$\begin{aligned}
T_2 &= \sum_{K\in\mathcal{T}^l}\int_K w\left(\Pi_h u-u_h\right)\mathrm{d}x+\sum_{K\in\mathcal{T}^u}\int_K w\left(\Pi_h u-u_h\right)\mathrm{d}x+\sum_{K\in\mathcal{T}^b}\int_K w\left(\Pi_h u-u_h\right)\mathrm{d}x\\
&= \sum_{K\in\mathcal{T}\setminus\mathcal{T}^+}\int_{K\cap\Omega^+} w\left(\Pi_h u-u_h\right)\mathrm{d}x\\
&\qquad+\int_{K\cap\Omega^l} w\left(\Pi_h u-u_h\right)\mathrm{d}x+\int_{K\cap\Omega^u} w\left(\Pi_h u-u_h\right)\mathrm{d}x
\end{aligned}$$

Clearly, the first integral vanishes as $w\equiv 0$ on Ω^+ by (3.37), i. e.

$$T_2=\sum_{K\in\mathcal{T}\setminus\mathcal{T}^+}\int_{K\cap\Omega^l} w\left(\Pi_h u-u_h\right)\mathrm{d}x+\int_{K\cap\Omega^u} w\left(\Pi_h u-u_h\right)\mathrm{d}x.$$

We will now consider the integrals over $K\cap\Omega^l$. The integrals over $K\cap\Omega^u$ can be handled analogously.

Define w_l, as w restricted to Ω^l and extended by zero outside, i. e.

$$\begin{aligned}
w_l|_{\Omega^l} &= w,\\
w_l|_{\Omega\setminus\Omega^l} &\equiv 0.
\end{aligned}$$

This gives

$$\int_{K\cap\Omega^l} w\left(\Pi_h u-u_h\right)\mathrm{d}x=\int_K w_l\left(\Pi_h u-u_h\right)\mathrm{d}x.$$

This integral can be rewritten as

$$\begin{aligned}
\int_K w_l\left(\Pi_h u-u_h\right)\mathrm{d}x &= \underbrace{\int_K w_l\left(\Pi_h u-u+\underline{\psi}-\Pi_h\underline{\psi}\right)\mathrm{d}x}_{=:T_3}\\
&\qquad+\int_K w_l\left(u-\underline{\psi}\right)\mathrm{d}x+\underbrace{\int_K w_l\left(\Pi_h\underline{\psi}-u_h\right)\mathrm{d}x}_{=:T_4}.
\end{aligned}\tag{3.47}$$

Clearly, the middle integral vanishes as $u\equiv\underline{\psi}$ in Ω^l by definition.

The first integral of (3.47), T_3, vanishes on $\mathcal{T}^l$ and on $\mathcal{T}^u$. For the remaining elements, we have

$$\begin{aligned}\sum_{K\in\mathcal{T}^b} T_3 &= \sum_{K\in\mathcal{T}^b}\int_K w_l\left(\Pi_h u - u + \underline{\psi} - \Pi_h\underline{\psi}\right)\mathrm{d}x\\ &= \sum_{K\in\mathcal{T}^b}\int_K w_l\left(\underline{\psi} - u - \Pi_h(\underline{\psi}-u)\right)\mathrm{d}x\\ &= \left(w_l, \underline{\psi} - u - \Pi_h(\underline{\psi}-u)\right)_{\Omega^l}\\ &\leq \|w_l\|_0\left\|\underline{\psi} - u - \Pi_h(\underline{\psi}-u)\right\|_0\\ &\leq C\|w_l\|_0\, h^2\left|\underline{\psi}-u\right|_2.\end{aligned}$$

For the last integral from (3.47), T_4, (similarly to the proofs in [38, 111, 112]), we will use the local L^2-projection as defined in Definition 3.41 to estimate the remaining integrals. By Corollary 3.42, we have $P_K^0(\Pi_h\underline{\psi}-u_h)\,\mathrm{d}x \leq 0$ and thus, since $w_l \geq 0$ on K, we get

$$\int_K w_l P_K^0\left(\Pi_h\underline{\psi}-u_h\right)\mathrm{d}x \leq 0.$$

Thus, we can estimate

$$\begin{aligned}T_4 &= \int_K w_l\left(\Pi_h\underline{\psi}-u_h\right)\mathrm{d}x\\ &\leq \int_K w_l\left(\Pi_h\underline{\psi}-u_h\right)\mathrm{d}x - \int_K w_l P_K^0\left(\Pi_h\underline{\psi}-u_h\right)\mathrm{d}x\\ &= \int_K w_l R_K^0\left(\Pi_h\underline{\psi}-u_h\right)\mathrm{d}x\end{aligned}$$

We define R^0 piecewise by

$$R^0 v|_K = R_K^0 v.$$

Applying interpolation estimates for piecewise constant approximations, we get

$$\begin{aligned}\sum_{K\in\mathcal{T}\setminus\mathcal{T}^+}\int_K w_l R_K^0\left(\Pi_h\underline{\psi}-u_h\right) &= \left(w_l, R^0\left(\Pi_h\underline{\psi}-u_h\right)\right)_{\Omega^l}\\ &\leq \|w_l\|_0\left\|R^0\left(\Pi_h\underline{\psi}-u_h\right)\right\|_{0,\Omega^l}\\ &\lesssim h\|w_l\|_0\left|\Pi_h\underline{\psi}-u_h\right|_{1,\Omega^l}\\ &\leq h\|w_l\|_0\left(\left|\Pi_h\underline{\psi}-\underline{\psi}\right|_{1,\Omega^l}+\left|\underline{\psi}-u_h\right|_{1,\Omega^l}\right).\end{aligned}$$

The first term in the last line can be estimated by

$$\left|\Pi_h\underline{\psi} - \underline{\psi}\right|_{1,\Omega^l} \lesssim h\left|\underline{\psi}\right|_{2,\Omega^l} \leq h\left|\underline{\psi}\right|_2 , \tag{3.48}$$

since $\underline{\psi} \in H^2(\Omega)$.

For the latter term, it holds

$$\left|\underline{\psi} - u_h\right|_{1,\Omega^l} = |u - u_h|_{1,\Omega^l} \leq \|u - u_h\|_h \,.$$

Repeating the argument for $\sum_{K\in\mathcal{T}\setminus\mathcal{T}^+}\int_{K\cap\Omega^u} w\,(\Pi_h u - u_h)\,\mathrm{d}x$ and inserting the interpolation estimates for the first term in (3.46), we arrive at

$$\begin{aligned}\|u - u_h\|_h \leq{}& \left(1 + \frac{C_b}{C_s}\right)|u|_1 h \\ &+ \left(\frac{2}{C_s}C\|w\|_0|(|\underline{\psi} - u|_2 + |\overline{\psi} - u|_2 + |\underline{\psi}|_2 + |\overline{\psi}|_2\right)^{1/2} h \\ &+ 2\left(\frac{2}{C_s}\|w\|_0\|u - u_h\|_h\, h\right)^{1/2} .\end{aligned}$$

Using Young's inequality on the last term finishes the proof. □

In Theorem 3.44, we established convergence of the DG discretization with at least linear order. It extends the result from [110] in the sense that arbitrary polynomial degrees are allowed. In particular, the polynomial degree influences the construction of admissible sets $\mathcal{K}_{hp}$.

We now establish an analog result for basis functions of higher order, that is, at least piecewise quadratic.

Theorem 3.45. *Assume $\underline{\psi}, \overline{\psi} \in H^3(\Omega) \cap W^{2,\infty}(\Omega)$. Let $u \in W^{2+\frac{1}{p}-\varepsilon,p}(\Omega)$ (for all $1 < p < \infty$ and $\varepsilon > 0$) be the solution to Problem 2.4 and u_h be the solution to the discrete Problem 3.37. Moreover, assume that in the definition of $V_{\mathcal{T}}^p$, we have $p(K) \geq 2 \quad \forall K \in \mathcal{T}$ in (3.4). Then, we have*

$$\|u - u_h\|_h \leq Ch^{1.5-\varepsilon}. \tag{3.49}$$

Proof. At first, we can repeat the arguments made in the proof of Theorem 3.44, arriving at (3.46), namely

$$\|u - u_h\|_h \leq \left(1 + \frac{C_b}{C_s}\right)\|\Pi_h u - u\|_h + \left(\frac{2}{C_s}\sum_{K\in\mathcal{T}}\int_K -(f - Lu)(\Pi_h u - u_h)\,\mathrm{d}x\right)^{1/2} .$$

Since by assumption $u \in H^{2.5-\varepsilon}$ and we are using piecewise polynomials of order at least 2, we have

$$\|\Pi_h u - u\|_h \lesssim h^{1.5-\varepsilon}|u|_{2.5-\varepsilon}.$$

Again, it remains to estimate the product $\big(-(f-Lu), \Pi_h u - u_h\big)_0$. Following the further steps of the previous proof, in particular splitting the estimate into integrals on Ω^l and Ω^r, again, we arrive once more at (3.47),

$$\begin{aligned}\int_K w_l\,(\Pi_h u - u_h)\,\mathrm{d}x &= \int_K w_l\left(\Pi_h u - u + \underline{\psi} - \Pi_h\underline{\psi}\right)\mathrm{d}x \\ &\quad + \int_K w_l\left(u - \underline{\psi}\right)\mathrm{d}x + \int_K w_l\left(\Pi_h\underline{\psi} - u_h\right)\mathrm{d}x \\ &= \underbrace{\int_K w_l\left(\Pi_h u - u + \underline{\psi} - \Pi_h\underline{\psi}\right)\mathrm{d}x}_{=:T_3} + \underbrace{\int_K w_l\left(\Pi_h\underline{\psi} - u_h\right)\mathrm{d}x}_{=:T_4}.\end{aligned} \tag{3.50}$$

The first integral on the right hand side, T_3, again vanishes on $\mathcal{T}^l$ and on $\mathcal{T}^u$. In contrast to the previous proof, however, we have to employ an additional argument here to retrieve the desired order. For the remaining elements, we will use the following result from [112]:

$$\|\Pi_h v - v\|_{L^1(\Omega)} \le C h^{3-\varepsilon}\|v\|_{W^{s^*,p^*}(\Omega)} \tag{3.51}$$

for some values of s^* and p^* depending on ε.

Since by assumption $\underline{\psi} \in W^{2,\infty}$, we have $w_l \in L^\infty(\Omega)$. By applying Hölder's inequality and using (3.51), we get

$$\begin{aligned}\sum_{K\in\mathcal{T}^b} T_3 &= \sum_{K\in\mathcal{T}^b}\int_K w_l\left(\Pi_h u - u + \underline{\psi} - \Pi_h\underline{\psi}\right)\mathrm{d}x \\ &= \sum_{K\in\mathcal{T}^b}\int_K w_l\left(\underline{\psi} - u - \Pi_h(\underline{\psi} - u)\right)\mathrm{d}x \\ &= \left(w_l, \underline{\psi} - u - \Pi_h(\underline{\psi} - u)\right)_{\Omega^l} \\ &\le \|w_l\|_{L^\infty(\Omega)}\left\|\underline{\psi} - u - \Pi_h(\underline{\psi} - u)\right\|_{L^1(\Omega)} \\ &\le C\|w_l\|_{L^\infty(\Omega)}\, h^{3-\varepsilon}\left\|\underline{\psi} - u\right\|_{W^{s^*,p^*}(\Omega)}.\end{aligned}$$

It remains to estimate the sum over the second integral from (3.50), i. e.

$$\sum_{K\in\mathcal{T}\setminus\mathcal{T}^+} T_4 = \sum_{K\in\mathcal{T}\setminus\mathcal{T}^+}\int_K w_l\,(\Pi_h u - u_h)\,.$$

Following the proof of Theorem 3.44, we have

$$T_4 = \int_K w_l\,(\Pi_h u - u_h) \le \int_K w_l R_K^0(\Pi_h\underline{\psi} - u_h)\,\mathrm{d}x.$$

Since $P_K^0 w_l$ is a constant function and $\int_K R_K^0 g\,\mathrm{d}x = 0$ for any function g, we have

$$P_K^0 w_l R_K^0(\Pi_h\underline{\psi} - u_h) = 0,$$

and hence

$$\begin{aligned}
T_4 &= \int_K w_l \left(\Pi_h u - u_h\right) \\
&\leq \int_K w_l R_K^0 (\Pi_h \underline{\psi} - u_h) \, \mathrm{d}x \\
&= \int_K w_l R_K^0 (\Pi_h \underline{\psi} - u_h) \, \mathrm{d}x - \int_K P_K^0 w_l R_K^0 (\Pi_h \underline{\psi} - u_h) \, \mathrm{d}x \\
&= \int_K R_K^0 w_l R_K^0 \left(\Pi_h \underline{\psi} - u_h\right) \mathrm{d}x.
\end{aligned}$$

Note that $w_l = \max(w, 0)$ and hence $w_l \in H^{0.5-\varepsilon}(\Omega)$ just like w, see, e. g. Lemma A.2. Applying interpolation estimates for piecewise constant approximations, we get

$$\begin{aligned}
\sum_{K \in \mathcal{T} \setminus \mathcal{T}^+} T_4 &= \sum_{K \in \mathcal{T} \setminus \mathcal{T}^+} \int_K w_l \left(\Pi_h \underline{\psi} - u_h\right) \mathrm{d}x \\
&\leq \sum_{K \in \mathcal{T} \setminus \mathcal{T}^+} \int_K R_K^0 w_l R_K^0 \left(\Pi_h \underline{\psi} - u_h\right) \\
&= \left(R^0 w_l, R^0 \left(\Pi_h \underline{\psi} - u_h\right)\right)_{\Omega^l} \\
&\leq \left\| R^0 w_l \right\|_0 \left\| R^0 \left(\Pi_h \underline{\psi} - u_h\right) \right\|_{0,\Omega^l} \\
&\lesssim h \left\| R^0 w_l \right\|_0 \left| \Pi_h \underline{\psi} - u_h \right|_{1,\Omega^l} \\
&\leq h \left\| R^0 w_l \right\|_0 \left(\left| \Pi_h \underline{\psi} - \underline{\psi} \right|_{1,\Omega^l} + \left| \underline{\psi} - u_h \right|_{1,\Omega^l} \right) \\
&\leq h^{1.5-\varepsilon} |w_l|_{0.5-\varepsilon} \left(\left| \Pi_h \underline{\psi} - \underline{\psi} \right|_{1,\Omega^l} + \left| \underline{\psi} - u_h \right|_{1,\Omega^l} \right)
\end{aligned}$$

for $\varepsilon > 0$.

The first term in the last line can be estimated by

$$\left| \Pi_h \underline{\psi} - \underline{\psi} \right|_{1,\Omega^l} \lesssim h^2 \left| \underline{\psi} \right|_{3,\Omega^l} \leq h^2 \left| \underline{\psi} \right|_3 , \tag{3.52}$$

since $\underline{\psi} \in H^3(\Omega)$.

For the latter term, we have once more

$$\left| \underline{\psi} - u_h \right|_{1,\Omega^l} = |u - u_h|_{1,\Omega^l} \leq \|u - u_h\|_h .$$

Putting everything together, we have

$$
\begin{aligned}
\|u-u_h\|_h \leq{}& \left(1+\frac{C_b}{C_s}\right)|u|_{2.5-\varepsilon}h^{1.5-\varepsilon}\\
&+\left(\frac{2}{C_s}C\Big[\|w\|_{L^\infty(\Omega)}(\|\underline{\psi}-u\|_{W^{s^*,p^*}(\Omega)}+\|\overline{\psi}-u\|_{W^{s^*,p^*}(\Omega)})\right.\\
&\left.\quad+|w|_{0.5-\varepsilon}(|\underline{\psi}|_3+|\overline{\psi}|_3)\Big]\right)^{1/2}h^{1.5-\varepsilon}\\
&+2\left(\frac{2}{C_s}|w|_{0.5-\varepsilon}\|u-u_h\|_h\,h^{1.5-\varepsilon}\right)^{1/2}.
\end{aligned}
$$

Applying Young's inequality to the last term finishes the proof. □

In the following, we want to take local refinements into account to overcome the reduced convergences rate caused by the nonsmoothness near the free boundary. We will assume (e. g. through an adaptive procedure) that the mesh is locally finer in elements that contain parts of the free boundary. More precisely, we have

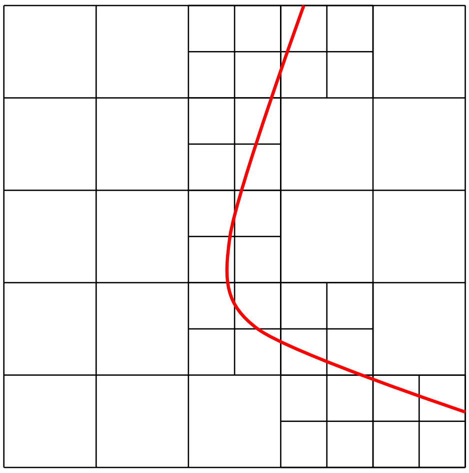
Definition 3.46. *Let $\mathcal{T}^b$ be defined as in Definition 3.40. We define the maximal element size in $\mathcal{T}^b$ as*

$$h_F=\max_{K\in\mathcal{T}^b}\operatorname{diam}(K),$$

and the maximal element size in the remaining grid as

$$h_C=\max_{K\in\mathcal{T}\setminus\mathcal{T}^b}\operatorname{diam}(K).$$

Figure 3.3.1.: Example of a grid having finer resolution near free boundary (red line).



The higher regularity of u in regions that do not interfere with the free boundary will be manifested in the following assumption.

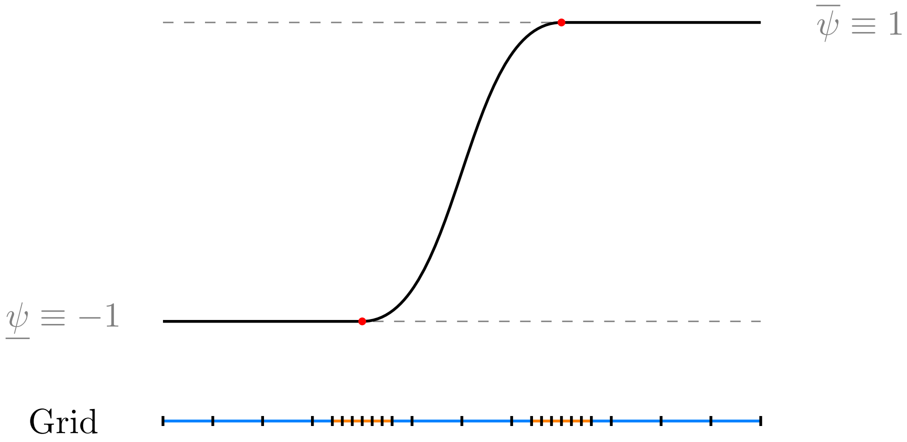
Assumption 3.47. *Let p be the minimal ansatz degree on elements in $\mathcal{T} \setminus \mathcal{T}^b$. There exists $q \in \mathbb{N}$, $1 \leq q \leq p$, such that*

$$u|_K \in H^{q+1}(K) \quad \forall K \in \mathcal{T} \setminus \mathcal{T}^b. \tag{3.53}$$

In particular, this implies the obstacle functions $\underline{\psi}$ and $\overline{\psi}$ are locally smooth enough.

Example 3.47.1. As an example of Assumption 3.47, consider the one-dimensional phase field profile in Figure 3.3.2. The areas around the free boundary (marked with red dots) use a finer grid elements, typically with lower degree (orange elements), while the remaining elements in the smoother sections have coarser grid elements with higher degrees (blue elements). While one might expect that only one or two small elements are actually touching the free boundary, in practice one will only have an approximation of the (analytical) free boundary. Having nearby elements also refined (as in the picture) is therefore a more realistic discretization. Note that in the special case of (piecewise) constant obstacles (as in this example), it is not necessary to employ high order functions in order to resolve the solution on $\mathcal{T}^l$ and $\mathcal{T}^u$ sufficiently.

Figure 3.3.2.: Phase Field Profile with Discretization



We can now state the following lemma that shows how to incorporate the different element sizes from Definition 3.46 into the interpolation estimate.

Lemma 3.48. *Let $v \in H^2(\Omega)$ be such that the smoothness Assumption 3.47 holds.*

$$\|v - \Pi_h v\|_h \leq C h_C^q |v|_{q+1,\mathcal{T}\setminus\mathcal{T}^b} + \tilde{C} h_F |v|_{1,\mathcal{T}^b}, \tag{3.54}$$

where the constant depends on the grid, the penalty parameter and the chosen discrete space.

Proof. Let $e \in \Gamma$ be a face of an element K. From equation 2.4 and 2.5 in [11], we have

$$\begin{aligned}\|v\|_{0,e}^2 &\leq C\left(|e|^{-1}\|v\|_{0,K}^2 + |e||v|_{1,K}^2\right), \quad v \in H^1(K),\\ \left\|\frac{\partial v}{\partial \mathbf{n}}\right\|_{0,e}^2 &\leq C\left(|e|^{-1}|v|_{1,K}^2 + |e||v|_{2,K}^2\right), \quad v \in H^2(K).\end{aligned}$$

Using this and the Cauchy–Schwarz inequality, we get

$$\begin{aligned}\|v - \Pi_h v\|_h^2 = &\sum_{K\in\mathcal{T}} |v - \Pi_h v|_{1,K}^2 + \sum_{e\in\Gamma} \sigma^{-1}|e|\| \left\{\nabla(v - \Pi_h v)\right\} \|_{0,e}^2\\ &+ \sigma|e|^{-1}\|[\![v - \Pi_h v]\!]\|_{0,e}^2\\ \leq C &\left(\sum_{K\in\mathcal{T}} |v - \Pi_h v|_{1,K}^2 + |h_K||v - \Pi_h v|_{1,K}^2 + |h_K|^2|v - \Pi_h v|_{2,K}^2\right).\end{aligned}$$

Splitting the sum over the elements between the nonsmooth ($\mathcal{T}^b$) and the smooth parts and applying the local interpolation estimate (3.29) yields the result. □

The following assumption can be verified without knowing the solution since only the obstacle functions and the discretization are involved.

Assumption 3.49. *We assume that in every element $K_u \in \mathcal{T}^u$ and every basis function $\varphi_i^{K_u}$ defined on K_u, we have*

$$\int_{K_u} \left(f - L\overline{\psi}\right)\varphi_i^{K_u}\,\mathrm{d}x \geq 0. \tag{3.55}$$

Analogously, we assume for $K_l \in \mathcal{T}^l$ it holds for every basis function

$$\int_{K_l} \left(f - L\underline{\psi}\right)\varphi_i^{K_l}\,\mathrm{d}x \leq 0. \tag{3.56}$$

Clearly, for piecewise linear or other non-negative basis functions, the conditions are always fulfilled. In Appendix A.1, we provide a sufficient (yet not necessary) condition for Assumption 3.49 to be true.

Equipped with these results and assumptions, we can state a modified version of Theorem 3.44 that takes local refinements into account.

Theorem 3.50. *Assume $\underline{\psi}, \overline{\psi} \in H^2(\Omega)$. Let $u \in H^2(\Omega)$ be the solution to Problem 2.4 and u_h be the solution to the discrete problem 3.37. Assume u is locally smooth in the sense that Assumption 3.47 holds. Moreover, we require Assumption 3.49 to be true. Then, we have*

$$\|u - u_h\|_h \leq C_0 h_C^q + C_1 h_F. \tag{3.57}$$

Proof. As expected, the proof is very similar to the proofs of Theorem 3.44 and Theorem 3.45.

As before, we deduce (3.46), i. e.

$$\|u - u_h\|_h \leq \left(1 + \frac{C_b}{C_s}\right) \|\Pi_h u - u\|_h + \left(\frac{2}{C_s} \sum_{K \in \mathcal{T}} \int_K -(f - Lu)(\Pi_h u - u_h)\,\mathrm{d}x \right)^{1/2} .$$

By applying Lemma 3.48, we know the first term on the right hand side can be estimated by

$$\left(1 + \frac{C_b}{C_s}\right) \|\Pi_h u - u\|_h \leq \left(1 + \frac{C_b}{C_s}\right) \left(C h_C^q |v|_q + \tilde{C} h_F |u|_1 \right) .$$

Therefore we are once more left with estimating the square root of

$$\sum_{K \in \mathcal{T}} \int_K \underbrace{-(f - Lu)}_{=:w}(\Pi_h u - u_h)\,\mathrm{d}x = \sum_{K \in \mathcal{T}} \int_{K \cap \Omega^l} w(\Pi_h u - u_h)\,\mathrm{d}x + \int_{K \cap \Omega^u} w(\Pi_h u - u_h)\,\mathrm{d}x$$

as argued before. Note that all integrals vanish for $K \in \mathcal{T}^+$ as $w \equiv 0$ there.

We will again consider the regions Ω^l and Ω^u separately and use the function w_l which is equal to w in Ω^l and 0 outside. This gives, again, (3.47), namely

$$\begin{aligned}
\int_{K \cap \Omega^l} w\,(\Pi_h u - u_h)\,\mathrm{d}x &= \int_K w_l\,(\Pi_h u - u_h)\,\mathrm{d}x \\
&= \int_K w_l \left(\Pi_h u - u + \underline{\psi} - \Pi_h \underline{\psi} \right) \mathrm{d}x \\
&\quad + \int_K w_l \left(u - \underline{\psi} \right) \mathrm{d}x + \int_K w_l \left(\Pi_h \underline{\psi} - u_h \right) \mathrm{d}x \\
&= \underbrace{\int_K w_l \left(\Pi_h u - u + \underline{\psi} - \Pi_h \underline{\psi} \right) \mathrm{d}x}_{=:T_3} + \underbrace{\int_K w_l \left(\Pi_h \underline{\psi} - u_h \right) \mathrm{d}x}_{=:T_4} .
\end{aligned} \tag{3.58}$$

The first integral of (3.58), T_3, vanishes on $\mathcal{T}^l$ and on $\mathcal{T}^u$. The remaining elements are from $\mathcal{T}^b$ and thus have maximal diameter h_F. Hence, we have

$$\begin{aligned}
\sum_{K \in \mathcal{T}^b} T_3 &= \sum_{K \in \mathcal{T}^b} \int_K w_l \left(\Pi_h u - u + \underline{\psi} - \Pi_h \underline{\psi} \right) \mathrm{d}x \\
&= \sum_{K \in \mathcal{T}^b} \int_K w_l \left(\underline{\psi} - u - \Pi_h(\underline{\psi} - u) \right) \mathrm{d}x \\
&= \left(w_l, \underline{\psi} - u - \Pi_h(\underline{\psi} - u) \right)_{\Omega^l \cap \mathcal{T}^b} \\
&\leq \|w_l\|_0 \left\| \underline{\psi} - u - \Pi_h(\underline{\psi} - u) \right\|_{0, \Omega^l \cap \mathcal{T}^b} \\
&\leq C \|w_l\|_0 \, h_F^2 \left| \underline{\psi} - u \right|_2 .
\end{aligned}$$

The latter integral of (3.58), $T_4 = \int_K w_l \left(\Pi_h \underline{\psi} - u_h\right) \mathrm{d}x$, again vanishes on $\mathcal{T}^u$ by definition of w_l and is nonpositive on $\mathcal{T}^l$ by Assumption 3.49. Thus, we only have to consider elements which are in $\mathcal{T}^b$. Repeating the arguments of the proof of Theorem 3.44 verbatim (only replacing h with h_F), we get

$$\sum_{K \in \mathcal{T}^b} T_4 = \sum_{K \in \mathcal{T}^b} \int_K w_l \left(\Pi_h \underline{\psi} - u_h\right) \mathrm{d}x \le C h_F \|w_l\|_0 \left(|\Pi_h \underline{\psi} - \underline{\psi}|_{1,\Omega^l} + |\underline{\psi} - u_h|_{1,\Omega^l}\right).$$

As seen before, the first term in the last line can be estimated by

$$\left|\Pi_h \underline{\psi} - \underline{\psi}\right|_{1,\Omega^l} \lesssim h_F \left|\underline{\psi}\right|_{2,\Omega^l},$$

since $\underline{\psi} \in H^2(\Omega)$.

For the latter term, it holds

$$\left|\underline{\psi} - u_h\right|_{1,\Omega^l} = |u - u_h|_{1,\Omega^l} \le \|u - u_h\|_h .$$

Repeating the arguments starting from (3.58) for Ω^u, i.e. estimating $\int_{K \cap \Omega^u} w \left(\Pi_h u - u_h\right) \mathrm{d}x$ along the same lines, we finally get

$$\begin{aligned} \|u - u_h\|_h \le & C\left(1 + \frac{C_b}{C_s}\right) |u|_1 h_C^q \\ & + \left((1 + \frac{C_b}{C_s})\tilde{C} + \frac{2}{C_s}\overline{C}\|w\|_0((|\underline{\psi} - u|_2 + |\overline{\psi} - u|_2 + |\underline{\psi}|_2 + |\overline{\psi}|_2\right)^{1/2} h_F \\ & + 2\left(\frac{2}{C_s}\|w\|_0 \|u - u_h\|_h h_F\right)^{1/2} . \end{aligned}$$

Applying Young's inequality on the last term, we arrive at the conclusion. □

Remark 3.51. For DG spaces that employ at least piecewise quadratic basis functions and problems satisfying the assumptions of Theorem 3.45, we can easily extend Theorem 3.50 to

$$\|u - u_h\|_h \le C h_C^q + \tilde{C} h_F^{1.5-\varepsilon}.$$

3.4. Discretization of Variational Inequalities of the Second Kind

While we have not developed much theory for this case, we will briefly suggest a discretization scheme for variational inequalities of the second kind. In [111], a priori error estimates for a discontinuous Galerkin discretization of a simplified friction model, are proved. We will introduce a discretization of a more general class of variational inequalities of the second kind which is a special case of Problem 2.11.

Problem 3.52. *For $H = H^1(\Omega)$, let $a(\cdot,\cdot)$ be a symmetric bilinear form, $l \in H'$ a bounded linear functional and Φ convex and chosen such that the map $v \mapsto \int_\Omega \Phi(v)\,\mathrm{d}x$ is lower semicontinuous and proper.*

Find $u \in H$ such that

$$a(u, v-u) - l(v-u) + \int_\Omega \Phi(v(x))\,\mathrm{d}x - \int_\Omega \Phi(u(x))\,\mathrm{d}x \geq 0 \quad \forall v \in H. \tag{3.59}$$

Equivalently (see e. g. [57]), we want to minimize the energy such that

$$\mathcal{J}(u) \leq \mathcal{J}(v) \quad \forall v \in H,$$

with $\mathcal{J}(v) = \frac{1}{2}a(v,v) - l(v) + \int_\Omega \Phi(v(x))\,\mathrm{d}x$.

We stress once more that Problem 3.52 is just Problem 2.11 with $j(\cdot) = \int_\Omega \Phi(\cdot)\,\mathrm{d}x$.

While the discretization of the quadratic part of the functional, namely $\mathcal{J}_0(v) = a(v,v) - l(v)$ with SIPG is straightforward (see the preceding sections), we need to take special care of the nonlinear (and possibly nonsmooth) part $\int_\Omega \Phi(v(x))\,\mathrm{d}x$. Computing and optimizing the exact integral would be challenging or even impossible.

The idea, which has been successfully employed for continuous finite element spaces before (see e. g. [57]) is to replace integrals in the nonlinearity by appropriate quadrature rules (also known as *mass lumping* [61]). Since we assumed before that the Lagrange points of basis functions are distributed such that a reasonable quadrature rule emerges (e. g. Gauss–Lobatto points), see Assumption 3.34, we approximate the integral in the following way:

Let $\{\varphi_i\}_i$ be the set of (discontinuous) nodal basis functions, $w_i = \int_\Omega \varphi_i(x)\,\mathrm{d}x$, and let $x_i \in \overline{\Omega}$ be the Lagrange node corresponding to the i-th basis function. Then, we replace $\int_\Omega \Phi(v(x))\,\mathrm{d}x$ by the approximate integral:

$$\int_\Omega \Phi(v(x))\,\mathrm{d}x \approx \sum_i w_i \Phi(v(x_i)).$$

Indeed, having negative weights ω_i would make us lose the convexity of the nonlinearity. Note that for a finite element function $v \in V_\mathcal{T}^p$, the point values $v(x_i)$ are readily available as the coefficients in the Lagrange basis. Finally, we arrive at a discrete version of Problem 3.52:

Problem 3.53. *Let $n = \dim(V_\mathcal{T}^p)$. Find $u_h \in V_\mathcal{T}^p$ such that*

$$a_h\left(u_h, v-u_h\right) - F_h(v-u_h) + \sum_{i=1}^n w_i \left(\Phi(v(x_i)) - \Phi(u(x_i))\right) \geq 0 \quad \forall v \in V_\mathcal{T}^p, \tag{3.60}$$

or, equivalently,

$$\begin{aligned}\mathcal{J}_h(u_h) &\leq \mathcal{J}_h(v) \quad \forall v \in V_\mathcal{T}^p,\\ \mathcal{J}_h(v) &= \frac{1}{2}a_h\left(v,v\right) - F_h(v) + \sum_{i=1}^n w_i \Phi(v(x_i)).\end{aligned}$$

Since both the mappings $v \mapsto \int_\Omega \Phi(v(x))\,\mathrm{d}x$ for $v \in H^1(\Omega)$ and

$$v_h \mapsto \sum_{i=1}^{n} w_i \Phi(v_h(x_i))$$

for $v_h \in V_\mathcal{T}^p$ are convex, lower semicontinuous and proper, we have that Problem 3.52 and Problem 3.53 have unique solutions in H and $V_\mathcal{T}^p$, respectively.

All of the algorithms and heuristics in the following chapters can be applied to this class of variational inequalities, too.

4. Algebraic Solution

The problems discussed in the previous chapters have a common theme. They are not only not linear but in general not even smooth. Thus, standard techniques for solving the arising algebraic problems such as for example a classical Newton method are not applicable since second order derivatives of energy functionals may not be available globally. Therefore other, more specialized optimization algorithms, in particular stemming from the area of nonsmooth optimization, have to be applied.

Much research has been performed to find methods which are particularly efficient for optimization problems arising from the finite element discretization of variational inequalities and other related problems. In particular, these problems often exhibit a very large number of unknowns. This implies that efficient algorithms with optimal complexity are needed. While -for linear problems- several well established methods such as e. g. multigrid methods exist, the nonlinear case we are concerned with is more involved. Many attempts have been made to construct solvers that are both efficient and convergent. Examples of these are *constraint decomposition method* by Tai [107], FAS based approaches due to Brandt [33], the *projected multilevel relaxation* and *standard monotone multigrid methods* by Mandel [83], or *truncated monotone multigrid methods* by Kornhuber [75]. Another popular algorithm is the *Primal-Dual Active Set Strategy*, see, e. g. [69]. Many of these algorithms can be understood as multigrid algorithms as summarized in [64]. In particular for obstacle problems, they may require the construction of coarse grid obstacles.

A particularly elegant algebraic solver that does not need coarse grid obstacle functions is the Truncated Nonsmooth Newton Multigrid (TNNMG) method. This method has been first introduced for obstacle problems in [64] and analyzed in [61]. Since then, it has been applied to various problems in material science (e. g. [62]), mechanics (e. g. [92]), etc. Part of the method's beauty stems from its simplicity, manifested in the fact that many of its components, such as nonlinear smoothers and algebraic solvers for linear systems, are readily available in many finite element codes, yet the method converges rapidly for many problems. Despite being a solution procedure for nonlinear problems, in general only very small nonlinear (sub-)problems have to be solved. Much of the heavy lifting can be delegated to an approximate linear solver. There, a single step of a multigrid method (if available) is the canonical choice. For reasonable initial iterates, the convergence rate of the method is asymptotically the rate of the linear solver for many problems [67]. In the following section, we will present the TNNMG method in more detail and discuss how to apply it to the discontinuous Galerkin discretizations we consider.

4.1. Truncated Nonsmooth Newton Multigrid

The TNNMG method has been introduced as a nonsmooth minimization algorithm for block-separable convex problems. In particular, the problem at hand may arise from a different source than the discretization of a differential equation [67]. Therefore we may frequently identify a given finite dimensional space V with $\mathbb{R}^{\dim(V)}$ by a suitable isomorphism. For example, on a given element $K \subset \Omega \subset \mathbb{R}^d$, we can identify a finite element function $v \in \mathsf{P}^k(K)$, $v = \sum_i v_i \phi_i$ by its coefficient vector $(v_i)_i \subset \mathbb{R}^{\dim\left(\mathsf{P}^k(K)\right)}$.

A detailed analysis of the TNNMG method can be found in [61]. Several further additions and insights culminated in the general framework presented in [67], see also the references therein. There, it was also shown that the requirements on the convexity of the problem can be weakened.

The general idea of the method is to take a method that is proven to converge to a stationary point by monotonically reducing energy. A typical example would be a successive energy minimization for smaller sub-problems, i. e. a nonlinear Gauss–Seidel method. While these are known to converge theoretically, they exhibit very poor performance for growing numbers of unknowns. Thus, the TNNMG method accelerates their convergence by computing another search direction that will be damped by a line search to guarantee energy descent and therefore convergence of the method. The additional search direction will be determined by applying a Newton method in an appropriate subspace and using a suitable projection afterwards.

In the following, we will present a short introduction to the TNNMG method. For more details, we again refer to [67] and the references therein. We will also loosely follow their notation.

Problem 4.1. *Consider a functional $\mathcal{J} \colon \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, with $\mathcal{J}$ being proper, coercive, lower semicontinuous and continuous in its domain*

$$\operatorname{dom} \mathcal{J} = \left\{ x \in \mathbb{R}^n : \mathcal{J}(x) < \infty \right\}.$$

Moreover, let $\operatorname{dom} \mathcal{J}$ be convex. Then, we seek a minimizer $u^* \in \mathbb{R}^n$ such that

$$\mathcal{J}(u^*) \leq \mathcal{J}(v) \quad \forall v \in \mathbb{R}^n. \tag{4.1}$$

A crucial assumption for the TNNMG method on the energy $\mathcal{J}$ is that it exhibits a certain splitting into a rather well-behaved functional $\mathcal{J}_0$ and a block-separable nonlinearity φ:

Assume there is a disjoint partition of $\{1, \dots, n\}$ into M non-empty index sets N_i, defining subspaces $V_i \simeq \mathbb{R}^{|N^i|}$. Then, we have $\mathbb{R}^n = V_1 \oplus \cdots \oplus V_M$. Let $v \in \mathbb{R}^n$. For each index set $N^i = \left\{ N^i_1, \dots, N^i_{n_i} \right\} \subset \{1, \dots, n\}$, we denote the canonical restriction of v to V_i by

$$v_i = \begin{pmatrix} v_{N^i_1} \\ \vdots \\ v_{N^i_{n_i}} \end{pmatrix}.$$

Given this notation, we can state our assumption on $\mathcal{J}$:

Assumption 4.2. *Assume $\mathcal{J}$ has the following form:*

$$\mathcal{J} = \mathcal{J}_0 + \varphi. \tag{4.2}$$

Here, $\mathcal{J}_0$ is assumed to be coercive and continuously differentiable. For the other part, assume φ is such that

$$\varphi(v) = \sum_{i=1}^{M} \varphi_i(v_i), \tag{4.3}$$

with each φ_i being convex, proper, lower semicontinuous and continuous on its domain.

Example 4.2.1. For the discretized obstacle problem from Chapter 3, we can split the nonlinearity such that each "block" is scalar, i. e. $N^i = \{i\}$ (and therefore $M = n$) and the functional can be decomposed into

$$\begin{aligned} \mathcal{J}_0(v) &= \frac{1}{2} a_h(v, v) - F_h(v) \\ \varphi_i(v_i) &= \chi_{\left[\underline{\psi}_i, \overline{\psi}_i\right]}(v_i). \end{aligned}$$

Since the discretization scheme was constructed such that $a_h(\cdot, \cdot)$ leads to a positive definite matrix, we have that $\mathcal{J}_0$ is a coercive, smooth functional. Moreover, it can easily be seen that indicator functionals of the kind

$$\chi_A(x) = \begin{cases} 0 & \text{if } x \in A, \\ \infty & \text{else} \end{cases}$$

fulfill the requirements on φ_i and thus the discretized obstacle problem can be treated in this framework.

4.1.1. Nonlinear Smoothing

Problems of type (4.2) can be solved e. g. by iterative schemes such as successive subspace minimization methods (also known as relaxation or nonlinear (Block-)Gauss–Seidel methods), see e. g. [58].

To this end, we define the local minimization operators $\mathcal{M}_i : \operatorname{dom} \mathcal{J} \to \operatorname{dom} \mathcal{J}$, $i = 1, \ldots, M$, by

$$\mathcal{M}_i(\cdot) = (\cdot) + \operatorname*{arg\,min}_{v \in V_i} \mathcal{J}(\cdot + v). \tag{4.4}$$

Composition of these gives the nonlinear Gauss–Seidel operator

$$\mathcal{M} = \mathcal{M}_M \circ \cdots \circ \mathcal{M}_1. \tag{4.5}$$

The application of this operator,

$$x^{\nu+1} = \mathcal{M}(x^\nu), \tag{4.6}$$

can also be written in algorithmic form, see Algorithm 1.

Algorithm 1 Nonlinear Gauss–Seidel method

```
procedure NonlinearGS($x^\nu$)
    Set $w^0 = x^\nu$
    for $i = 1, \dots, M$ do
        $w^i = w^{i-1} + \arg\min_{v \in V_i} \mathcal{J}(w^{i-1} + v)$
    end for
    $x^{\nu+1} \leftarrow w^M$
end procedure
```

Remark 4.3. Since typically the size of the subspaces V_i is small (or even scalar) and fixed, the sub-problems (4.4) can often be solved exactly in reasonable (i. e. $\mathcal{O}(1)$) time. In other works, particularly [61, 67], also inexact local minimization is considered.

It is well known that the algorithm (4.6) converges if reasonable assumptions on $\mathcal{J}$ are made, for instance $\mathcal{J}$ being strictly convex, see e. g. [58]. As mentioned before, however, this algorithm tends to exhibit very poor convergence rates with a growing number of unknowns. Therefore we will introduce additional elements to the algorithm which will be motivated in the following.

4.1.2. Abstract TNNMG Algorithm

By assuming the block-separable structure of $\mathcal{J}$ and applying exact minimization in the sub-problems (4.4), we directly obtain continuity of $\mathcal{J} \circ \mathcal{M}$, see [67, Lemma 5.1]. Moreover, we clearly have that each local minimization does not increase the energy of the current iterate:

$$\mathcal{J}(\mathcal{M}_i(v)) \leq \mathcal{J}(v) \quad \forall v \in \mathbb{R}^n.$$

This directly implies that the Gauss–Seidel iteration (4.6) is also monotone:

$$\mathcal{J}(\mathcal{M}(v)) \leq \mathcal{J}(v) \quad \forall v \in \mathbb{R}^n. \tag{4.7}$$

This monotonicity property is both central to the convergence proof of the TNNMG method and motivates its construction:

The central idea is that if the algorithm converges by decreasing energy, performing another correction that does not increase energy should not harm the theoretical convergence property. This is expressed in Algorithm 2. We can now state a conver-

Algorithm 2 Abstract TNNMG algorithm

```
Require: $x^\nu \in \operatorname{dom} \mathcal{J}$
procedure AbstractTNNMG($x^\nu$)
    $x^{\nu+1/2} \leftarrow \mathcal{M}(x^\nu)$                          ▷ Perform nonlinear Gauss–Seidel step
    Compute $\mathcal{C}(x^{\nu+1/2})$, s.t. $\mathcal{J}(x^{\nu+1/2} + \mathcal{C}(x^{\nu+1/2})) \leq \mathcal{J}(x^{\nu+1/2})$
    $x^{\nu+1} \leftarrow x^{\nu+1/2} + \mathcal{C}(x^{\nu+1/2})$                ▷ Add linear correction
end procedure
```

gence result (which is a special case of Theorem 4.1 in [67]) for the abstract TNNMG algorithm.

Theorem 4.4. *Let $(x^\nu)_{\nu=1}^\infty$ be generated by the abstract TNNMG Algorithm 2. Additionally to the formerly stated assumptions, assume that the local minimization problem*

$$\arg\min_{v \in V_i} \mathcal{J}(w+v)$$

has a unique solution for all $w \in \operatorname{dom} \mathcal{J}$ and $i \in \{1, \dots, M\}$.

Then, any accumulation point x of $(x^\nu)_\nu$ is stationary in the sense that

$$\mathcal{J}(x) \leq \mathcal{J}(x+v) \qquad \forall v \in V_i, \quad i = 1, \dots, M.$$

Proof. Apply [67, Theorem 4.1], noting that $\mathcal{J}(\cdot)$, $\mathcal{M}(\cdot)$ and $\mathcal{C}(\cdot)$ comply with the assumptions of that theorem by construction. □

More specifically, one can show that if the problem has a unique solution, the abstract TNNMG algorithm will converge to this minimizer.

Corollary 4.5. *Under the assumptions of Theorem 4.4, if $\mathcal{J}$ possesses a unique minimizer u^*, we have*

$$x^\nu \to u^*. \tag{4.8}$$

Proof. Apply [67, Corollary 4.4], noting that by Assumption 4.2, $\mathcal{J}$ is block-separable nonsmooth. □

4.1.3. Linear Correction

We are left with the question how to construct a suitable search direction $\mathcal{C}(x^{\nu+1/2})$. If $\mathcal{J}$ were smooth, one could apply a Newton-step of the form

$$\mathcal{C}(x^{\nu+1/2}) = -\left(\mathcal{J}''\left(x^{\nu+1/2}\right)\right)^{-1} \mathcal{J}'(x^{\nu+1/2}).$$

As it is well known from optimization theory, this would lead to rapid convergence if $x^{\nu+1/2}$ is close to a stationary point x and if $\mathcal{J}$ is smooth enough. This would render the TNNMG method as a Newton method in terms of the nonlinear preconditioner.

To account for the possible nonsmoothness of $\mathcal{J}$, we have to construct iteration dependent subspaces W_ν such that

$$v \in W_\nu \mapsto \mathcal{J}(x^{\nu+1/2} + v)$$

is C^2 near the origin. For a good search direction it is in general desirable to have W_ν to be as large as possible. Finding such a space, however, might be a challenging task. In [67], several methods for constructing the subspaces W_ν are presented for different applications.

Example 4.5.1. Consider again the quadratic obstacle problem from Example 4.2.1. There, we have that $\mathcal{J}$ is not differentiable in direction e_i exactly if the function touches an obstacle in the i-th node. Thus, we can define W_ν to be the subset of nodes that are inactive, i. e. bounded away from the obstacles:

$$\begin{aligned} W_\nu &= \mathbb{R}^{\mathcal{I}_\nu}, \\ \mathcal{I}_\nu &= \left\{ i \in \{1, \dots, n\} : x_i^{\nu+1/2} \in \left(\underline{\psi}_i, \overline{\psi}_i \right) \right\}. \end{aligned}$$

Assume we have found a suitable space W_ν. Then, we compute an approximate Newton-correction on that space that serves as a candidate for $\mathcal{C}(x^{\nu+1/2})$:

$$v_\nu \approx - \left(\mathcal{J}'' \left(x^{\nu+1/2} \right) |_{W_\nu \times W_\nu} \right)^{-1} \mathcal{J}' \left(x^{\nu+1/2} \right) |_{W_\nu}. \tag{4.9}$$

In implementations, the restriction to W_ν is implemented by setting the corresponding rows and columns or vector entries to zero, thus *truncating* the matrix and right hand side. Strictly speaking, the system (4.9) is not well-posed. As a remedy, one might formally introduce the Moore–Penrose pseudoinverse for $\mathcal{J}''(\cdot)|_{W_\nu \times W_\nu}$. For $\mathcal{J}$ convex, the system (4.9) then indeed possesses a unique solution, see [67].

In practice it is often sufficient to perform a single step of a suitable iterative solver for the linear problems to obtain a reasonable approximation in (4.9). The canonical choice would be a *multigrid* method, if available. For details how to construct a multigrid method for our *hp*-DG discretized system, consult Section 4.2.

As the correction v_ν is agnostic of any nonlinearities of $\mathcal{J}$, we cannot use it directly since we cannot guarantee the crucial error descent $\mathcal{J}(x^{\nu+1/2} + v_\nu) \leq \mathcal{J}(x^{\nu+1/2})$. To overcome this issue, a damping parameter $\hat{\rho}$ can be introduced that can be computed for example by a simple line search:

$$\text{Find } \hat{\rho} \in \mathbb{R} \text{ such that } \mathcal{J}(x^{\nu+1/2} + \hat{\rho} v_\nu) \leq \mathcal{J}(x^{\nu+1/2}).$$

Thus, ρv_ν could be used as the linear correction $\mathcal{C}(x^{\nu+1/2})$ in the abstract TNNMG algorithm. In practice, however, it turned out that often v_ν is so close to the boundary of $\operatorname{dom} \mathcal{J}$ that only very small damping parameters are possible, thus slowing down the method significantly. Therefore, before applying the line search to guarantee energy descent, a Euclidean projection to $\mathcal{J}$'s domain is performed.

Example 4.5.2. Consider the situation as illustrated in Figure 4.1.1. As $x^{\nu+1/2}$ is close to the boundary of $\mathcal{J}$'s domain, only a short increment can be achieved by optimizing along the direction v_ν (the red line). If the projection is performed, however, the correction can be done along a greater length (the blue line). However, this might come at the price of a worse descent direction.

This leaves us with

$$\mathcal{C}(x^{\nu+1/2}) = \rho \, P_{\operatorname{dom}(\mathcal{J})}(x^{\nu+1/2} + v_\nu), \tag{4.10}$$

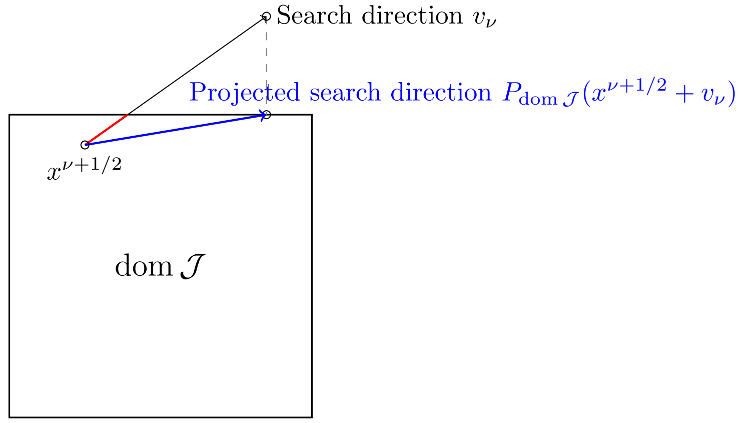

Search direction v_ν
Projected search direction $P_{\mathrm{dom}\,\mathcal{J}}(x^{\nu+1/2}+v_\nu)$
$x^{\nu+1/2}$
dom $\mathcal{J}$

Figure 4.1.1.: Projected search direction (blue) allows for more optimization than the nonprojected direction (red)

where $P_{\mathrm{dom}(\mathcal{J})} : \mathbb{R}^n \to \mathrm{dom}(\mathcal{J})$ is the Euclidean projection into $\mathrm{dom}(\mathcal{J})$ and ρ is chosen such that

$$\mathcal{J}\left(x^{\nu+1/2} + \mathcal{C}(x^{\nu+1/2})\right) \leq \mathcal{J}\left(x^{\nu+1/2}\right).$$

Thus, we clearly have that the particular choice of the linear correction (4.10) fulfills the assumption of Theorem 4.4 and we have that the TNNMG Algorithm 3 converges.

Algorithm 3 Full TNNMG Algorithm

```
procedure TNNMG($x^\nu$)
    $x^{\nu+1/2} \leftarrow \mathcal{M}(x^\nu)$                                    ▷ Perform nonlinear Gauss–Seidel step
    $v_\nu \approx -\left(\mathcal{J}''\left(x^{\nu+1/2}\right)|_{W_\nu \times W_\nu}\right)^{-1} \mathcal{J}'\left(x^{\nu+1/2}\right)|_{W_\nu}$        ▷ Approximately solve system (4.9)
    Compute projection into dom $\mathcal{J}$, $P_{\mathrm{dom}\,\mathcal{J}}(v_\nu)$
    Compute damping factor $\rho$ that guarantees energy is not increased
    $x^{\nu+1} \leftarrow x^{\nu+1/2} + \rho P_{\mathrm{dom}\,\mathcal{J}}(v_\nu)$
end procedure
```

4.2. Linear Multilevel Solver

As seen in Section 4.1.3, the TNNMG method requires the approximate solution of a (truncated) linear system. For iterative schemes that converge rapidly for a given problem, often a single iteration step of that scheme is sufficient to generate a reasonable search direction [61]. The canonical example would be a geometric multigrid step for a $\mathcal{P}^1$-finite-element discretization of an elliptic problem. This would require the existence of a suitable grid hierarchy. If no such hierarchy is available or the problem at hand does not respond well to geometric multigrid, any other linear solution scheme, e. g. an algebraic multigrid scheme or even direct solvers, are valid alternatives [67]. Note that in practice special care has to be taken since the truncation of the full system leads to problems that might be not invertible on the full space $\mathbb{R}^n$ but only on W_ν.

4.2.1. hp-Multigrid for Discontinuous Galerkin Discretizations

In the following we will present a multigrid strategy which is suitable for systems arising from geometric ("h–")refinements of the underlying grid or by increasing the local ansatz degree ("p–refinement"). Abusing terminology, we still speak of a multi*grid* method in the latter case.

Multigrid methods for DG discretizations (using h-, p- or hp-refinements) have been considered in several works before, see e. g. [8, 34, 59, 84]. In these articles, different kinds of multigrid methods have been presented and analyzed. We will present an approach that is most similar to the one from [8] (albeit using Gauss–Seidel type smoothers instead of a Richardson iteration). In numerical examples the same authors confirmed that using Gauss–Seidel smoothers (either blocked or scalar type) leads to better convergence rates while exhibiting less dependence on the polynomial degree used for the ansatz functions [9].

Assume there is a sequence of DG finite element spaces

$$V_{\mathcal{T}_1}^{p_1} \subset \cdots \subset V_{\mathcal{T}_J}^{p_J}.$$

A finer space $V_{\mathcal{T}_{j+1}}^{p_{j+1}}$ is obtained from $V_{\mathcal{T}_j}^{p_j}$ ($j \in \{1, \dots, J-1\}$) by local modifications:

Assumption 4.6. *For every "fine" element K in $\mathcal{T}_{j+1}$, it holds*

- *if $K \in \mathcal{T}_{j+1} \cap \mathcal{T}_j$, we have $p_{j+1}(K) \geq p_j(K)$,*
- *if $K \in \mathcal{T}_{j+1} \setminus \mathcal{T}_j$, K was obtained from a geometric refinement of an element $K' \in \mathcal{T}_j$ and we have $p_{j+1}(K) \geq p_j(K')$.*

Note that in general it is permitted that in some elements neither (or both!) types of refinement happened. For shorter notation, we denote will abbreviate $V_{\mathcal{T}_j}^{p_j}$ by V^j.

As a model problem, consider again the SIPG discretization of the (linear) Poisson problem from Chapter 3. For a given basis $\{\varphi_i\}_i$ of V^J, this induces the linear system

$$A^J x^J = b^J \tag{4.11}$$

defined by

$$\begin{aligned} A_{ij}^J &= a_h\left(\varphi_i, \varphi_j\right), \\ b_i^J &= F_h(\varphi_i). \end{aligned}$$

One key ingredient of a multigrid method is the definition of prolongation and restriction operators. Since the discrete spaces $\{V^j\}_j$ are nested, we can define the prolongation operator

$$T_j^{j+1} : V^j \to V^{j+1} \tag{4.12}$$

as the canonical injection operator. Assume that for each space V^j a suitable basis has been chosen. For these bases, the corresponding matrix to the linear map (4.12) can be obtained through interpolation of the coarse basis functions using the finer basis functions. The resulting matrix representing the map T_j^{j+1} in these bases will be called P_j^{j+1}.

The corresponding restriction

$$T_{j+1}^j : \left(V^{j+1}\right)' \to \left(V^j\right)' \tag{4.13}$$

is defined as being the adjoint of the operator T_j^{j+1}. The algebraic representation R_{j+1}^j of the restriction operator with respect to the chosen basis is the transposed prolongation matrix, i. e.

$$R_{j+1}^j = \left(P_j^{j+1}\right)^\top .$$

For nonconsecutive level pairs $(j, j+l)$, we can define restriction and prolongation in the obvious way:

$$\begin{aligned} P_j^{j+l} &= P_{j+l-1}^{j+l} \cdot \dots \cdot P_j^{j+1}, \\ R_{j+l}^j &= R_{j+1}^j \cdot \dots \cdot R_{j+l}^{j+l-1}. \end{aligned}$$

Given prolongation and restriction operators, we can define coarse-grid matrices recursively through

$$A^j = R_{j+1}^j A^{j+1} P_j^{j+1} = \left(P_j^{j+1}\right)^\top A^{j+1} P_j^{j+1}, \quad j = 1, \dots, J-1. \tag{4.14}$$

The definition of the coarse-grid matrices through the prolongation to finer levels and subsequent restriction has an unfavorable effect. Evaluation of the bilinear form induced by A^J for functions that are defined on coarser grids means that we compute the penalty terms by integrating over the **fine** faces:

Say we have $v, w \in V^j$ with $j < J$ and let $v_j, w_j \in \mathbb{R}^{\dim(V^j)}$ be their respective coefficient vectors in V^j's chosen basis. Then, we have

$$\langle A^j v_j, w_j \rangle = \dots + \sum_{e \in \Gamma_J} \frac{\sigma}{|e|} \int_e [\![v]\!][\![w]\!] \,\mathrm{d}S,$$

where Γ_j denotes the set of faces on the j-th level. Since the faces on coarser level can be understood as unions of finer faces, we have for every coarse face $e_j \in \Gamma_j$,

$$\sum_{e \in e_j \cap \Gamma_J} \int_e [\![v]\!][\![w]\!] \,\mathrm{d}S = \int_{e_j} [\![v]\!][\![w]\!] \,\mathrm{d}S.$$

However, if we have that the face e_j is actually coarser than the finest faces from Γ_J, we have

$$\sum_{e \in e_j \cap \Gamma_J} \frac{\sigma}{|e|} \int_e [\![v]\!][\![w]\!] \, \mathrm{d}S \neq \frac{\sigma}{|e_j|} \int_{e_j} [\![v]\!][\![w]\!] \, \mathrm{d}S,$$

since than $|e_j| < |e|$ for finer faces e. Hence, with this approach to coarse-grid matrices, we are are overpenalizing the jumps of DG functions on coarse level by a factor that is proportional to the number of subdivisions applied to a coarse face to obtain a fine face. Similar arguments hold for p-transfers as well, since the penalization factor σ is assumed to be of order p^2.

The former argument suggests that the coarser matrices exhibit a worsening condition depending on the number of levels. Indeed, in [8] it was proved that the use of plain Galerkin restriction for the coarse-grid bilinear forms ("inherited" bilinear forms) leads to a dependence of the convergence rate on the number of levels, rendering the multigrid method in this case as not grid independent. As a remedy, most authors choose to not use inherited bilinear forms but to re-discretize the problem on each level, i. e.

$$\tilde{A}^j_{kl} = a_h \left(\varphi^j_k, \varphi^j_l \right)$$

for a given basis $\left\{ \varphi^j_i \right\}_i$ of V^j.

In this work, however, we will not follow that approach. Besides the fact that assembling a matrix from scratch is a relatively costly operation, the full coarse levels might not be directly accessible due to the way how locally refined grids are stored. More importantly, the truncation of basis functions as introduced by the TNNMG algorithm (see Section 4.1) would lead to basis functions on coarser levels which lack some of their finer components due to truncation and that must be recalculated in every TNNMG iteration step. Moreover, explicitly computing these might arguably be quite involved. To keep computational efficiency and since the TNNMG method only requires a rather rough approximation of the solution to the arising linear systems anyway, we accept the level dependence of the method and increase the number of smoothing steps if required.

Remark 4.7. Another approach would be to store the penalty terms of the matrix representation separately and use these to subtract parts of the overpenalized terms in the coarse grid representation when using Galerkin restriction. In our experiments, this approach led to better convergence rates than using plain Galerkin restriction. However, this comes of course at a significant computational cost and increased complexity of the underlying code, so we decided to abandon this approach.

The remaining building blocks are the choices of a suitable smoother and optionally, a coarse grid solver. In practice, using the chosen smoother as coarse grid "solver" will suffice, given that the system corresponding to the coarsest level is sufficiently small. The coarse grid "solver" will be written as the application of a matrix B_1, which may or may not be equal to $\left(A^1\right)^{-1}$.

For simplicity and because of its good smoothing properties, we will restrict ourselves to simple (Block-)Gauss–Seidel smoothers. It is, however, reported, that other methods, e. g. polynomial smoothers [14] or overlapping Schwarz preconditioners (see, e. g. [106]) work well for DG-discretized systems.

The Gauss–Seidel smoothers for linear systems such as (4.11) are defined just as their nonlinear counterpart (4.4)–(4.6), in this case using the unconstrained quadratic functional

$$\mathcal{J}_0(v) = \frac{1}{2}\langle A^j v, v\rangle - \langle b^j, v\rangle$$

as the underlying energy. Then, the (linear) Gauss–Seidel method is to successively minimize energy in the coordinate directions (or in small subspaces for the blocked version). On each level j, this reads

$$\begin{aligned} F_i^j(x) &= x + \operatorname*{arg\,min}_{e \in V_i^j} \mathcal{J}_0(x+e), \\ \mathcal{F}^j &= F_{n_j}^j \circ \cdots \circ F_1^j, \end{aligned}$$

where $V_1^j, \ldots, V_{n_j}^j$ is a partition of $\mathbb{R}^{\dim(V^j)}$ into subspaces, e. g. $V_i^j = \operatorname{span}\{e_i\}$. It is also conceivable to use other subspaces, e. g to exploit the natural element-wise blocking when using discontinuous Galerkin spaces. For this unconstrained system, the local minimizations can often be performed exactly (up to floating point precision).

Besides Gauss–Seidel, other suitable suitable smoothers can be used. Hence, we will assume for each level j some smoother has been chosen and it can be represented by the application of a matrix B^j. For Gauss–Seidel, we would have $B^j = (L^j + D^j)^{-1}$, where L^j and D^j are the lower-diagonal and diagonal parts of A^j, respectively. However, it has to be guaranteed that the smoothers (and also the coarse grid solver B^1) can deal with truncated rows and columns due to truncated subspaces in the TNNMG method. This can e. g. be achieved for example by explicitly ignoring the corresponding indices (this can be easily be done in the Gauss–Seidel method since the subspaces are treated individually) or by replacing the diagonal entry of a truncated row by 1 instead of 0.

In Algorithm 4, we define the application of a single multigrid step. Note that we use the residual as input parameter. Thus, if we have an approximation x_ν^J on the finest level J, we compute the residual $r^J = b^J - A^J x_\nu^J$ and call MultigridStep(r^J, J), which will give a correction e_ν^J. For the linear correction problems in Algorithm 4 we usually use $x^j = 0$ as initial guess. Then, the current iterate x_ν^J is obtained through

$$x_{\nu+1}^J = x_\nu^J + e_\nu^J.$$

Choice of Subspaces

It remains to clarify how a suitable strategy for the choice of coarse spaces should look like. In this work we chose to first reduce the ansatz degree and afterwards, after having piecewise $\mathcal{Q}^1$ function on all elements, perform a geometric multigrid approach. More formally, consider the "fine" space $V_{\mathcal{T}_J}^{p_J}$ and pick an increasing sequence of degrees between 1 and the maximal degree of p_J,

$$1 = k_1 < k_2 < \cdots < k_m = \max_{K \in \mathcal{T}_J} p_J(K).$$

Algorithm 4 Single Step of Multigrid Scheme

```
procedure MultigridStep(Residual $r$, Level $j$)
    $x = 0 \in \mathbb{R}^{\dim(V_j)}$                    ▷ Create correction vector
    if $j = 1$ then
        $x \leftarrow B^1 r$                              ▷ (Approximately) solve system
    else
        for $i = 1, \dots, \mu_1$ do
            $x \leftarrow B^j x$                          ▷ Apply $\mu_1$ smoothing steps
        end for
        $r \leftarrow r - A^j x$                          ▷ Compute new residual
        $r^{j-1} = R_j^{j-1} r$                           ▷ Restrict to coarser space
        $c = $ MultigridStep($r^{j-1}, j-1$)              ▷ Compute coarse correction
        $x \leftarrow x + P_{j-1}^j c$                    ▷ Update correction vector
        for $i = 1, \dots, \mu_2$ do
            $x \leftarrow B^j x$                          ▷ Apply $\mu_2$ smoothing steps
        end for
    end if
    return $x$
end procedure
```

The sequence is arbitrary and may for example contain all integers between 1 and k_m or just hold powers of two. Mathematically it is not crucial which sequence is chosen, however computationally one may need to trade off convergence rates and time per iteration.

With a given degree sequence, the finest spaces are constructed by using the fine grid $\mathcal{T}_J$ and locally capped degree functions, i. e. on any fine element $K \in \mathcal{T}_J$, we have

$$p_{J-i}(K) = \min\big(p_J(K), k_{m-i}\big), \quad i = 0, \dots, m-1.$$

Thus, the fine spaces are defined as the DG spaces on grid $\mathcal{T}_J$ and degree functions as above,

$$V_{\mathcal{T}_J}^{p_{J-m+1}} \subset \cdots \subset V_{\mathcal{T}_J}^{p_J}. \tag{4.15}$$

Clearly, by construction, we have $p_{J-m+1} \equiv 1$. We will therefore also write $V_{\mathcal{T}_j}^1$ for spaces that contain only piecewise linear functions.

Up to now, we only constructed coarse levels by reducing the degree of the local ansatz functions. This is also called p-multigrid, see e. g. [84], and is a viable strategy on its own. The coarse space $V_{\mathcal{T}_J}^1$, however, is usually still too big to be solved directly within reasonable time. Therefore another iterative scheme might be applied to the system

$$A^{J-m+1}x = b, \quad x, b \in \mathbb{R}^{\dim(V_{\mathcal{T}_J}^1)}.$$

If the grid includes a hierarchy of coarser levels (i. e. it was obtained through geometric refinement of a coarse grid $\mathcal{T}_1$), we apply a geometric multigrid approach by defining the coarse spaces as the DG spaces with $\mathcal{Q}^1$ functions on the coarser levels:

$$V^1_{\mathcal{T}_1} \subset \dots V^1_{\mathcal{T}_l} = V^1_{\mathcal{T}_J}. \tag{4.16}$$

Combining (4.16) and (4.15), the sequence of spaces

$$V^1_{\mathcal{T}_1} \subset \dots V^1_{\mathcal{T}_l} \subset V^{p_{l+1}}_{\mathcal{T}_J} \subset \dots \subset V^{p_J}_{\mathcal{T}_J}$$

defines a *hp*-multigrid method that stacks geometric multigrid and *p*-multigrid on top of each other.

Remark 4.8. If the underlying mesh is conforming (or has conforming coarser levels), a particular elegant way of structuring the coarse grid solver might be to use *continuous* finite element spaces as coarse grid solvers. These often show superior multigrid convergence and are not prone to the overpenalizing introduced through the non-inherited bilinear forms. For further evidence, in [39], it is argued that an additive Schwarz scheme can only work if coarser spaces are spanned by continuous functions. In the popular approach from [21], the system reduced to a subspace containing only continuous finite element functions (which is solved with an algebraic multigrid approach) serves as coarse grid correction.

For systems that are very nonconforming, e. g. through a large number of hanging nodes generated by local adaptive refinement, the construction of suitable continuous finite element spaces is however not always feasible.

4.3. Parallel Smoothers

Error estimates in the numerical analysis of PDEs are usually stated in a form that, loosely speaking, implies that a higher number of degrees of freedom lead to more accurate approximations of the solution to the continuous problem. Thus, one is in general interested in computing problems large enough to comply with given error thresholds, possibly bounded by the available computing power. Rapid advancements in computer technology lead to ever growing problem sizes in the last decades. However, it turned out that the predicted exponential growth of computing power for a single computer (Moore's "law") is saturating. As a remedy (among other variants such as, e. g. using architectural parallelization through SIMD) a common approach is to split the problem into parts and distribute them among many processes ("nodes"). These processes can reside on different machines and communicate through some protocol, usually the Message Passing Interface (MPI)[1] standard.

Since communication is slow compared to the computations on each node, we want an approach that allows to perform as much work as possible locally. Thus, the common practice is to apply a domain decomposition ansatz by splitting the grid into parts. We will not, however, follow the classical domain decomposition technique of solving (e. g. with a multigrid approach) in each subdomain and combining (possibly amended by the use of a global coarse space) the solutions of the individual parts to

[1] https://www.mpi-forum.org

a global solution. While this approach requires less communication, the convergence of the algebraic solver will usually suffer due to the fact that less information and coupling is used. Moreover, it is not so clear how to solve the nonlinear problems in this fashion (see for example [107] for an attempt for variational inequalities). Rather, we will employ a multigrid strategy as outlined in the previous sections to the global problem. Hence, we can solve the problem in the same fashion as we are used to from the serial case, with the crucial difference that the building blocks of the solver (e. g. matrix-vector-products, updates to residuals, application of smoothers, etc.) will be distributed between the machines and therefore frequent communication is required to keep the data on each machine in a valid state. We accept the increased communication costs (compared to the other approach where the local problems are first solved and commucation happens afterwards) for the benefit of faster solvers and the possibility to reuse our existing solvers, in particular the TNNMG method. At some points, algorithms which are inheritely non-parallel (think, e. g., the Gauss–Seidel method) still have to be adapted or replaced to allow for parallel computations. Naturally, we have to make sure that the employed algorithms still yield satisfying properties even when performed in parallel. Moreover, we have to make sure that we do not lose any properties that are essential for the convergence of the methods (e. g. the error reduction property of the nonlinear smoother).

Regardless of whether one chooses to use domain decomposition with multigrid or (as we decided to) the parallel multigrid approach, the technical starting point is similar: The domain will be split into a number of sub-domains matching the number of available nodes. Each node will only know its part of the grid (and maybe some of the neighboring elements in form of *ghost* or *overlap* elements) and perform computations on that sub-domain. For the PDE numerics framework DUNE[2] which is used for the numerical examples in this thesis, the parallelization approach is described in detail in [19, 20, 25]. From a technical perspective, these methods are well understood, see e. g. [17] for a detailed explanation. We will therefore only discuss the parts that might introduce problems in our case. This concerns mostly the smoothers used for the algebraic solution of the problems, both the nonlinear smoother in the beginning of each step of the TNNMG method and the linear smoothers used on each level of the multigrid step. The classical approach is to apply a hybrid Gauss–Seidel method, i. e. performing a Gauss–Seidel step locally on each node in parallel and communicating the result afterwards. This is similar to a block-Jacobi method with inexact block solvers, given that the degrees of freedom are appropriately blocked.

Such a hybrid Gauss–Seidel method, however, does not converge for all s. p. d. matrices [14]. Moreover, it is not clear if the nonlinear variant of the hybrid method converges, a result crucial for the converges proof of the TNNMG method. In the linear case, convergence can be guaranteed with proper damping of the corrections. The estimation of a suitable damping factor, however, is not trivial. As a remedy Baker et al. [14] introduced so-called ℓ_1-smoothers that are proven to converge without the need to explicitly compute proper damping factors. This, however, does not answer the question of the convergence of the nonlinear variant of the smoother.

[2] https://www.dune-project.org

In the following section, we will prove a sufficient criterion for the convergence of a preconditioned nonlinear smoother. As a corollary, we obtain that the nonlinear variant of the ℓ_1-smoother converges, rendering it as a good candidate for the use in a parallel TNNMG method.

4.3.1. Preconditioned Nonlinear Gauss–Seidel

We seek for a minimizer to the following convex energy:

$$\mathcal{J}(x) = \frac{1}{2}\langle Ax, x\rangle - \langle b, x\rangle + \sum_i \varphi_i(x_i), \tag{4.17}$$

where $A \in \mathbb{R}^{N\times N}$ is a symmetric, positive definite matrix, b is a linear functonal and the φ_i have the same properties as in Assumption 4.2. As argued before, $\mathcal{J}$ has a unique minimizer $u \in \mathbb{R}^n$.

As outlined in Section 4.1.1, functionals of the form (4.17) can be minimized through a nonlinear Gauss–Seidel method, i. e. by successively minimizing energy in subspaces V_i.

Reformulating (4.6) in a correction form, we have

$$x^{\nu+1} = \mathcal{M}(x^\nu) = x^\nu + \mathcal{F}(x^\nu), \tag{4.18}$$

with the correction operator $\mathcal{F} = \mathcal{M} - \mathrm{Id}$. More general, we introduce the notation $\mathcal{M}^G$, which shall denote the application of a nonlinear Gauss–Seidel step to a specific functional $G : \mathbb{R}^N \to \mathbb{R}$. In particular, we have $\mathcal{M}^{\mathcal{J}} = \mathcal{M}$. Similarly, we introduce the correction operator $\mathcal{F}^G = \mathcal{M}^G - \mathrm{Id}$.

For arbitrary $v \in \mathbb{R}^N$, rewriting (4.17) gives

$$\mathcal{J}(x+v) = \mathcal{J}(x) + \frac{1}{2}\langle Av, v\rangle - \langle b - Ax, v\rangle + \sum_i \varphi_i(x_i + v_i) - \varphi_i(x_i).$$

Denoting the residual part by

$$D_x(v) = \frac{1}{2}\langle Av, v\rangle - \langle b - Ax, v\rangle + \sum_i \varphi_i(x_i + v_i) - \varphi_i(x_i),$$

we have $\mathcal{J}(x+v) = \mathcal{J}(x) + D_x(v)$. Clearly, D_x is a convex functional with the same properties as $\mathcal{J}$. Instead of evaluating the nonlinear Gauss–Seidel $\mathrm{Id} + \mathcal{F}^{\mathcal{J}}$ at x^ν to get the new iterator $x^{\nu+1}$, we can also get the correction $x^{\nu+1} - x^\nu$ by performing a nonlinear Gauss–Seidel step $\mathcal{F}^{D_x}$ to the functional D_x at 0. Thus, we can rewrite the iteration (4.18) as

$$x^{\nu+1} = x^\nu + \mathcal{F}^{D_{x^\nu}}(0).$$

In particular, we have $\mathcal{F}(x^\nu) = \mathcal{F}^{D_{x^\nu}}(0)$. This also resembles a frequently used technique when implementing the smoother.

We are now interested in computing a preconditioned version of the Gauss–Seidel algorithm. More precisely, we want to replace the matrix A in the quadratic part of the functional by another matrix B which might be more suitable for specific reasons (in our case, e. g., to reduce coupling). For the modified iteration, consider a symmetric matrix $B \in \mathbb{R}^{N\times N}$ such that we have

$$0 \leq \langle Ax, x\rangle \leq \langle Bx, x\rangle \quad \forall x \in \mathbb{R}^N. \tag{4.19}$$

Obviously, with A being positive definite, B is also positive definite. This allows us to modify the functional D_x by defining

$$\tilde{D}_x(v) = \frac{1}{2}\langle Bv, v\rangle - \langle b - Ax, v\rangle + \sum_i \varphi_i(x_i + v_i) - \varphi_i(x_i),$$

where we replaced A by B in the *quadratic part.* Similarly as before, we introduce the notation $\tilde{\mathcal{F}}(x) := \mathcal{F}^{\tilde{D}_x}(0)$ for the application of a nonlinear Gauss–Seidel step to the modified functional. Clearly, by (4.19), we have

$$D_x(v) \leq \tilde{D}_x(v) \quad \forall v \in \mathbb{R}^N. \tag{4.20}$$

Moreover, since $\tilde{D}_x$ is a quadratic functional just as $\mathcal{J}$, we get that the application of the nonlinear Gauss–Seidel method to $\tilde{D}_x$ at zero decreases energy:

$$\tilde{D}_x(\tilde{\mathcal{F}}(x)) \leq \tilde{D}_x(0) = 0. \tag{4.21}$$

It is well-known for Gauss–Seidel methods that in (4.21), we have equality if and only if 0 is the unique minimizer of $\tilde{D}_x(v)$, $v \in \mathbb{R}^N$.

We can now define a sequence of iterates induced by applying the nonlinear Gauss–Seidel method to the modified residual $\tilde{D}_x$, i. e.

$$\tilde{x}^{\nu+1} = \tilde{x}^{\nu} + \tilde{\mathcal{F}}(\tilde{x}^{\nu}), \quad \nu = 0, 1, \ldots. \tag{4.22}$$

This algorithm is the same as the Gauss–Seidel method only using the matrix B instead of A in the quadratic part.

Since the nonlinear smoother will be integrated into a TNNMG method, it is crucial that the sequence (4.22) is a minimizing sequence. This will be shown in the following lemma.

Lemma 4.9. *Let the sequence* $\{\tilde{x}^{\nu}\}_{\nu}$ *be defined as in* (4.22)*. Then, for every* $\nu \in \mathbb{N}$*, we have*

$$\mathcal{J}(\tilde{x}^{\nu+1}) \leq \mathcal{J}(\tilde{x}^{\nu})$$

with equality if and only if $\tilde{x}^{\nu} = \tilde{x}^{\nu+1} = u$*.*

Proof. Let $\nu \in \mathbb{N}$.

$$\begin{aligned}
\mathcal{J}\left(\tilde{x}^{\nu+1}\right) &= \mathcal{J}\left(\tilde{x}^{\nu} + \tilde{\mathcal{F}}(\tilde{x}^{\nu})\right) \\
&= \mathcal{J}\left(\tilde{x}^{\nu}\right) + D_{\tilde{x}^{\nu}}\left(\tilde{\mathcal{F}}(\tilde{x}^{\nu})\right) \\
&\leq \mathcal{J}\left(\tilde{x}^{\nu}\right) + \tilde{D}_{\tilde{x}^{\nu}}\left(\tilde{\mathcal{F}}(\tilde{x}^{\nu})\right) && \text{(by (4.20))} \\
&\leq \mathcal{J}\left(\tilde{x}^{\nu}\right). && \text{(by (4.21))}
\end{aligned}$$

For the second part, assume it holds $\mathcal{J}(\tilde{x}^{\nu+1}) = \mathcal{J}(\tilde{x}^{\nu})$. This implies

$$\tilde{D}_{\tilde{x}^\nu}\left(\tilde{\mathcal{F}}(\tilde{x}^\nu)\right) = 0.$$

Thus, 0 must be the unique minimizer of $\tilde{D}_{\tilde{x}^\nu}$ and we have that $\tilde{F}(\tilde{x}^\nu) = 0$.

We need to show that no correction through $\tilde{D}$ implies that also no correction through D would have been made and hence $\tilde{x}^\nu = u$.

Let $\varepsilon := \mathcal{F}(x^\nu)$ be the correction the nonlinear Gauss–Seidel method applied to the (unmodified) functional $D_{\tilde{x}^\nu}$ yields, and analogously $\tilde{\varepsilon} := \tilde{\mathcal{F}}(\tilde{x}^\nu)$ for the modified functional. Let $i \in \{1, \dots, n\}$ be the lowest index such that $\varepsilon_i \neq 0$ (if no such index exists, the statement follows immediately). We have

$$\begin{aligned}\tilde{\varepsilon}_i &= \operatorname*{arg\,min}_{\alpha \in \mathbb{R}} \frac{1}{2}\alpha^2 B_{ii} - \alpha(b - A\tilde{x}^\nu)_i + \varphi_i(\tilde{x}_i^\nu + \alpha) - \varphi_i(\tilde{x}_i^\nu),\\ \varepsilon_i &= \operatorname*{arg\,min}_{\alpha \in \mathbb{R}} \frac{1}{2}\alpha^2 A_{ii} - \alpha(b - A\tilde{x}^\nu)_i + \varphi_i(\tilde{x}_i^\nu + \alpha) - \varphi_i(\tilde{x}_i^\nu),\end{aligned}$$

ignoring terms that are independent of α. Both equations only differ by the factor in front of the quadratic term. As both A_{ii} and B_{ii} are greater than 0, we can write both minimization problems as (scalar) variational inequalities of the second kind:

$$\begin{aligned}b(\tilde{\varepsilon}_i, v - \tilde{\varepsilon}_i) - \ell(v - \tilde{\varepsilon}_i) + j(v) - j(\tilde{\varepsilon}_i) &\geq 0 \qquad &\forall v \in \operatorname{dom}(j),\\ a(\varepsilon_i, v - \varepsilon_i) - \ell(v - \varepsilon_i) + j(v) - j(\varepsilon_i) &\geq 0 \qquad &\forall v \in \operatorname{dom}(j).\end{aligned}$$

Here, we used the notation $a(v, w) = A_{ii}vw$, $b(v, w) = B_{ii}vw$, $\ell(v) = v(b - A\tilde{x}^\nu)_i$ and $j(v) = \varphi_i(\tilde{x}_i^\nu + v) - \varphi_i(\tilde{x}_i^\nu)$ with $v, w \in \mathbb{R}$.

Since $\tilde{\varepsilon}_i = 0$ by assumption, we have

$$-\ell(v) + j(v) \geq 0 \tag{4.23}$$

for all admissible v. By assumption, $\varepsilon_i \neq 0$, hence we have, testing with $v = 0$,

$$0 = a(\varepsilon_i, 0) - \ell(0) + j(0) \geq a(\varepsilon_i, \varepsilon_i) - \ell(\varepsilon_i) + j(\varepsilon_i) > -\ell(\varepsilon_i) + j(\varepsilon_i),$$

a contradiction to (4.23). Hence, it follows that whenever the Gauss–Seidel minimization of $\tilde{D}_{\tilde{x}^\nu}$ gives no correction, the analogous minimization of $D_{\tilde{x}^\nu}$ would not yield any correction either, hence $\tilde{x}^\nu = u$.

The other direction, namely $\tilde{x}^\nu = u$ implying $\mathcal{J}(\tilde{x}^{\nu+1}) = \mathcal{J}(\tilde{x}^\nu)$, follows directly from the uniqueness of the minimizer. □

Lemma 4.10. *The operator $\tilde{\mathcal{F}}$ is Lipschitz continuous.*

Proof. Apply Lemma A.4 for the individual subspaces. Inductively, one can see there is a constant C such that $\|\mathcal{F}(x) - \mathcal{F}(\tilde{x})\| \leq C\|x - \tilde{x}\|$. □

Equipped with these lemmas, the following theorem ensures the convergence of the nonlinear preconditioner. This in particular guarantees the convergence of the TNNMG algorithm [67].

Theorem 4.11. *Assume B is symmetric, positive-definite and* (4.19) *holds. Then, the TNNMG algorithm using the preconditioned nonlinear Gauss–Seidel smoother* $\tilde{\mathcal{M}} = \operatorname{Id} + \tilde{\mathcal{F}}$ *converges.*

Proof. Use the monotonicity proved in Lemma 4.9, deduce the continuity of $\mathcal{J} \circ \tilde{\mathcal{M}}$ on $\operatorname{dom} \mathcal{J}$ from Lemma 4.10 and finally apply [67, Theorem 4.1] or [61, Theorem 4.1]. □

4.3.2. Application: Parallel TNNMG using Nonlinear ℓ_1-Smoother

In the following, we construct a nonlinear variant of the ℓ_1-smoother derived in [14]. The smoother replaces the matrix A by a matrix B that is block-diagonal where the blocks are induced by the domain decomposition (given a suitable numbering of unknowns). As we learned in Section 4.3.1, replacing A with B when doing the nonlinear Gauss–Seidel step will not harm the convergence, given that (4.27) holds. If the matrix B decouples parts of the matrix from each other (through the block diagonal structure), we can *compute* the correction $\tilde{\mathcal{F}}(x)$ in parallel while theoretically performing a (inherently sequential) nonlinear Gauss–Seidel step.

We introduce a partition of the index set $P = \{1, \ldots, N\}$ into p disjoint nonempty subsets, i. e.

$$P = \dot{\bigcup}_{k=1}^{p} P_k \tag{4.24}$$

with $P_k = \big\{j_1, \ldots, j_{n_k}\big\} \subset P$. In practice, p will be the number of nodes in a distributed setup. For every index i there is a unique $k(i) \in \{1, \ldots, p\}$ such that $i \in P_{k(i)}$. For this i, we associate the set of all indices that do not belong to the same index set $P_{k(i)}$:

$$P_o^{(i)} = P \setminus P_{k(i)}.$$

The idea of the ℓ_1-smoother is to remove all matrix entries A_{ij} where $k(i) \neq k(j)$ and adding them to the diagonal term A_{ii}. This immediately removes all coupling between the nodes.

More formally, we set $B = A^{\text{local}} + D^{\ell_1}$, where

$$A^{\text{local}}_{ij} = \begin{cases} A_{ij} & \text{if } k(i) = k(j), \\ 0 & \text{if } k(i) \neq k(j), \end{cases} \quad i, j = 1, \ldots, N \tag{4.25}$$

and D_{ℓ_1} is a diagonal matrix with entries

$$D^{\ell_1}_{ii} = \sum_{j \in P_o^{(i)}} |A_{ij}|, \quad i = 1, \ldots, N. \tag{4.26}$$

As mentioned in [14], we have the following estimate:

Lemma 4.12. *For $A \in \mathbb{R}^{N \times N}$ s.p.d., and $B \in \mathbb{R}^{N \times N}$ defined as before, we have*

$$\langle Av, v \rangle \leq \langle Bv, v \rangle, \quad \forall v \in \mathbb{R}^N. \tag{4.27}$$

Proof. Let $v \in \mathbb{R}^N$ be arbitrary.

$$\begin{aligned} \langle Av, v \rangle &= \langle A^{\text{local}} v, v \rangle + \langle (A - A^{\text{local}}) v, v \rangle \\ &\leq \langle A^{\text{local}} v, v \rangle + \sum_i \sum_{j \in P_o^{(i)}} |A_{ij}||v_i||v_j|. \end{aligned}$$

Using A's symmetry, we have $|A_{ij}||v_i||v_j| = |A_{ji}||v_j||v_i|$. Hence, we can rewrite the former inequality as

$$\langle Av, v \rangle \leq \langle A^{\text{local}} v, v \rangle + \sum_i \sum_{\substack{j \in P_o^{(i)} \\ j > i}} 2|A_{ij}||v_i||v_j|.$$

Using Young's inequality, we have $2|A_{ij}||v_i||v_j| \leq |A_{ij}|v_i^2 + |A_{ij}|v_j^2$. A short calculation shows that

$$\sum_i \sum_{\substack{j \in P_o^{(i)} \\ j > i}} |A_{ij}|v_i^2 + |A_{ij}|v_j^2 = \sum_i v_i^2 \left(\sum_{j \in P_o^{(i)}} |A_{ij}| \right),$$

and hence

$$\langle Av, v \rangle \leq \langle A^{\text{local}} v, v \rangle + \langle D^{\ell_1} v, v \rangle = \langle Bv, v \rangle.$$

□

Equation (4.27) shows that we can use the modified matrix B in our smoothers as presented above. As argued before, this matrix can be used in a parallel computation as a Gauss–Seidel method which is computationally equivalent to a (blocked) Jacobi method. In particular, the convergence of the nonlinear smoother inside the TNNMG method is guaranteed through Theorem 4.11, yielding the convergence of the TNNMG method itself, cf. Theorem 4.4.

Also, when computing the linear correction of the TNNMG algorithm using a multigrid method, the same way of constructing a matrix B_J from A_J on the J-th level can be used in the linear smoothers. Convergence for this linear case was shown in [14] and is also a special case of the proof for the nonlinear case, choosing $\varphi_i \equiv 0$ for all i.

5. Adaptive Numerical Approximation

Solutions to the problems of Chapter 2 (and in general PDE problems) often times exhibit local phenomena that will only be visible if the discrete space can resolve them sufficiently well. Often, this means having a very fine mesh width, ansatz functions of higher polynomial degree, or both. An example in the context of phase field models is the interface area between different phases where the solution has a relatively sharp gradient. Another example for the obstacle problem was motivated earlier in the a priori *hp* estimate in Theorem 3.50. There, it was concluded that near the free boundary the local mesh width has to be fine enough to account for the local nonsmoothness while ansatz functions of higher order might not yield further benefits. On the other hand, it was shown that away from the free boundary, a high polynomial degree in the ansatz functions would be superior compared to a finer mesh (compare the $\mathcal{O}(h^p)$ convergence in p to the $\mathcal{O}(h)$ convergence in h). A naive solution to the dilemma would be to employ both a very fine mesh and a high polynomial degree globally. This, however, would lead to a very large finite element space where the discrete solution cannot be computed in a reasonable time if at all. A more economical approach is to locally adapt the discrete space such that the mentioned local phenomena are sufficiently resolved while keeping a coarser mesh where possible. Since the exact position where a higher resolution is needed is usually not known a priori (think again, e. g. of the free boundary of an obstacle problem), one has to set up a procedure that will identify the crucial areas in a given discrete solution and successively generate new discrete solutions on a finer finite element space. To be successful, such a procedure has to be build around an error estimator that serves (at least) two purposes. First, it needs to give a realistic estimate of the (unknown) global discretization error $\|u - u_h\|$, where u is the solution to a given PDE problem and u_h the solution to a discretized version of the same problem. Once this error is small enough, the procedure can terminate. Second, the error estimator should give a local criterion which allows an algorithm to identify the regions where a finer resolution will let the global error decrease most rapidly. If such a local criterion were not available, we would have no means but to refine uniformly, a technique that would lead to too large discrete spaces as argued before.

Let η be an estimate of the global error $\|u - u_h\|$. Two properties are expected from a suitable error estimator [28]:

1. The error estimator should be *reliable*, i. e.

$$\|u - u_h\| \lesssim \eta,$$

2. and it should be *efficient*, i. e.

$$\eta \lesssim \|u - u_h\| .$$

In summary, we want $\eta \approx \|u - u_h\|$ that can be computed without the explicit knowledge of u.

The chapter is organized as follows: At first, we will recall how hierarchical error estimators can be used to solve a linear PDE problem when a continuous $\mathcal{P}^1$ finite element discretization is used. Particularly, we explain how we can modify the problem of finding an estimator in a way that allows for a more efficient computation. Afterwards, we explain how the approach is used in a variational inequality setting. Having explained the approach for the classical (that is, continuous) Galerkin methods, we will switch to the discontinuous finite element spaces we are considering in this thesis. We will show how to apply a similar approach of hierarchical error estimating for the SIPG discretization for the linear model problem. Finally, we argue how the methods can be transferred to the error estimation of discretized variational inequalities.

5.1. Hierarchical A Posteriori Error Estimation

In this section, we will investigate a particular technique, namely hierarchical error estimators, of estimating the discretization error that is almost independent of the particular problem at hand. While there are several results for residual based error estimators for the problems presented in Chapter 2 discretized with discontinuous Galerkin methods (see, e. g. [72, 95]), there are few results for hierarchical error estimates in the DG context (see e. g. [15]). However, residual error estimators are often only efficient up to oscillation terms (see, e. g. [31]). Moreover, hierarchical error estimators impress through their simplicity, which makes it easy to transfer the approach from the simpler linear model problem to variational inequalities. Therefore, we will introduce the concepts and some results about hierarchical error estimators as they are known for continuous, piecewise linear finite elements and extend the ideas to discontinuous finite elements with varying order.

For a given elliptic PDE problem, let $u \in H$ be the solution to its weak formulation and $u_{\mathcal{S}} \in \mathcal{S}$ be the solution to a discretization in a suitable discrete space $\mathcal{S}$. We are interested in the error $\|u - u_{\mathcal{S}}\|$ in a suitable norm. Since we do not know u, we can only estimate by other means. For hierarchical error estimators, one considers another discrete space $\mathcal{Q}$ and the discrete solution in that space, $u_{\mathcal{Q}}$. Usually $\mathcal{Q}$ is obtained through an enlargement of $\mathcal{S}$ (hence $\mathcal{S} \subset \mathcal{Q}$), for example through a uniform grid refinement or by increasing the polynomial degree. We will assume that u is strictly better approximated in $\mathcal{Q}$ than it is in $\mathcal{S}$. This is manifested in the following saturation assumption:

Assumption 5.1 (Saturation Assumption)**.** *For $u_{\mathcal{S}}$ and $u_{\mathcal{Q}}$ being the discrete solutions in $\mathcal{S}$ and $\mathcal{Q}$, respectively, we assume there is a $\beta < 1$ such that*

$$\|u - u_{\mathcal{Q}}\| \leq \beta \|u - u_{\mathcal{S}}\| \,. \tag{5.1}$$

Obviously, the saturation assumption depends not only on the chosen spaces $\mathcal{S}$ and $\mathcal{Q}$ but also on their suitability for the given problem. Spaces that are well suited for a particular problem might perform badly at another problem, so a careful selection is advisable. For many problems, it can be argued that the saturation assumption is true if the problem data has only small oscillations [50]. Note that, however, in general counterexamples can be constructed such that Assumption 5.1 is not valid [28, Proposition 2.2].

The key idea is to compare the discrete solutions $u_{\mathcal{S}}$ and $u_{\mathcal{Q}}$ and consider their difference as an estimate of the discretization error $\|u - u_{\mathcal{S}}\|$. It is well known that the saturation assumption 5.1 implies that the difference $\eta = \|u_{\mathcal{S}} - u_{\mathcal{Q}}\|$ is an efficient and reliable error estimator [28], as summarized in the following Theorem:

Theorem 5.2. *Suppose Assumption 5.1 is true. This implies for $\eta = \|u_{\mathcal{S}} - u_{\mathcal{Q}}\|$ that*

$$\frac{1}{1+\beta}\eta \le \|u - u_{\mathcal{S}}\| \le \frac{1}{1-\beta}\eta$$

holds.

Proof. Apply the triangle inequality and insert (5.1). □

Remark 5.3. The theorem and its proof above assume that for all spaces the same norm $\|\cdot\|$ can be applied. This might not be true for grid dependent norms as they arise e. g. for Interior Penalty DG. Extra care for the norm equivalences has to be paid in those cases, see e. g. [15] for a treatment of obstacle problems discretized with IPDG.

Theorem 5.2 directly shows that the quality of the hierarchical error estimator is tightly linked to the constant β, i. e. to which extend the discrete solution $u_{\mathcal{Q}}$ is able to capture phenomena beyond $u_{\mathcal{S}}$. In particular if β is close to one (meaning $\mathcal{Q}$ does resolve the solution better than $\mathcal{S}$ only by a small amount), the constant $\frac{1}{1-\beta}$ will be very large which reduces the error estimator's reliability.

For the rest of this section, we assume that the saturation assumption 5.1 holds with a "reasonable" constant β. While in theory this suffices to estimate the discretization error reliably and efficiently, we are left with the challenge of solving an additional discrete problem, namely to compute $u_{\mathcal{Q}}$. By assumption, we have that $\mathcal{Q}$ is considerably larger than $\mathcal{S}$. For example if one considers $\mathcal{Q}$ to be the uniform h-refinement of a DG space $\mathcal{S}$ on a three-dimensional grid consisting of cubes, one would have that $\mathcal{Q}$ consists of 8 times as many unknowns as $\mathcal{S}$. Fully assembling and solving the larger problem is hence prohibitively expensive and would directly oppose to the idea of only solving local finer problems to save computing time. Therefore, a second layer of approximation was introduced that approximates the solution $u_{\mathcal{Q}}$ on the finer space by an approximation $\tilde{u}_{\mathcal{Q}} \in \mathcal{Q}$ that is easier to compute. If one can show (or at least has a heuristic reasoning) that the induced approximate error $\tilde{\eta} = \|u_{\mathcal{S}} - \tilde{u}_{\mathcal{Q}}\|$ is equivalent to the error estimator η, one gets the another efficient and reliable error estimator at a lower computational cost. The approximation $\tilde{u}_{\mathcal{Q}}$ is often achieved by replacing a bilinear form $a(\cdot, \cdot)$ of the given problem by another bilinear form $b(\cdot, \cdot)$ and solving the defect problem with respect to this bilinear form. This approach can be viewed as a preconditioning technique. The formulation as a preconditioner was first made explicit in [46]. Many concepts for preconditioning in the context of hierarchical error estimates are closely related to domain decomposition techniques, see e. g. [28]. As an example, we will briefly demonstrate how diagonal scaling can be used for a simple Poisson problem. For more details, see [28].

Example 5.3.1. Consider the discretized Poisson problem

$$u_{\mathcal{S}} \in \mathcal{S} : a(u_{\mathcal{S}}, v) = (\nabla u_{\mathcal{S}}, \nabla v) = \langle f, v\rangle \quad \forall v \in \mathcal{S},$$

where $\mathcal{S} = \mathcal{P}^1(\mathcal{T})$ is the finite element space of piecewise linear, continuous functions on a given triangulation $\mathcal{T}$ of $\Omega \subset \mathbb{R}^2$. Here, we assume that $u_{\mathcal{S}}$ is known exactly, i. e. we ignore any algebraic error stemming from, e. g., an iterative solution procedure. The corresponding defect problem for computing $d = u_{\mathcal{Q}} - u_{\mathcal{S}}$ is

Problem 5.4 (Defect problem).

$$\begin{aligned} d \in \mathcal{Q} : a(d, v) &= r_{u_{\mathcal{S}}}(v) \quad \forall v \in \mathcal{Q}, \\ r_{u_{\mathcal{S}}}(v) &= \langle f, v\rangle - a(u_{\mathcal{S}}, v). \end{aligned} \tag{5.2}$$

Typically, the larger space $\mathcal{Q}$ is constructed by adding linear independent basis functions $\{\psi_i : i = 1, \ldots, \dim(\mathcal{V})\} = \Psi$ (say, e. g. quadratic bubble functions) from another space $\mathcal{V} = \operatorname{span}\Psi$ (hence forming a *hierarchical* extension). These basis functions are the Lagrange basis functions with respect to a node set $\mathcal{N}_{\mathcal{Q}}$ and which vanish on the nodes $\mathcal{N}_{\mathcal{S}}$ of $\mathcal{S}$. The splitting $\mathcal{Q} = \mathcal{S} + \mathcal{V}$ can also be written as the hierarchical splitting

$$\mathcal{Q} = \mathcal{S} \oplus \bigoplus_{\psi\in\Psi} \operatorname{span}\{\psi\}, \tag{5.3}$$

hence every $w \in \mathcal{Q}$ can be uniquely written as

$$w = R_{\mathcal{S}} w + \sum_{i=1}^{\dim(\mathcal{V})} w_i, \qquad R_{\mathcal{S}} w \in \mathcal{S}, \quad w_i \in \operatorname{span}\{\psi_i\}.$$

Here, $R_{\mathcal{S}}$ is the orthogonal projection induced through the direct sum in (5.3).

To allow a cheaper approximation, we replace the bilinear form $a(\cdot, \cdot)$ by a preconditioner:

$$\begin{aligned} &a(v, w) \approx b(v, w) \\ &\qquad := a(R_{\mathcal{S}} v, R_{\mathcal{S}} w) + \sum_{i=1}^{\dim(\mathcal{V})} a(\psi_i, \psi_i)(v - R_{\mathcal{S}} v)(p_i)(w - R_{\mathcal{S}} w)(p_i), \end{aligned} \tag{5.4}$$

where $p_i \in \mathcal{N}_{\mathcal{Q}} \setminus \mathcal{N}_{\mathcal{S}}$ is the i-th Lagrange node corresponding to the basis function ψ_i. As we can see, the bilinear form $b(\cdot, \cdot)$ does not couple the subspaces involved in (5.3). The modified defect problem now reads:

Problem 5.5 (Preconditioned Defect problem).

$$\tilde{d} \in \mathcal{Q} : b(\tilde{d}, v) = r_{u_{\mathcal{S}}}(v) \quad \forall v \in \mathcal{Q}. \tag{5.5}$$

In particular, this defines an approximation of $u_{\mathcal{Q}}$,

$$\tilde{u}_{\mathcal{Q}} = \tilde{d} + u_{\mathcal{S}}. \tag{5.6}$$

Since $u_{\mathcal{S}}$ solves the variational problem on $\mathcal{S}$, we have $r_{u_{\mathcal{S}}}(v) = 0$ for $v \in \mathcal{S}$. Also, note that $v - R_{\mathcal{S}}v = 0$ if $v \in \mathcal{S}$ and therefore the latter terms in (5.4) vanish. Hence, we have

$$b(\tilde{d}, v) = a(R_{\mathcal{S}}\tilde{d}_{\mathcal{S}}, v) = a(R_{\mathcal{S}}(\tilde{u}_{\mathcal{Q}} - u_{\mathcal{S}}), v) = r_{u_{\mathcal{S}}}(v) = 0 \quad \forall v \in \mathcal{S}.$$

In particular, this implies $R_{u_{\mathcal{S}}}(\tilde{u}_{\mathcal{Q}} - u_{\mathcal{S}}) = 0$ and thus $R_{\mathcal{S}}u_{\mathcal{Q}} = u_{\mathcal{S}}$. Moreover, we have that b decouples $\mathcal{S}$ and $\mathcal{V}$ as for $w \in \mathcal{V}$ it follows

$$b(v, w) = 0 \quad \forall v \in \mathcal{S} \tag{5.7}$$

from $R_{\mathcal{S}}w = 0$. Therefore, we have $\tilde{d} = \tilde{u}_{\mathcal{Q}} - u_{\mathcal{S}} \in \mathcal{V}$ and suffices to solve the variational problem in the extension space:

$$\tilde{d} \in \mathcal{V} : b_{\mathcal{V}}(\tilde{d}, w) = r_{u_{\mathcal{S}}}(v) \quad \forall v \in \mathcal{V}. \tag{5.8}$$

Here, $b_{\mathcal{V}}(\cdot,\cdot)$ is the bilinear form b without the $\mathcal{S}$ contributions,

$$b(v, w) := \sum_{i=1}^{\dim(\mathcal{V})} a(\psi_i, \psi_i)(v - R_{\mathcal{S}}v)(p_i)(w - R_{\mathcal{S}}w)(p_i).$$

Since the resulting system is diagonal, we can solve the following local defect problems to obtain $\tilde{d}$:

$$\tilde{d}_i \in \operatorname{span}\{\psi_i\} : a(\tilde{d}_i, \psi_i) = r_{u_{\mathcal{S}}}(\psi_i), \quad i = 1, \ldots, \dim(\mathcal{V}). \tag{5.9}$$

This is equivalent to an additive Schwarz method on the subspaces spanned by each of $\mathcal{V}$'s basis functions which emphasizes the relationship to domain decomposition preconditioning techniques.

Finally, we can propose the following global error estimate:

$$\tilde{\eta}^2 = \sum_{i=1}^{\dim(\mathcal{V})} \tilde{\eta_i}^2$$

with $\tilde{\eta_i} = \left\|\tilde{d}_i\right\|$.

Remark 5.6. If we consider the energy norm $\|\cdot\| = a(\cdot,\cdot)$, we have

$$\tilde{\eta_i} = \frac{r_{u_{\mathcal{S}}}(\psi_i)^2}{a(\psi_i, \psi_i)}.$$

If we put certain assumptions on the extension space $\mathcal{V}$, namely that (5.3) is a *stable splitting*, i. e.

$$\|v\|^2 \approx \|v_{\mathcal{S}}\|^2 + \sum_{\psi \in \Psi} \|v_\psi\|^2, \tag{5.10}$$

where $v = v_{\mathcal{S}} + \sum_{\psi} v_{\psi}$ is the unique decomposition according to (5.3), we get the following equivalence, cf. [28, Theorem 3.1]:

$$\tilde{\eta} \approx \|u_{\mathcal{Q}} - u_{\mathcal{S}}\| = \eta. \tag{5.11}$$

Clearly, this implies that the preconditioned error estimator $\tilde{\eta}$ is also an efficient and reliable error estimator.

Moreover, the constructed error estimator directly gives us a criterion for local refinements as the local contributions $\tilde{\eta}_i$ carry information about how large the error is in the support of ψ_i.

Extension to Variational Inequalities

After having discussed the hierarchical error estimator for the linear problem, we want to briefly comment on how to use a similar approach for variational inequalities. Consider again the finite element space $\mathcal{S}$ and the hierarchical extension $\mathcal{V}$ such that $\mathcal{Q} = \mathcal{S} \oplus \mathcal{V}$. Assume $u_{\mathcal{S}}$ to be the solution to a discrete variational inequality

$$a(u_{\mathcal{S}}, v - u_{\mathcal{S}}) - \langle b, v - u_{\mathcal{S}} \rangle + j_{\mathcal{S}}(v) - j_{\mathcal{S}}(u_{\mathcal{S}}) \geq 0 \quad \forall v \in \mathcal{S}.$$

Here, we understand that j_V is an appropiate discretization of a nonlinearity ϕ in the respective finite element space V by using a quadrature rule, i. e. we have $j(v) = \sum_i \phi(v(x_i))\omega_i$, where ω_i is the integral of the i-th basis function of V and x_i its corresponding Lagrange node. Similarly, let $u_{\mathcal{Q}}$ be the solution to the variational inequality in $\mathcal{Q}$,

$$a(u_{\mathcal{Q}}, v - u_{\mathcal{Q}}) - \langle b, v - u_{\mathcal{Q}} \rangle + j_{\mathcal{Q}}(v) - j_{\mathcal{Q}}(u_{\mathcal{Q}}) \geq 0 \quad \forall v \in \mathcal{Q}.$$

Assuming a saturation assumption (as in Assumption 5.1) for this problem, we get that $\|u_{\mathcal{S}} - u_{\mathcal{Q}}\|$ is a reliable and efficient error estimator, cf. Theorem 5.2. $u_{\mathcal{Q}}$ can be computed by computing again the correction $d = u_{\mathcal{Q}} - u_{\mathcal{S}}$ using the defect problem

$$a(d, v - d) - \langle b - Au_{\mathcal{S}}, v - d \rangle + j(u_{\mathcal{S}} + v) - j(u_{\mathcal{S}} + d) \geq 0 \quad \forall v \in \mathcal{Q}.$$

Here, we used $a(u_{\mathcal{S}}, \cdot) =: \langle Au_{\mathcal{S}}, \cdot \rangle$.

Of course, just as in the linear case, we do not want to compute $u_{\mathcal{Q}}$ in the large space $\mathcal{Q}$ to obtain the error estimator. Instead, we once more want to solve only an easier (preconditioned) problem in the extension space $\mathcal{V}$. When we replace the bilinear form $a(\cdot, \cdot)$ by the preconditioner $b(\cdot, \cdot)$ as the in the preceding section for the linear problem, we get by the same arguments as before that $\mathcal{S}$ and $\mathcal{V}$ decouple in the *quadratic part*. This gives the following problem in the extension space:

$$\tilde{d} \in \mathcal{V}: \qquad b(\tilde{d}, v - \tilde{d}) - \langle b - Au_{\mathcal{S}}, v - \tilde{d} \rangle + j_{\mathcal{V}}(u_{\mathcal{S}} + v) - j_{\mathcal{V}}(u_{\mathcal{S}} + \tilde{d}) \geq 0 \quad \forall v \in \mathcal{V}.$$

$\tilde{d}$ can now be used as an error estimator as explained in the previous section. However, note that we are ignoring the coupling between the spaces $\mathcal{S}$ and $\mathcal{V}$ which is induced through the nonlinearity $j_{\mathcal{Q}}$. Indeed, we are just optimizing in the nodes $\mathcal{N}_{\mathcal{Q}} \setminus \mathcal{N}_{\mathcal{S}}$

which correspond to $\mathcal{V}$, omitting the fact that the values of $u_{\mathcal{S}}$ on the nodes $\mathcal{N}_{\mathcal{S}}$ which were optimal in $\mathcal{S}$ might not be the right values for $u_{\mathcal{Q}}$ on these nodes. In fact, it can be shown that ignoring this coupling can lead to the loss of reliability of the error estimator [76]. On the other hand, numerical experiments show that the estimator still works well for many problems [76]. In [77, 116], modified error estimates for the obstacle problem were derived which take into account missing terms (and show that indeed they are of higher order) such that the error estimators are reliable and efficient.

5.1.1. Hierarchical Error Estimators with Interior Penalty DG

Now, we want to discuss how we can translate the concept of hierarchical error estimates to our DG setting. While the abstract idea of having a discrete space $\mathcal{S}$ and a larger space $\mathcal{Q}$ is not tied to having piecewise linear or even continuous finite elements, most of the cited results so far assume these in one way or another. In particular, it will turn out that there are aspects to it which made us adopt a related, yet different approach to compute an approximation to the error estimator $u_{\mathcal{Q}} - u_{\mathcal{S}}$. Instead of using a diagonal scaling to compute the correction in the extension space $\mathcal{V}$, we will instead compute an approximation to the defect $u_{\mathcal{Q}} - u_{\mathcal{S}}$ directly in the extended space $\mathcal{Q}$ while making sure that the method is still reasonably computationally efficient.

Remark 5.7. Some of the algorithmic techniques referenced in this chapter (e. g. matrix-free evaluation of the differential operator) have only been implemented for $\mathcal{Q}^k$ elements, i. e. those based on cube reference elements, cf. Chapter 7. The mathematical results presented here, however, hold for general P^k elements.

Reliability and Efficiency

Before we consider problems that are more involved than the Poisson problem, we discuss how an approach similar to Example 5.3.1 can be adopted to Interior Penalty on discontinuous Galerkin spaces with varying order. Let $\mathcal{S}$ be a given DG finite element space. Consider the discrete solution $u_{\mathcal{S}}$ to some PDE problem discretized by a Symmetric Interior Penalty method, e. g. the Poisson problem

$$u_{\mathcal{S}} \in \mathcal{S} : a_{\mathcal{S}}\left(u, v\right) = F_h(v) \quad \forall v \in \mathcal{S}.$$

We still consider a larger space $\mathcal{Q} \supset \mathcal{S}$ to estimate the discretization error $|||u - u_{\mathcal{S}}|||$ by considering $|||u_{\mathcal{Q}} - u_{\mathcal{S}}|||$ with $u_{\mathcal{Q}}$ being the solution to

$$u_{\mathcal{Q}} \in \mathcal{Q} : a_{\mathcal{Q}}\left(u, v\right) = F_h(v) \quad \forall v \in \mathcal{Q}.$$

Remark 5.8. Many of the objects that will be considered in this section depend on the respective discrete space. We will therefore use sub- or superscripts $\mathcal{S}$ and $\mathcal{Q}$ to highlight this connection but may abstain from explicitly defining the space-dependent objects where appropiate. For example, we have that $a_{\mathcal{S}}\left(\cdot, \cdot\right) = a_h\left(\cdot, \cdot\right)$ is the SIPG bilinear form with respect to the grid, polynomial degree distribution and penalty function corresponding to the finite element space $\mathcal{S}$.

Possible choices to derive an extended space $\mathcal{Q}$ from $\mathcal{S}$ are to increase the local polynomial degree by using a polynomial degree $p_{\mathcal{Q}}(K) > p_{\mathcal{S}}(K)$, a refinement of the underlying grid $\mathcal{T}_{\mathcal{S}}$ to $\mathcal{T}_{\mathcal{Q}}$, or both methods.

Assumption 5.9. *Let $\mathcal{S}$ be the DG space defined on a grid $\mathcal{T}_{\mathcal{S}}$ and polynomial degree distribution $p_{\mathcal{S}} : \mathcal{T}_{\mathcal{S}} \to \mathbb{N}$. Then, the extended space $\mathcal{Q} \supset \mathcal{S}$ is defined on a grid $\mathcal{T}_{\mathcal{Q}}$ with degree distribution $p_{\mathcal{Q}} : \mathcal{T}_{\mathcal{Q}} \to \mathbb{N}$. On every element $K_{\mathcal{Q}} \in \mathcal{T}_{\mathcal{Q}}$, one of the following conditions holds:*

1. *$K_{\mathcal{Q}} \in \mathcal{T}_{\mathcal{S}} \cap \mathcal{T}_{\mathcal{Q}}$ and $p_{\mathcal{Q}}(K_{\mathcal{Q}}) > p_{\mathcal{S}}(K_{\mathcal{Q}})$ ("p-refinement"),*
2. *$K_{\mathcal{Q}} \in \mathcal{T}_{\mathcal{Q}} \setminus \mathcal{T}_{\mathcal{S}}$ was obtained through a refinement of a unique father-element $K_{\mathcal{S}} \in \mathcal{T}_{\mathcal{S}}$ ("h-refinement") and $p_{\mathcal{Q}}(K_{\mathcal{Q}}) \geq p_{\mathcal{S}}(K_{\mathcal{S}})$.*

Recall from Chapter 3, equation (3.24), the definition of the DG-norm

$$|||v|||^2 = |v|_1^2 + \sum_{e\in\Gamma} \sigma_e \int_e [\![v]\!]^2 \,\mathrm{d}S.$$

Note that for Interior Penalty methods, the norm of the discrete spaces will depend on the grid (due to the integrals which are evaluated on the boundary between grid elements) and on the penalty factor respectively the polynomial degree (through the $\mathcal{O}(p^2)$ scaling of the penalty factor). Therefore, we will introduce subscripts $|||\cdot|||_{\mathcal{S}}$ and $|||\cdot|||_{\mathcal{Q}}$ when appropriate. As the penalty constant should scale roughly as $\mathcal{O}(p^2)$ [94], we assume the penalty constant in $\mathcal{Q}$ on a given face should not be smaller than the corresponding penalty constant in $\mathcal{S}$:

Assumption 5.10. *Let $e^{\mathcal{Q}}$ be a face in $\mathcal{T}_{\mathcal{Q}}$, i. e. $e^{\mathcal{Q}} \in \Gamma_{\mathcal{Q}}$. For any*

$$e^{\mathcal{S}} \in \left\{ e \in \Gamma_{\mathcal{S}} : e^{\mathcal{Q}} \subseteq e \right\},$$

we have

$$\sigma^{\mathcal{S}}_{e^{\mathcal{S}}} \leq \sigma^{\mathcal{Q}}_{e^{\mathcal{Q}}}.$$

Finally, we have to state a modified version of the saturation assumption.

Assumption 5.11. *Let $u \in H^1(\Omega)$ be the analytic solution to the given problem and $u_{\mathcal{S}}$ and $u_{\mathcal{Q}}$ be its discrete solutions in $\mathcal{S}$ and $\mathcal{Q}$, respectively. Then, we assume there is a $\beta < 1$ such that*

$$|||u - u_{\mathcal{Q}}|||_{\mathcal{Q}} \leq \beta |||u - u_{\mathcal{S}}|||_{\mathcal{S}}.$$

We can relate the different norms of the discretization error $u - u_{\mathcal{S}}$ through the following equivalence lemma.

Lemma 5.12. *For $v \in H^1(\Omega)$ and $v_h \in \mathcal{S} \subset \mathcal{Q}$, we have*

$$\begin{aligned} |||v - v_h|||_{\mathcal{S}}^2 &\leq |||v - v_h|||_{\mathcal{Q}}^2, \\ |||v - v_h|||_{\mathcal{Q}}^2 &\leq C(\sigma^{\mathcal{S}}, \sigma^{\mathcal{Q}}) |||v - v_h|||_{\mathcal{S}}^2. \end{aligned}$$

Proof. For the full argument, but with a different norm, see [15, Lemma 14].

In our case, we have that the first equation is a direct consequence of Assumption 5.10. For the second equation, we can bound the difference in penalty constants of the jump terms by

$$C(\sigma^{\mathcal{S}}, \sigma^{\mathcal{Q}}) = \max_{e \in \Gamma \mathcal{Q}} \frac{\sigma_e^{\mathcal{Q}}}{\sigma_e^{\mathcal{S}}} \geq 1.$$

□

Example 5.12.1. For a typical choice of the penalty constant which is a constant times the square of the local polynomial degree, we have $C(\sigma^{\mathcal{S}}, \sigma^{\mathcal{Q}}) \leq 4$ if the polynomial degree was increased and $C(\sigma^{\mathcal{S}}, \sigma^{\mathcal{Q}}) = 1$ if only the grid was refined.

Similarly to the continuous finite element case, we can deduce from the saturation assumption that the difference of $u_{\mathcal{Q}}$ and $u_{\mathcal{S}}$ is again a suitable error estimator. The following two theorems also appear in [15].

Theorem 5.13. *Let Assumption 5.11 hold and let $\mathcal{Q}$ be obtained from $\mathcal{S}$ by h- or p-refinement. Then it holds for $\eta_{\mathcal{Q}} = |||u_{\mathcal{Q}} - u_{\mathcal{S}}|||_{\mathcal{Q}}$,*

$$\frac{1}{\sqrt{C(\sigma^{\mathcal{S}}, \sigma^{\mathcal{Q}})} + \beta} \eta_{\mathcal{Q}} \leq |||u - u_{\mathcal{S}}|||_{\mathcal{S}} \leq \frac{1}{1-\beta} \eta_{\mathcal{Q}}. \tag{5.12}$$

Proof. For the upper bound, apply the triangle inequality and insert the saturation assumption.

The proof for the lower bound is almost identical to the standard case where the same norms are used everywhere. Additionally, here one has to use Lemma 5.12 to relate the different norms.

$$\begin{aligned}
|||u_{\mathcal{Q}} - u_{\mathcal{S}}|||_{\mathcal{Q}} &= |||u_{\mathcal{Q}} - u + u - u_{\mathcal{S}}|||_{\mathcal{Q}} \\
&\leq |||u_{\mathcal{Q}} - u|||_{\mathcal{Q}} + |||u - u_{\mathcal{S}}|||_{\mathcal{Q}} \\
&\leq \beta |||u_{\mathcal{S}} - u|||_{\mathcal{S}} + \sqrt{C(\sigma^{\mathcal{S}}, \sigma^{\mathcal{Q}})} |||u - u_{\mathcal{S}}|||_{\mathcal{S}}. \\
&= \left(\beta + \sqrt{C(\sigma^{\mathcal{S}}, \sigma^{\mathcal{Q}})}\right) |||u - u_{\mathcal{S}}|||_{\mathcal{S}}.
\end{aligned}$$

□

As one can see, the different norms on the discrete spaces introduce a constant into the equivalence relation (5.12). This is because the coarse norm $|||\cdot|||_{\mathcal{S}}$ is not suitable for functions from $\mathcal{Q}$ if any grid refinement happened. Functions from the finer space $\mathcal{Q}$ created by grid refinements may have discontinuities across the new faces that will not be captured by the coarse norm $|||\cdot|||_{\mathcal{S}}$ because it lacks the appropriate face integral terms. For a pure *p*-refinement, however, no new faces are introduced and therefore $|||\cdot|||_{\mathcal{S}}$ is also a suitable norm on $\mathcal{Q}$. Since $H^1(\Omega)$ functions do not contribute to the face terms and due to the smaller penalty constant (cf. Assumption 5.10), we can evaluate functions from $H^1(\Omega) + \mathcal{Q}$ in the $|||\cdot|||_{\mathcal{S}}$-norm and in particular, we have

$$|||u - u_{\mathcal{Q}}|||_{\mathcal{S}} \leq |||u - u_{\mathcal{Q}}|||_{\mathcal{Q}}. \tag{5.13}$$

In that case, we can derive a simpler version of the error estimator because we can evaluate $|||u_{\mathcal{S}} - u_{\mathcal{Q}}|||$ now in the $|||\cdot|||_{\mathcal{S}}$-norm instead of the $|||\cdot|||_{\mathcal{Q}}$-norm.

Theorem 5.14. *Let $\mathcal{Q}$ be obtained from $\mathcal{S}$ by increasing in the polynomial degree $p_{\mathcal{S}}$ while leaving the grid $\mathcal{T}_{\mathcal{S}}$ unchanged. Then, given Assumption 5.11 holds, we have for $\eta_{\mathcal{S}} = |||u_{\mathcal{Q}} - u_{\mathcal{S}}|||_{\mathcal{S}}$ the following equivalence:*

$$\frac{1}{1+\beta}\eta_{\mathcal{S}} \leq |||u - u_{\mathcal{S}}|||_{\mathcal{S}} \leq \frac{1}{1-\beta}\eta_{\mathcal{S}}. \tag{5.14}$$

Proof. Clearly, (5.13) implies that we can state the saturation assumption in the form

$$|||u - u_{\mathcal{Q}}|||_{\mathcal{S}} \leq \beta |||u - u_{\mathcal{S}}|||_{\mathcal{S}}.$$

Because we are now also allowed to evaluate $|||u_{\mathcal{Q}} - u_{\mathcal{S}}|||_{\mathcal{S}}$, we can apply the same arguments as in Theorem 5.2 to derive (5.14). □

As we can see, the theoretic foundation for hierarchical error estimates using Interior Penalty methods is almost the same as for the classical, continuous Galerkin case.

Remark 5.15. In the following, we will denote the error estimator $|||u_{\mathcal{Q}} - u_{\mathcal{S}}|||$ by η, where the $\mathcal{S}$-norm is used if possible (i. e. if $\mathcal{Q}$ was obtained via p-refinement) and the $\mathcal{Q}$-norm otherwise.

Approximation through Preconditioning

Similarly as for the case with piecewise linear, continuous finite elements, computing the full solution $u_{\mathcal{Q}}$ on the finer space would be prohibitively expensive. As a remedy, we will introduce a preconditioning technique that is based on subspace splittings as used e. g. in [6, 7, 72].

First, note that the appealing structure of hierarchical extensions (say, e. g. quadratic bubble functions) are not easily generalized to higher order ansatz functions.

Example 5.15.1. To illustrate this, consider a space $\mathcal{Q}$ which was obtained by a p–$(p+1)$ refinement of a DG space $\mathcal{S}$ on a partition $\mathcal{T}$ of a one-dimensional domain $\Omega = (a, b)$. Since ultimately our goal is to solve variational inequalities (in contrast to linear variational problems), it is reasonable to require that $\mathcal{Q}$'s elementwise basis functions are again Lagrange polynomials based on a quadrature rule (see Chapter 3 and in particular Lemma 3.39). Say for a given element $K \in \mathcal{T}$, the local degree is $p > 1$ and the local basis functions $\{\phi_i\}_i$ are Lagrange polynomials based on the Gauss–Lobatto nodes $\mathbb{X}_K = \{x_1, \ldots, x_{p+1}\}$ with $p+1$ nodes. If we wanted to generalize the concept of hierarchical extensions, we had to pick another node $x_{p+2} \in K$ and construct the Lagrange basis function

$$\ell_{p+1}(x) = \prod_{i=1}^{p+1} \frac{x - x_i}{x_{p+2} - x_i}.$$

By a dimension argument, it's easy to see that $\operatorname{span}(\{\phi_i\}_i \cup \{\ell_{p+1}\}) = \mathcal{Q}^{p+1}(K)$. However, we also required that the quadrature weights are positive, i. e. $\int_K \ell_{p+1} \,\mathrm{d}x > 0$, which is not the case:

Since the Gauss–Lobatto quadrature rule can integrate rules up to order $2p-1$ exactly and we required $p > 1$, ℓ_{p+1} can be integrated using this quadrature rule.

$$\int_K \ell_{p+1}(x) \,\mathrm{d}x = \sum_{i=1}^{p+1} \ell_{p+1}(x_i)\omega_i = 0$$

by the Lagrange construction of ℓ_{p+1}.

Example 5.15.1 shows that hierarchical extensions based on Lagrange functions are not advisable if the increase in p is small. While there are techniques to enrich quadrature rules by additional nodes (see, e. g. Gauss–Kronrod rules) these will inevitable lead to rather large jumps in the polynomial degree (for example, the finer space would need a polynomial degree of at least $2p$ if Gauss–Lobatto nodes are used as illustrated in the preceding example). Moreover, these may not be readily available in many finite element software packages.

In summary, the classical (polynomial) hierarchical extension approach might not be a viable way to construct $\mathcal{Q}$, because it's not clear how to construct a basis for the hierarchical extension

$$\mathcal{V} = \mathcal{Q} \setminus \mathcal{S} \cup \{0\}$$

that preserves the quadrature rule structure for the basis of $\mathcal{Q}$. Therefore, we will drop the concept of an extension space $\mathcal{V}$ and consider preconditioning by applying an non-overlapping additive Schwarz method to the defect problem on the whole of $\mathcal{Q}$ instead. The main idea is to apply a hybrid Schwarz method (see, e. g. [109]) where our solution $u_{\mathcal{S}}$ on $\mathcal{S}$ acts as a coarse space solver and for the fine space $\mathcal{Q}$, one applies local solvers on the subspaces spanned by the basis functions grouped by their associated elements. Since the solution on the coarse space is known, only the latter part has to be computed. Algebraically, this corresponds to the application of a block Jacobi method if using a pure p-refinement. Such an approach has been described in [72] for a DG method similar to Interior Penalty. There, the fine space $\mathcal{Q}$ is obtained through a positive number of uniform grid refinements of the *conforming* grid $\mathcal{T}_{\mathcal{S}}$ optionally amended by an increasing $p_{\mathcal{S}}$.

In the following, we will construct a similar method for Symmetric Interior Penalty DG methods together with the possibility of pure p-refinements without altering the grid. To this end, we will use the proof structure of [72, Theorem 4.1] paired with techniques also used in [7] for a Schwarz method on the same subspace splitting.

For simplicity of notation, we will only consider the case where $\mathcal{T}_{\mathcal{S}}$ is quasi-uniform and both $p_{\mathcal{S}}$ and $p_{\mathcal{Q}}$ are constant on their respective grid. While this is of course an unrealistic assumption in an hp-adaptive setting, we emphasize that the analysis can be translated to the general case using min- and max-operators at the appropriate places. To stay consistent with the notation used in [7], we consider

$$\begin{aligned}
H &= \max_{K\in\mathcal{T}_\mathcal{S}} \operatorname{diam}(K),\\
h &= \max_{K\in\mathcal{T}_\mathcal{Q}} \operatorname{diam}(K),\\
q &\equiv p_\mathcal{S},\\
p &\equiv p_\mathcal{Q}.
\end{aligned}$$

The following two lemmas are from [7]:

Lemma 5.16. *For every $v_\mathcal{Q} \in \mathcal{Q}$, there exists a $\mathcal{H}_\mathcal{S}(v_\mathcal{Q}) \in \mathcal{S}$, such that*

$$\left\|v_\mathcal{Q} - \mathcal{H}_\mathcal{S}(v_\mathcal{Q})\right\|_{L^2(\Omega)} \lesssim \frac{H}{q} |||v_\mathcal{Q}|||_\mathcal{Q}, \tag{5.15}$$

$$|v_\mathcal{Q} - \mathcal{H}_\mathcal{S}(v_\mathcal{Q})|_{H^1(\mathcal{T}_\mathcal{Q})} \lesssim |||v_\mathcal{Q}|||_\mathcal{Q}. \tag{5.16}$$

Proof. See [7, Lemma 5.1].

The basic idea of the proof is to find a suitable approximation $\mathcal{H}(v_\mathcal{Q})$ of $v_\mathcal{Q}$ in $H^1_0(\Omega)$ and to define $\mathcal{H}_\mathcal{S}(v_\mathcal{Q})$ as the $\mathcal{S}$-interpolation of that approximation, i. e.

$$\mathcal{H}_\mathcal{S}(v_Q) = \Pi_\mathcal{S}\mathcal{H}(v_\mathcal{Q}).$$

Then, interpolation and approximation results can be used to obtain (5.15)–(5.16). □

Another useful lemma from [7, Lemma 5.3] is the following trace inequality, which has been proven in [104].

Lemma 5.17. *Let $v_\mathcal{Q} \in \mathcal{Q}$, then it holds*

$$\begin{aligned}
\sum_{K\in\mathcal{T}_\mathcal{S}} \|v_\mathcal{Q}\|^2_{L^2(\partial K)} \lesssim & |v_\mathcal{Q}|_{H^1(\mathcal{T}_\mathcal{Q})} \|v_\mathcal{Q}\|_{L^2(\Omega)} + \frac{1}{H}\|v_\mathcal{Q}\|^2_{L^2(\Omega)} \\
& + \left(\sum_{K\in\mathcal{T}_\mathcal{S}} \sum_{e\in\Gamma^\mathcal{Q}_K} \left\|\sigma^{1/2}[\![v_\mathcal{Q}]\!]\right\|^2_{L^2(e)} \right)^{1/2} \|v_\mathcal{Q}\|^2_{L^2(\Omega)}.
\end{aligned} \tag{5.17}$$

Equipped with Lemma 5.16 and 5.17, we can construct an approximation γ of $u_\mathcal{Q} - u_\mathcal{S}$ that is equivalent in the $|||\cdot|||_V$-norm (with V being either $\mathcal{S}$ or $\mathcal{Q}$, see Remark 5.15) and therefore a reliable and efficient error estimator on its own.

Consider the elementwise subspace decomposition of $\mathcal{Q}$: For every $K \in \mathcal{T}_\mathcal{S}$, define

$$Q_K = \left\{ v|_K : v \in \mathcal{Q} \right\},$$

i. e. Q_K is the restriction of $\mathcal{Q}$ to the coarse element $K \in \mathcal{T}_\mathcal{S}$. Let $R_K : \mathcal{Q} \to Q_K$ be the restriction operator of $\mathcal{Q}$ to a single element $K \in \mathcal{T}_\mathcal{S}$,

$$R_K v = v|_K, \quad v \in \mathcal{Q}.$$

Analogously, let $R_K^\top : Q_K \to \mathcal{Q}$ be the prolongation defined by extending the function by zero outside K. Clearly, every $v \in \mathcal{Q}$ can be rewritten as

$$v = \sum_{K \in \mathcal{T}_\mathcal{S}} R_K^\top v_K, \tag{5.18}$$

where $v_K = R_K v$.

For every $K \in \mathcal{T}_\mathcal{S}$, we define the local norm on Q_K by

$$|||v|||_K = |||R_K^\top v|||, \quad v \in Q_K.$$

Lemma 5.18. *Let $v \in \mathcal{Q}$. Then, it holds*

$$|||v|||^2 \lesssim \sum_{K \in \mathcal{T}_\mathcal{S}} |||R_K v|||_K^2. \tag{5.19}$$

Proof. Since the bulk terms, the face integrals on $\partial\Omega$, and the integrals for faces in the inner of every coarse element K are equal, we only need to compare the integrals that are located on coarse faces.

Let $v \in \mathcal{Q}$ and consider a face $e \subset K^+ \cap K^-$ for $K^+, K^- \in \mathcal{T}_\mathcal{S}$. Summing over all K, the face integral $\|v\|_{L^2(e)}$ will be evaluated both on the K^+ and the K^- side. Applying the basic inequality $(a-b)^2 \leq 2a^2 + 2b^2$, we get

$$\left\|v^+\right\|_{L^2(e)}^2 + \left\|v^-\right\|_{L^2(e)}^2 \gtrsim \left\|v^+ - v^-\right\|_{L^2(e)}^2 = \left\|[\![v]\!]\right\|_{L^2(e)}^2.$$

Summing over all these faces, we get (5.19). □

For the coarse space $\mathcal{S}$, we define the prolongation $R_\mathcal{S}^\top : \mathcal{S} \to \mathcal{Q}$ to be the classical injection operator.

Remark 5.19. Since $\mathcal{S} \subset \mathcal{Q}$, we will simplify notation by dropping the prolongation operator where appropriate, e. g.

$$|||u_\mathcal{Q} - R_\mathcal{S}^\top u_\mathcal{S}|||_\mathcal{Q} = |||u_\mathcal{Q} - u_\mathcal{S}|||_\mathcal{Q}.$$

For every $K \in \mathcal{T}_\mathcal{S}$, we define the restriction of the global bilinear form $a_\mathcal{Q}(\cdot,\cdot)$ to an element $K \in \mathcal{T}_\mathcal{S}$ by

$$a_K(v, w) = a_\mathcal{Q}\left(R_K^\top v, R_K^\top w\right),$$

dropping the superscript $\mathcal{Q}$ for readability. Now, it is time to define the approximate error estimator $\gamma \in \mathcal{Q}$. To do so, we consider the following local defect problems

$$\gamma_K \in Q_K : a_K(\gamma_K, v) = F_h(R_K^\top v) - a_\mathcal{Q}\left(R_\mathcal{S}^\top u_\mathcal{S}, R_K^\top v\right) \quad \forall v \in Q_K. \tag{5.20}$$

Naturally, the global error function γ is defined through

$$\gamma = \sum_{K \in \mathcal{T}_\mathcal{S}} R_K^\top \gamma_K.$$

For the proof of the main result of this section, we need another technical lemma from [7], proven in [6]:

Lemma 5.20. *Let $v \in \mathcal{Q}$. Consider the decomposition of v into $v_K \in Q_K$, $K \in \mathcal{T}_\mathcal{S}$, as in* (5.18). *Then, it holds*

$$a_\mathcal{Q}(v,v) = \sum_{K \in \mathcal{T}_\mathcal{S}} a_K(v_K, v_K) + \sum_{\substack{K,\tilde{K} \in \mathcal{T}_\mathcal{S} \\ K \neq \tilde{K}}} a_\mathcal{Q}\left(R_K^\top v_K, R_{\tilde{K}}^\top v_{\tilde{K}}\right). \tag{5.21}$$

Additionally, we can estimate the cross terms by

$$\sum_{\substack{K,\tilde{K} \in \mathcal{T}_\mathcal{S} \\ K \neq \tilde{K}}} a_\mathcal{Q}\left(R_K^\top v_K, R_{\tilde{K}}^\top v_{\tilde{K}}\right) \lesssim |||v|||^2 + \overline{\sigma} \sum_{K \in \mathcal{T}_\mathcal{S}} \|v\|^2_{L^2(\partial K)}, \tag{5.22}$$

where

$$\overline{\sigma} = C_\sigma \max_{K \in \mathcal{T}_\mathcal{Q}} \frac{p_\mathcal{Q}(K)^2}{\operatorname{diam}(K)} \approx C_\sigma \frac{p^2}{h}$$

is an upper bound for the penalty terms of $\mathcal{Q}$.

We are now left to prove that the equivalence $|||\gamma||| \approx |||d||| = |||u_\mathcal{Q} - u_\mathcal{S}||| = \eta$ holds. First, just as in [72], we need to make an assumption on the penalty parameter:

Assumption 5.21. *Assume that the penalty parameters in $\mathcal{S}$ and $\mathcal{Q}$ are chosen such that we have*

$$a_\mathcal{S}(v,w) = a_\mathcal{Q}\left(R_\mathcal{S}^\top v, R_\mathcal{S}^\top w\right) \quad v, w \in \mathcal{S}.$$

In particular, Assumption 5.21 implies the Galerkin orthogonality

$$a_\mathcal{Q}\left(u_\mathcal{Q} - R_\mathcal{S}^\top u_\mathcal{S}, v\right) = 0 \quad \forall v \in \mathcal{S}. \tag{5.23}$$

Remark 5.22. Assumption 5.21 implies that the penalty factors of $\mathcal{S}$ and $\mathcal{Q}$ have to be carefully calibrated to account for the smaller face volumes or higher polynomial degrees of $\mathcal{Q}$. Since $\mathcal{Q}$ typically requires a higher penalty term than $\mathcal{S}$ to be positive definite (depending on the increase in p), we might need to employ a higher than necessary penalty constant for $\mathcal{S}$, which in turn might degrade the numerical condition of the algebraic system in $\mathcal{S}$. In our computations, however, we experienced satisfying performance of the error estimator without ensuring Assumption 5.21, see also Section 7.1.4. Therefore it might be possible to weaken Assumption 5.21.

Now, we can prove the following theorem which is based on the proof structure of [72, Theorem 4.1] and the tools used in [7].

Theorem 5.23. *Let Assumption 5.21 hold. Then, we have*

$$|||\gamma||| \lesssim |||u_\mathcal{Q} - u_\mathcal{S}||| \lesssim C_\sigma \frac{H}{h} \frac{p^2}{q} |||\gamma|||. \tag{5.24}$$

Proof. For convenience, we define $d := u_Q - R_\mathcal{S}^\top u_\mathcal{S} \in \mathcal{Q}$.

First, we observe that by (5.20), we have

$$a_K(\gamma_K, v) = a_{\mathcal{Q}}\left(d, R_K^\top v\right) \quad \forall v \in Q_K. \tag{5.25}$$

Together with the continuity of $a_{\mathcal{Q}}(\cdot,\cdot)$, this implies

$$\sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K(\gamma_K, \gamma_K) = a_{\mathcal{Q}}(d, \gamma) \lesssim |||d|||\,|||\gamma|||. \tag{5.26}$$

On the other hand, a_K inherits the coercivity from $a_{\mathcal{Q}}$ and therefore

$$\sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K(\gamma_K, \gamma_K) \gtrsim \sum_{K \in \mathcal{T}_{\mathcal{S}}} |||\gamma_K|||_K^2 \gtrsim |||\gamma|||^2,$$

where the last inequality is due to (5.19). Combining both estimates, we get

$$|||\gamma||| \lesssim |||d|||, \tag{5.27}$$

which is the first part of (5.24).

For the other direction, consider $d_{\mathcal{S}} := \mathcal{H}_{\mathcal{S}}(d)$ as defined in Lemma 5.16. Since $d_{\mathcal{S}} \in \mathcal{S}$, we can apply the Galerkin orthogonality (5.23) and get, by applying (5.25) again,

$$\sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K\left(\gamma_K, R_K(d - d_{\mathcal{S}})\right) = a_{\mathcal{Q}}(d, d - d_{\mathcal{S}}) = a_{\mathcal{Q}}(d, d).$$

Consequently, it holds $|||d|||^2 \lesssim \sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K\left(\gamma_K, R_K(d - d_{\mathcal{S}})\right)$. Since $\sum_K a_K(\cdot,\cdot)$ induces a scalar product, we get by the Cauchy–Schwarz inequality

$$\begin{aligned}&\sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K\left(\gamma_K, R_K(d - d_{\mathcal{S}})\right) \leq \\ &\left(\sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K(\gamma_K, \gamma_K)\right)^{1/2} \left(\sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K\left(R_K(d - d_{\mathcal{S}}), R_K(d - d_{\mathcal{S}})\right)\right)^{1/2}.\end{aligned} \tag{5.28}$$

From (5.26) we have

$$\left(\sum_{K \in \mathcal{T}_{\mathcal{S}}} a_K(\gamma_K, \gamma_K)\right)^{1/2} \lesssim |||d|||^{1/2} |||\gamma|||^{1/2}. \tag{5.29}$$

It remains to estimate the second factor in the right hand side of (5.28). Employing

Lemma 5.20, we have

$$
\begin{aligned}
\sum_{K\in\mathcal{T}_{\mathcal{S}}} & a_K\left(R_K(d-d_{\mathcal{S}}),R_K(d-d_{\mathcal{S}})\right) \\
&= a_{\mathcal{Q}}\left(d-d_{\mathcal{S}},d-d_{\mathcal{S}}\right) - \sum_{\substack{K,\tilde{K}\in\mathcal{T}_{\mathcal{S}}\\ K\neq\tilde{K}}} a_{\mathcal{Q}}\left(R_K^\top v_K, R_{\tilde{K}}^\top v_{\tilde{K}}\right) \\
&\leq a_{\mathcal{Q}}\left(d-d_{\mathcal{S}},d-d_{\mathcal{S}}\right) + \left|\sum_{\substack{K,\tilde{K}\in\mathcal{T}_{\mathcal{S}}\\ K\neq\tilde{K}}} a_{\mathcal{Q}}\left(R_K^\top v_K, R_{\tilde{K}}^\top v_{\tilde{K}}\right)\right| \\
&\lesssim |||d-d_{\mathcal{S}}|||^2 + \overline{\sigma}\sum_{K\in\mathcal{T}_{\mathcal{S}}}\|d-d_{\mathcal{S}}\|^2_{L^2(\partial K)}\,. \qquad (5.30)
\end{aligned}
$$

Since $[\![d_{\mathcal{S}}]\!]=0$ on all interior faces that are not on the boundary of a coarse element, and using $\left\|v^+\right\|^2_{L^2(e)}+\left\|v^-\right\|^2_{L^2(e)}\gtrsim\left\|[\![v]\!]\right\|^2_{L^2(e)}$ (cf. proof of Lemma 5.18), we see that

$$
|||d-d_{\mathcal{S}}|||^2 \lesssim |||d|||^2 + |d-d_{\mathcal{S}}|^2_{H^1(\mathcal{T}_{\mathcal{Q}})} + \overline{\sigma}\sum_{K\in\mathcal{T}_{\mathcal{S}}}\|d-d_{\mathcal{S}}\|^2_{L^2(\partial K)}\,.
$$

Applying Lemma 5.16, we can further estimate (5.30) by

$$
|||d-d_{\mathcal{S}}|||^2 + \overline{\sigma}\sum_{K\in\mathcal{T}_{\mathcal{S}}}\|d-d_{\mathcal{S}}\|^2_{L^2(\partial K)} \lesssim |||d|||^2 + \overline{\sigma}\sum_{K\in\mathcal{T}_{\mathcal{S}}}\|d-d_{\mathcal{S}}\|^2_{L^2(\partial K)}\,.
$$

By using the trace inequality from Lemma 5.17 and the estimates in Lemma 5.16, we get

$$
\begin{aligned}
\overline{\sigma}\sum_{K\in\mathcal{T}_{\mathcal{S}}}\|d-d_{\mathcal{S}}\|^2_{L^2(\partial K)} &\lesssim \overline{\sigma}\frac{H}{q}(1+1/q)|||d|||^2 \\
&\lesssim C_\sigma\frac{H}{h}\frac{p^2}{q}|||d|||^2.
\end{aligned}
$$

Combining the preceding estimates, we derived the following bound:

$$
\sum_{K\in\mathcal{T}_{\mathcal{S}}} a_K\left(R_K(d-d_{\mathcal{S}}),R_K(d-d_{\mathcal{S}})\right) \lesssim C_\sigma\frac{H}{h}\frac{p^2}{q}|||d|||^2. \qquad (5.31)
$$

Inserting (5.29) and (5.31) into (5.28), we obtain

$$
\begin{aligned}
|||d|||^2 &\lesssim \sum_{K\in\mathcal{T}_{\mathcal{S}}} a_K\left(\gamma_K, R_K(d-d_{\mathcal{S}})\right) \\
&\lesssim \left(C_\sigma\frac{H}{h}\frac{p^2}{q}\right)^{1/2}|||\gamma|||^{1/2}|||d|||^{3/2},
\end{aligned}
$$

which implies the second estimate in (5.24) and therefore concludes the proof. □

Thus, we have shown that the approximate error estimator γ is equivalent to the error estimator $u_{\mathcal{Q}} - u_{\mathcal{S}}$ where the constants only depend on the penalty parameter and the amount by which $\mathcal{S}$ was refined to obtain $\mathcal{Q}$, cf. Assumption 5.9. Since all local corrections γ_K are mutually independent, the quantities can be computed by a fully parallel method where only small systems have to be solved. The right hand side of the defect problems and the evaluation of the energy norms can be obtained through a matrix-free approach (see, e. g., [79]) allowing us to compute γ without ever assembling the full stiffness matrix. For an efficient implementation of matrix-free operator evaluation (as done in, e. g., [80, 81, 88]) one might even consider solving the small systems corresponding to the coarse elements in a completely matrix-free way. This renders the approach much more feasible than a full computation of $u_{\mathcal{Q}} - u_{\mathcal{S}}$.

Extension to Variational Inequalities

In the previous section, we have shown how an approximation strategy based on subspace decomposition can be employed to obtain reasonable estimates of the exact solution in the extended space $\mathcal{Q}$. This proof was, however, based on a simple linear Poisson problem. Similar approaches to the hierarchical one described earlier have been applied to various variational inequality problems, in particular obstacle problems, before, cf. the elaboration in Section 5.1. It has to be noted, though, that due to the nonlinearity in the equations (say, e. g. an obstacle condition), the unknowns of $\mathcal{S}$ and $\mathcal{V}$ can become coupled. For the obstacle problem, simple counterexamples (demonstrated for example in [77]) show that ignoring this coupling may lead to an error estimator that is no longer reliable.

We will now demonstrate how the technique as described before for the linear problem can be employed for variational inequalities using again the obstacle problem as an example. Consider again a DG space $\mathcal{S}$ and an extension $\mathcal{Q}$. Let $u_{\mathcal{S}}$ and $u_{\mathcal{Q}}$ be the solution to the discretized obstacle problems in $\mathcal{S}$ and $\mathcal{Q}$, respectively, see also Chapter 3.3. Assuming again a saturation assumption as in 5.11, we have that $|||u_{\mathcal{Q}} - u_{\mathcal{S}}|||$ is a reliable and efficient error estimator, see also [15]. Note that the sets of nodes in which the obstacle conditions are controlled might be disjoint for a discretization in $\mathcal{S}$ and $\mathcal{Q}$, respectively. Instead of adding *additional* control nodes in $\mathcal{Q}$, we have in general a *different* set of nodes. Thus, a hierarchical approach similar to the continuous $\mathcal{P}^1$ case is once again not feasible. Instead, we will again solve independent, local subproblems to approximate $u_{\mathcal{Q}} - u_{\mathcal{S}}$. First, we have to specify the set of control nodes for the defect problem. Clearly, this is obtained by shifting the control set by $u_{\mathcal{S}}$:

$$\begin{aligned}&\mathcal{K}_{\mathcal{Q}} - u_{\mathcal{S}} = \\ &\left\{ v \in \mathcal{Q} : \underline{\psi}|_K(\hat{x}) - u_{\mathcal{S}}|_K(\hat{x}) \leq v|_K(\hat{x}) \leq \overline{\psi}|_K(\hat{x}) - u_{\mathcal{S}}|_K(\hat{x}) \forall K \in \mathcal{T}_{\mathcal{Q}} \forall \hat{x} \in \mathbb{X}_K^{\mathcal{Q}} \right\}.\end{aligned}$$

Therefore, we can state the global defect equation for $d = u_{\mathcal{Q}} - u_{\mathcal{S}}$:

$$\begin{aligned} d \in \mathcal{K}_{\mathcal{Q}} - u_{\mathcal{S}} : \quad a_{\mathcal{Q}}(d, v - d) &\geq r(v - d) \quad \forall v \in \mathcal{K}_{\mathcal{Q}} - u_{\mathcal{S}}, \\ r(v) &= F_h(v) - a_{\mathcal{Q}}(u_{\mathcal{S}}, v). \end{aligned}$$

For every $K \in \mathcal{T}_{\mathcal{S}}$, we denote the localized version of $\mathcal{K}_{\mathcal{Q}} - u_{\mathcal{S}}$ by

$$(\mathcal{K}_{\mathcal{Q}} - u_{\mathcal{S}})|_K = \left\{ v \in Q_K : \underline{\psi}|_K(\hat{x}) - u_{\mathcal{S}}|_K(\hat{x}) \leq v(\hat{x}) \leq \overline{\psi}|_K(\hat{x}) - u_{\mathcal{S}}|_K(\hat{x}) \forall \hat{x} \in \mathbb{X}_K^{\mathcal{Q}} \right\}.$$

This leads us directly to the approximate defect function $\gamma = \sum_K \gamma_K$ defined by the local defect problems on $K \in \mathcal{T}_{\mathcal{S}}$:

$$\gamma_K \in: (\mathcal{K}_{\mathcal{Q}} - u_{\mathcal{S}})|_K : a_K(\gamma_K, v_K - \gamma_K) \geq r(v_K - \gamma_K) \quad \forall v_K \in (\mathcal{K}_{\mathcal{Q}} - u_{\mathcal{S}})|_K. \tag{5.32}$$

In practice, these local problems could be solved using the TNNMG method or (given that the local problems are relatively small) by applying a number of steps of the nonlinear Gauss–Seidel method.

Remark 5.24. For discretized variational inequalities of the second kind (cf. Section 3.4), the approach would be very similar. We again localize the defect problems to the coarse elements and solve the variational inequalities (shifted by $u_{\mathcal{S}}$) arising in the small spaces, i. e. for each coarse $K \in \mathcal{T}_{\mathcal{S}}$, we solve for $\gamma_K \in Q_K$ such that

$$\begin{aligned} a_K(\gamma_K, v_K - \gamma_K) + \sum_i w_i \left(\Phi_i(v_K(x_i) + u_{\mathcal{S}}(x_i)) - \Phi_i(\gamma_K(x_i) + u_{\mathcal{S}}(x_i)) \right) \\ \geq r(v_K - \gamma_K) \quad \forall v_K \in Q_K. \end{aligned}$$

While this construction works well in practice (see also Chapter 7, in particular Section 7.1.4), we could not prove an equivalence relation between $|||d|||$ and $|||\gamma|||$ for discretized variational inequalities as we did for the linear problem. Heuristically one can argue that the defect d is of high frequency since it mostly contains parts of the solution that are not resolved by the coarser (yet possibly already high order) function space $\mathcal{S}$. Therefore, even simple preconditioners as the nonlinear additive Schwarz method induced by (5.32) and local "solvers" like nonlinear Gauss–Seidel methods can be expected to converge quickly. One can argue that the coarse scales are already resolved by $u_{\mathcal{S}}$, therefore no multigrid hierarchy as in a full TNNMG solver (see Chapter 4) is needed. Moreover, while a single application of a non-overlapping additive Schwarz might be insufficient to compute d accurately due lacking emphasize on inter-element continuity, the local solution process should at least lead to good local estimates, identifying elements with greater error reduction potential. Our numerical experiments show that usually it suffices to solve the local defect problems only approximately, say by applying a fixed number of nonlinear Gauss–Seidel steps.

5.2. Adaptive Algorithm

After having introduced an error estimator in the previous section, we will briefly sketch how it can be included in an *hp*-adaptive algorithm. The main idea is, after having computed the local and global error estimates, to use a marking algorithm to determine a set of elements where the ansatz space should be locally refined. Afterwards, on each marked element one needs to make a decision whether one wants to refine the grid ("h-refinement") or increase the local polynomial order ("p-refinement"). Naturally, one would like to employ a higher polynomial degree where the analytic solution is assumed to be smooth and use a finer mesh where it is not, see also Section 3.3 where this concept is formulated in terms of a priori estimates.

The abstract *hp*-adaptive algorithm can be described in the following way:

1. For a given DG space $\mathcal{S}$, compute discrete solution $u_{\mathcal{S}}$ for the given problem.
2. Compute local and global error estimates $\{\varepsilon_K\}_{K\in\mathcal{T}}$ and ε, respectively.
3. If ε is less than a required tolerance, STOP.
4. Mark a subset $\mathcal{T}^* \subset \mathcal{T}$ of grid elements where the DG space should be refined according to a marking strategy, cf. Section 5.2.1.
5. For every element $K \in \mathcal{T}^*$, decide, whether the DG space should be refined by splitting K into smaller elements or by increasing the local ansatz degree, cf. Section 5.2.2
6. Refine the space $\mathcal{S}$ and apply algorithm again.

We stress that the individual components, i. e. the marking strategy and the criterion for h- or p-refinement, can be chosen independently. In the following, we will briefly introduce the algorithms used for the numerical examples in this thesis.

5.2.1. Marking Strategy

A simple, yet very common marking strategy was introduced by Dörfler in [49]. We define the global error estimate (computed from local estimates ε_K as suggested in the previous section) by

$$\varepsilon_{\mathcal{T}}^2 = \sum_{K\in\mathcal{T}} \varepsilon_K^2,$$

and, analogously, for any subset $\mathcal{A} \subset \mathcal{T}$,

$$\varepsilon_{\mathcal{A}}^2 = \sum_{K\in\mathcal{A}} \varepsilon_K^2.$$

To reduce the (estimated) error by a given fraction, we choose a parameter $\theta \in [0,1]$ and define the set of admissible subsets of $\mathcal{T}$ by

$$\mathcal{N}_\theta = \left\{ \mathcal{A} \in 2^{\mathcal{T}} : \varepsilon_{\mathcal{A}} \geq (1-\theta)\varepsilon_{\mathcal{T}} \right\}.$$

Naturally, one is interested in a subset of grid elements from $\mathcal{N}_\theta$ which has minimal cardinality, i. e. we choose an $\mathcal{A}^* \in \mathcal{N}_\theta$ such that

$$|\mathcal{A}^*| = \min_{\mathcal{A}\in\mathcal{N}_\theta} |\mathcal{A}|. \tag{5.33}$$

While this set is in general not uniquely determined, we can computationally determine such a set by sorting the elements of $\mathcal{T}$ by their corresponding local error estimates ε_K. After a set $\mathcal{A}^*$ according to (5.33) has been chosen, we say that an element $K \in \mathcal{T}$ is *marked* if and only if $K \in \mathcal{A}^*$.

For more details on this marking strategy, we refer to [49].

5.2.2. *hp*-Refinement Criterion

As indicated earlier, we want to assess on each marked element whether increasing the order of the ansatz function or applying a grid refinement would lead to a more efficient error reduction. In our case, applying either strategy is particularly easy as discontinuous Galerkin discretizations allow for locally varying degrees or nonconforming grid refinements without having to ensure that finite element functions remain continuous. Thus, some aspects of the burden of implementing an *hp*-adaptive strategy are reduced compared to classical (continuous) finite element methods.

Conceptually, it is clear that if the underlying analytic solution is locally smooth, increasing the polynomial degree should decrease the error at a lower computational cost. While some a priori information about the solution and its regularity might be known (for example for the obstacle problem, we know that regularity might be low at the free boundary), in general we cannot predict the smoothness of the solution sufficiently. Thus, similarly to the error estimation case, we want to make educated guesses using the current state of the discrete solution. In the literature, there are several suggestions how to proceed in this way, see e. g. [86] for an overview. It is the author's impression that many of these suffer from at least one of the following problems:

1. They are tightly interwoven with the particular error estimator used (e. g. [85], where the local error estimates are compared against predicted error estimates), rendering them impractical for our rather general framework.

2. They require large additional problems to be solved (e. g. [45]) with no obvious way to localize computations (cf. the previous section on localizing the hierarchical error estimator). Thus, in particular for higher dimensions, theses approaches might be inefficient.

3. They require algorithms which are hard to implement and/or are rather costly (e. g. [40, 48], where not only the regularity is taken into account but also the amount of new degrees of freedom have to be optimized).

We will present two methods which have none of these drawbacks. The first is presented in more depth in [71]. There, also an overview about other existing methods is given. In fact, in [71], two new methods are introduced. The first one is based on judging whether a function is locally analytic by measuring the decay of the Legendre coefficients of the solution. The second method estimates the local Sobolev regularity index, i. e. the highest $k > 0$ such that the function under consideration is still in H^k. Consequently, if that k is larger than the local degree, the degree of the ansatz function can be increased.

In the following, we will at first outline the first approach from [71]. The same approach was used for the numerical methods for an obstacle problem in [15]. While the latter method sounds convincingly straightforward, the actual implementation is not. Meanwhile for obstacle type problems, we often know that the solution is locally either very smooth or has quite limited regularity. As indicated before, one seeks to judge if a function is (locally) analytic and therefore in C^∞. To this end, one estimates the size of a region around an element where a given function is still analytic using the fact that the Legendre coefficients vanish at an exponential rate for analytic functions [71].

We call the Bernstein ellipse with foci ±1 and radius $\rho = a_\rho + b_\rho$ (where a_ρ and b_ρ are the lengths of the semi-major and semi-minor axes, respectively) $\hat{\mathcal{E}}_\rho$, cf. [71]. Let v be a function $\mathbb{C} \to \mathbb{C}$ that is analytic in the interior of an ellipse $\hat{\mathcal{E}}_\rho$ with radius $\rho \geq 1$ containing the reference interval $(-1, 1)$ but not in an ellipse $\hat{\mathcal{E}}_{\rho'}$ of radius $\rho' > \rho$. Then, for the Legendre series

$$v(z) = \sum_{i=0}^{\infty} b_i L_i(z)$$

(where the L_i are the Legendre polynomials) it holds

$$\frac{1}{\rho} = \limsup_{i\to\infty} |b_i|^{1/i}, \tag{5.34}$$

cf. [71]. Values of ρ close to 1 indicate a small area of analyticity while large values correspond to a larger area where the function is analytic. Thus, the we have that $\theta = \frac{1}{\rho} \in [0, 1]$ is a criterion of local smoothness, where smaller values mean higher regularity. This approach can also be understood on more general (one dimensional) elements. There, ρ indicates the area of analyticity *relative to the given element* [71].

Of course, we do not know the Legendre coefficients $\{b_i\}_i$ for all $i \in \mathbb{N}$. However, if we have a discrete solution of order p, we have at least an approximation of the first $p + 1$ coefficients. For simple problems in 1D, these might even be exact, see [71, Remark 1]. Given an approximation of the Legendre coefficients

$$\left\{\tilde{b}_i\right\}_{i=0}^{p},$$

we deduce from (5.34)

$$|\tilde{b}_i| \approx |b_i| \sim (1/\rho)^i \quad \text{for } i \to \infty.$$

Applying the logarithm, we obtain $\log|\tilde{b}_i| \approx i\log(1/\rho)$ for $i \to \infty$. Since we have only a limited amount of coefficients available, we apply a linear regression to these to obtain a slope $m \in \mathbb{R}$ by assuming a linear relation of the form

$$|\log(|\tilde{b}_i|)| = im + b.$$

After having computed such an m by a least squares approach, we have

$$\theta = \rho^{-1} \approx e^m.$$

Finally, the decision whether to h- or p-refine is made by picking a parameter $\delta \in [0, 1]$. If $\rho \leq \delta$, the ansatz space should be (locally) p-refined while for larger values of ρ, grid refinement should be performed. Clearly, larger values of δ drive the hp-adaptive algorithm to favor p-refinement over h-refinement and vice versa. An extension of this approach for 2-dimensional $\mathcal{Q}^k$ functions is provided in [15].

The other promising approach we will briefly describe which was recently published is based on the use of continuous Sobolev embeddings. In [55, 114], it was observed that functions which are nonsmooth often exhibit steep gradients. By considering Sobolev embeddings like

$$\|u\|^2_{L^\infty(K)} \leq C \left(h^{-1} \|u\|^2_{L^2(K)} + h |u|^2_{1,K} \right) \tag{5.35}$$

for an H^1 function u on the real line (where $C = \coth(1)$, see [114]), one can define an alternative indicator

$$\theta = \|u\|^2_{L^\infty(K)} \left[C \left(h^{-1} \|u\|^2_{L^2(K)} + h |u|^2_{1,K} \right) \right]^{-1}. \tag{5.36}$$

Clearly, it holds $\theta \in [0, 1]$. Steep gradients will increase the denominator in (5.36) such that θ will be driven towards 0, while for flat gradients the ratio will be closer to 1.

Hoping that a discrete approximation v_h to a function v somehow resembles its features to a sufficient degree, we compute (5.36) for the discrete functions and pick again a threshold parameter δ, such that for values $\theta < \delta$ the grid is refined and for values $\theta \geq \delta$, the local polynomial degree is increased. For ansatz degrees $p > 1$, we take the $(p-1)$-th derivative to check if the p-th derivative has steep gradients.

Of course, equation (5.35) is only valid in one space dimension. An extension to $\mathcal{Q}^k$ elements for higher space dimensions was developed in [55]. There, H^2 functions have to be taken into account since H^1 is not continuously embedded in L^∞ for higher dimensions.

We emphasize once more that there are many more strategies to decide whether to choose h or p refinement and there is no definite answer which work best for which problems, yet. For an overview, see e. g. [86] and the cited literature. In our numerical experiments, we rely on the approach from [71] which estimates the analytical region by considering the Legendre coefficients as explained before.

6. Implementation Aspects

Before we test the strategies for obtaining efficient discretizations as described in the preceding chapters, we will briefly describe some technical aspects of the implementation. Due to the special structure of the algebraic problems arising from DG discretization, we were able to employ some non-standard techniques which were not readily available in the finite element codes we used.

The implementations are based on the C++ numerics framework DUNE ("Distributed Unified Numerics Environment") [18, 19, 20, 24, 101]. More precisely, all numerical experiments were performed using either DUNE releases 2.7., 2.8 or the `master` branch after 2.8.

DUNE emphasizes flexibility with respect to things such as the used data structures, algorithms, grid managers and other aspects where users might want to adopt the techniques most suited for their particular problem. This is mostly achieved by using generic interfaces through template code and type erasure. Consequently, after identifying areas where specialized data structures and algorithms promised efficiency gains, we integrated these without leaving the general framework or having to rework many other parts.

In the following, we will briefly describe which existing DUNE modules were used and afterwards where we went off the beaten paths.

6.1. Discretization in DUNE

Before we describe the particular route taken for our examples, we need to emphasize that there is no single representative approach on how to use DUNE to discretize PDEs. This is a natural consequence of the flexible and modular structure of DUNE. While there is a number of core modules which contain general data structures, algebraic solvers, shape functions, grid managers, geometry utilities and many other building blocks of any finite element code, the final act of combining them to obtain a fully-fledged PDE discretization is left to the user. To ease this task, several discretization modules exist, each with different strengths and emphasizes. Examples of these are DUNE-PDELab, DUNE-FEM [44], DUNE-FUFEM or more domain specific frameworks such as DuMux.

Since DUNE-FUFEM originated at Freie Universität Berlin, the author's research group traditionally relies on this framework and its siblings such as DUNE-Solvers, DUNE-Subgrid[66] and DUNE-TNNMG. A possible workflow in DUNE using these modules could be the following:

1. Generate a grid for the given domain using DUNE-Grid's interface and an appropriate grid manager.

2. Define a function space on this grid, e. g. using the function spaces (and respective bases) from DUNE-Functions [51, 52].

3. Choose a discretization scheme and assemble the resulting system, e. g. using DUNE-FUFEM for assembling and DUNE-ISTL for data structures.

4. Solve resulting algebraic systems using e. g. DUNE-Solvers and DUNE-TNNMG.

In addition to the mentioned modules, we developed a module DUNE-HPDG[1] which contains several additions for using *hp*-adaptive discontinuous Galerkin methods as described in this thesis. There we extended the ideas from the other modules for the *hp*-adaptive DG case. While some of the needed concepts were not present in the other modules (e. g. the appropriate function spaces), other aspects were optimizations built around certain assumptions in our models. Some of the additions will be described in the following section.

6.2. DUNE-HPDG

6.2.1. Function Spaces

As mentioned above, one of the first steps when discretizing a PDE with finite elements is to define a suitable finite-dimensional function space. In DUNE, a convenient way to do so is to use the function spaces offered by the DUNE-Functions module which offer a lot of flexibility with respect to creating tree-based bases and complex indexing schemes [52]. Several common function spaces are directly included such as e. g. the continuous Lagrange bases, the Taylor–Hood basis or the Raviart–Thomas basis. While there is also an implementation for a DG basis with Lagrange elements shipped, it is lacking two crucial features for our applications. First, the shape functions on the individual elements are with respect to equidistant nodes for $\mathcal{Q}^k$ elements. This would lead to basis functions which do not fulfill the crucial assumption of having a positive integral when using higher orders. As we have seen in Chapter 3, this would be a violation of a central assumption in our discretization scheme. Second, the polynomial order of the shape functions have to be set on a global level, thus not allowing for *p*-adaptivity where we want to set individual orders on different elements.

Since these limitations are not acceptable for our discretization, we set up our own DG function space using the framework provided by DUNE-Functions. This comes with the following modifications:

- We set up shape functions (also called “Local Finite Elements” in the DUNE context) using Lagrange functions based on quadrature rules such as Gauss–Lobatto, Gauss–Kronrod and Gauss–Legrende. Naturally, these basis functions have positive integrals.

[1] https://github.com/c1887/dune-hpdg

- The function space basis implementation allows to choose and change the degree of the shape function *per element* at runtime, thus allowing for having a p-adaptive algorithm.
- While the included DG basis uses a flat indexing scheme (i. e. every basis function has a single integer as index), we opted for a multi-index which emphasizes the DG structure: Each basis function is associated to a particular node on a *single* element. Our indexing scheme then reads

  $$(\text{Element Index}, \text{Local Index})$$

  assuming we have a flat indexing of all elements in the grid and some flat indexing of the nodes on each particular element. This indexing scheme requires almost no computation or manipulation of indices when computing global indices from a given local function on a particular element. In particular this allows for much simplified algorithms in many cases because one can safely assume a consecutive (local) indexing without having to query the global index in each step.

6.2.2. Data Structures

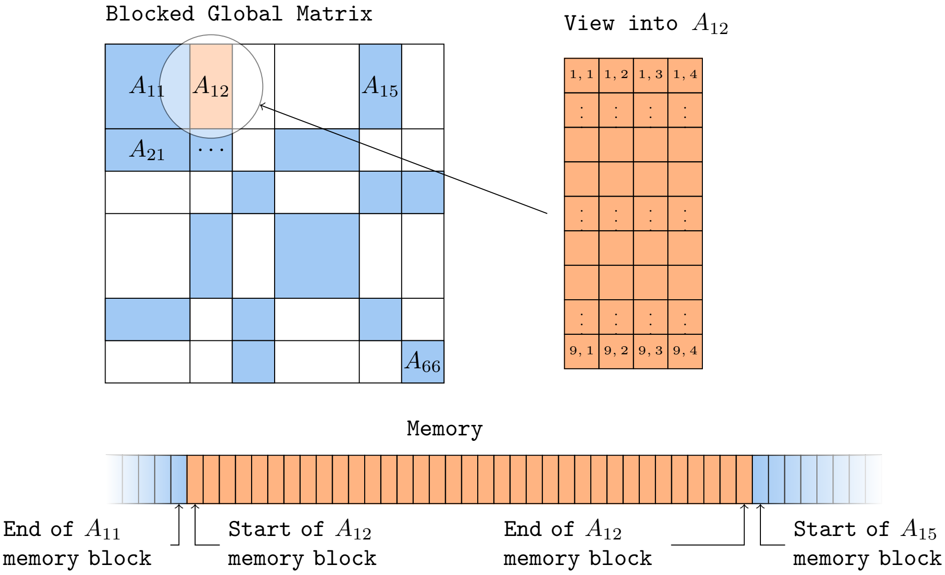
Inspired by the blocking of the indices described in the previous paragraph, we opted for a blocked version of the data structures such as vectors and matrices. Here, the individual entries will be addressed with the same index scheme as described above. The advantage of this becomes clear when one considers a typical stiffness matrix. There, in general every basis function associated to an element will couple (i. e. the corresponding matrix entry will be non-zero) with every other basis function on that element. Moreover, it will also have non-zero entries when paired with *all* basis function from *every* adjacent element. In particular for higher orders the number of non-zero entries for each basis function will therefore be relatively high. For typical sparse matrix formats such as Compressed Row Storage (CRS, see e. g. [99]), the data structure has to keep track of all these non-zero entries additionally to the actual *values* of the entries. In our DG setting, however, the structure of the non-zero entries is very regular. As explained earlier, we know that each local function will couple with most or all of the other functions on the particular element as well as the local functions from the neighboring elements. Hence, in a sparse matrix format, it suffices to know which elements are adjacent to which other elements. The resulting matrix is stored as a blocked CRS matrix. Since most of the local functions will couple with each other when the elements are adjacent (or the same), the individual blocks can be stored as small dense matrices. The vector data structures can analogously be stored in the same blocked fashion with small dense vectors as blocks.

Since these dense matrix and vector blocks can be multiplied in a straightforward and efficient fashion, we can expect fast arithmetic operations of the blocked data structures such as e. g. matrix-vector multiplications. Moreover, due to the much reduced storage requirements on the sparsity pattern, we observed a greatly reduced memory usage.

While DUNE-ISTL provides an implementation of the blocked CRS format (`Dune::BCRSMatrix`), the size of the matrix blocks must be static (and thus be chosen at compile time) and fixed for all elements. This is again not suitable for our *hp*-adaptive approach, where due to varying local orders the size of the individual matrix blocks must be controllable at run-time and on a per-element basis. While this could theoretically be achieved by using a dynamic matrix type as block type for the blocked CRS matrix, this approach has the disadvantage that the memory for the matrix blocks would be allocated individually and hence not be stored in a consecutive fashion. This is expected to induce drastic performance penalties.

As a remedy, we implemented a blocked CRS matrix (based on DUNE-ISTL's `BCRSMatrix`) which allows to control the individual block sizes at run-time while the underlying memory is stored consecutively. While the complete matrix data is stored as a single large memory block, the individual matrix blocks are accessed using views on their respective data chunks which act like normal matrix blocks. In particular, using pointers to the data chunks, it is possible to use external libraries like BLAS (Basis Linear Algebra Subprograms)[2] to perform fast arithmetic locally. Figure 6.2.1 illustrates our implementation. While the global matrix has to know the position *blocks* (corresponding to grid elements and their coupling, respectively) as well as their size, the *entries* are stored in a consecutive memory batch. Thus, in addition to the coordi-

Figure 6.2.1.: Illustration of the dynamically blocked CSR storage. White blocks are zero matrices.



[2] https://www.netlib.org/blas/

nates (the first index in our indexing as introduced earlier) and the size, the individual matrix blocks have also a pointer to the beginning of the corresponding memory block. When the block is accessed, a `MatrixWindow` is used, which is constructed using the memory pointer and the block size. Then, this window can be used as a regular dense matrix object. In particular, the local indexing of this matrix will always be flat and consecutive. This increases the locality as no additional calls (e. g. to the basis object or the sparsity pattern) have to be made in order to figure out global indices.

We also developed a corresponding vector format, which is quite similar to DUNE-ISTL's `VariableBlockVector` while having a slightly different interface as well as a set of utility functions that generates data structures such as matrices and vectors of appropriate size using the information provided by the basis.

6.2.3. Matrix-free Operators

When estimating the error in a larger space (cf. Section 5.1.1), we need to compute quantities in this space. Since we do not want to assemble full matrices in this space due to the increased costs, we perform several operations such as (nonlinear) smoothing and matrix-vector products on-the-fly, i. e without having the full matrix assembled at any stage. The strategy is that we iterate over all elements of the grid (possibly in parallel) and perform *local* operations on these. This can either be done by assembling only the needed local matrix block (e. g. when performing a block Jacobi method) or by computing the quantities directly. For example, when performing a matrix-vector product with the stiffness matrix consisting of a discretized Laplace operator in 2D, we implemented sum-factorized code which is faster than a full matrix-vector product due to reduced memory bottlenecks for higher orders. Moreover, the memory requirement is obviously much lower for all orders. More sophisticated and highly parallelized approaches are described e. g. in [80, 81, 88].

DUNE-HPDG offers the matrix-free infrastructure and several local operators which can be used for the algorithms described in this thesis. As a starting point, local assemblers from DUNE-FUFEM can also be used as local operators for a matrix-free matrix-vector product in this framework.

6.2.4. Multigrid Solver

While DUNE-SOLVERS offers a multigrid implementation, we chose to base our multigrid algorithm on an implementation called DUNE-PARMG[3] which allows for a parallel geometric multigrid algorithm. We extended the implementation such that also the p-multigrid part we described earlier is possible in parallel. Naturally, we also included the parallel ℓ_1–smoothers [14] (both nonlinear for the obstacle problem and linear) described in Section 4.3.

Moreover, we have transfer operators that work with non-uniform grids and also for the transfer to lower order spaces as described in Section 4.2. Here, we once more exploited the blocked structure and indices when using DG methods to implement simple yet efficient algorithms.

[3] Unfortunately, not publicly available at the time of writing.

7. Numerical Experiments

7.1. Obstacle Problem

In this section we will examine some of the methods introduced in the earlier chapters using numerical experiments for an elliptic obstacle problem as introduced in Chapter 2.2.1. All examples will be discretized using a Symmetric Interior Penalty DG method (SIPG) as described in Chapter 3. In particular, we will use Gauss–Lobatto bases and check the obstacle condition in the Lagrange nodes. Naturally, a heavy emphasis will be on the use of *hp*-adaptivity as introduced in Chapter 5. The arising algebraic problems will be solved by a TNNMG method using a multilevel linear solver built with a combination of *p*-multilevel and classical multigrid solvers, cf. Chapter 4.

To support our claim that the proposed methods are indeed working as expected, we start with a unilateral obstacle problem to which the analytical solution is known. The example was found in [16]:

Problem 7.1. *Consider* $\Omega = \left(-\frac{3}{2}, \frac{3}{2}\right)^2$. *Find* $u \in \mathcal{K}$ *such that*

$$a(u, v-u) \geq \ell(v-u) \quad \forall v \in \mathcal{K},$$

with $a(v,w) = (\nabla v, \nabla w)$ *and* $\ell(v) = \langle f, v\rangle$. *Here,* f *is the* L^2 *functional stemming from the* L^2 *product with the constant function* -2, *i. e.* $\langle f, \cdot\rangle = -2\int_\Omega \cdot \, \mathrm{d}x$.

The admissible functions are all H^1 *functions that satisfy the boundary condition* $v|_{\partial\Omega} = g$ *and which are greater or equal to zero almost everywhere, i. e.*

$$\mathcal{K} = \Big\{ v \in H^1 : v = g \text{ on } \partial\Omega, 0 \leq v < \infty \text{ a.e.} \Big\},$$

where $g(x) = \frac{|x|^2}{2} - \ln(|x|) - \frac{1}{2}$.

Theorem 7.2. *Problem 7.1 has the unique solution*

$$u(x) = \begin{cases} \frac{|x|^2}{2} - \ln(|x|) - \frac{1}{2} & \text{if } |x| \geq 1, \\ 0 & \text{else.} \end{cases}$$

Proof. For uniqueness, consider Theorem 2.6. It can be verified that u is indeed a solution to the given obstacle problem by a direct calculation. □

7.1.1. Discretization and Parameters

While the discretization has been described in detail in the earlier chapters, in particular Chapter 3, we will briefly recap the most important features.

As the domain Ω has a very regular structure, it will be (initially) discretized with a uniform structured grid $\mathcal{T}$ that consists of squares of equal size. Note that the grid might get refined nonconformingly in the adaptive process. If a grid element is marked for refinement, it will be subdivided into four new squares of equal size.

Our ansatz space is the DG space $V_{\mathcal{T}}^{p}$ consisting of the piecewise $\mathcal{Q}^k$ finite elements. Initially, the degree k is constant across all grid elements but might be locally adjusted during p-refinement.

The bilinear form $a(\cdot,\cdot) = (\nabla\cdot, \nabla\cdot)$ on the discrete space will be replaced by the SIPG form from (3.17),

$$\begin{aligned} a_h(v,w) = \sum_{K\in\mathcal{T}} (\nabla v, \nabla w)_K + \sum_{e\in\Gamma} &- \int_e \{\nabla v\} [\![w]\!] + \{\nabla w\} [\![v]\!] \,\mathrm{d}\mathcal{H}^{d-1} \\ &+ \frac{\sigma_e}{|e|} \int_e [\![v]\!][\![w]\!] \,\mathrm{d}\mathcal{H}^{d-1}. \end{aligned}$$

To emphasize that the penalty parameter might vary for different faces e, we denote it with σ_e. Let $p : \mathcal{T} \to \mathbb{N}$ be the function mapping grid elements to the degree of the respective polynomials on the given element have, i. e.

$$v|_K \in \mathcal{Q}^{p(K)} \quad \forall K \in \mathcal{T}.$$

Then, for a face $e = K_0 \cap K_1$, we have

$$\sigma_e = \tilde{\sigma} \cdot \max\left\{p(K_0), p(K_1)\right\}^2.$$

The value $\tilde{\sigma} > 0$ is fixed for all elements. We used a value of $\tilde{\sigma} = 1.6$ for the numerical examples in this section. Note that this construction of penalty parameters means that for different degree distributions p, the discretization is strictly speaking not the same. For example, for given distributions p and $\overline{p}$ with $\overline{p}(K) = p(K) + 1$, we have that

$$V_{\mathcal{T}}^{\overline{p}} \supset V_{\mathcal{T}}^{p},$$

yet the values of the bilinear forms might be different for given (coarse) functions from $V_{\mathcal{T}}^{p}$. In particular, for a quadratic energy involving $a_h(\cdot,\cdot)$, the global minimum might be lower for a bilinear form corresponding to $V_{\mathcal{T}}^{p}$ compared to the global minimum for the analog bilinear form of $V_{\mathcal{T}}^{\overline{p}}$ despite the latter being the greater set.

For the $\mathcal{Q}^k$ finite elements, we use Lagrange polynomials with respect to the Gauss–Lobatto nodes of the corresponding degree as basis functions. This has direct consequences for the discretization of the obstacle problem. We solve the problem

$$a_h(u_h, v - u_h) \geq \ell(v - u_h) \quad \forall v \in \mathcal{K}_{hp},$$

with $\mathcal{K}_{hp}$ being the set of functions from $V_{\mathcal{T}}^{p}$ whose *nodal values* (i. e. in the Gauss–Lobatto nodes) do not violate the nodal values of the obstacle function, see also Definition 3.35.

7.1.2. Convergence of Adaptive Algorithms

Ultimately, we are interested in approximating solutions to the given problems with the highest possible efficiency, i. e. obtaining a discrete solution which is close enough (in an appropriate norm) to the real solution while investing as little work as possible. An important statistic for this goal is of course the convergence rate, i. e. obtaining an exponent $c > 0$ such that $\|u - u_h\| \lesssim h^c$ with h being the maximal diameter of all grid elements and c being as large as possible. In the case of an adaptive refinement, however, the largest diameter h might be very large for some elements and therefore the convergence rate in h will not be the appropriate measure. Rather, we will investigate how the rate behaves with respect to the number of unknowns n, i. e. we are interested in an error reduction rate of the form $\mathcal{O}(n^{\tilde{c}})$. We want to investigate how convergence rates behave with respect to different adaptive and non-adaptive refinement strategies.

Fixed p, Uniform h Refinement

At first, we investigate the "classical" case where a fixed polynomial degree for all ansatz functions is chosen and the grid is successively uniformly refined. The grid is chosen to be a structured grid, dividing Ω into squares of equal size.

For a given (non-adaptively refined) grid, we expect $c = 1$ for piecewise linear DG elements and $c = (1.5 - \varepsilon)$ for higher order elements, cf. [112]. Figure 7.1.1 shows the error history of a DG-$\mathcal{Q}^k$ for varying values of $k \in \mathbb{N}$. All errors are computed in the SIPG norm

$$|||\cdot|||^2 = \|\cdot\|_1^2 + \sum_{e\in\Gamma} \int_e [\![\cdot]\!]^2 \,\mathrm{d}\mathcal{H}^1. \tag{7.1}$$

As the plot lines are almost parallel for $k > 1$, we conclude that indeed employing higher order elements does not lead to higher order convergence rates for the obstacle problem. We compute an approximation of the convergence rate $\tilde{c}$ by performing a linear regression for the equation

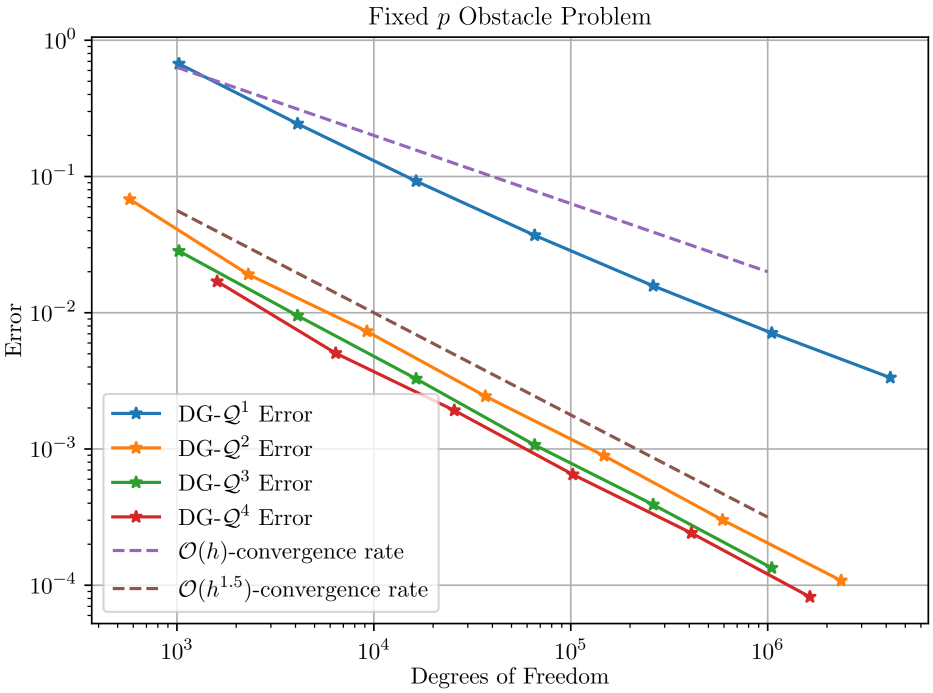
$$\ln(e_i) = \tilde{c}\ln(N_i) + b,$$

i. e. we compute the slope of the lines in Figure 7.1.1. Here, e_i is the computed error for N_i degrees of freedom for the respective degree. Since it holds $h \sim N^{-0.5}$, we obtain the convergence rate h^c by computing $c = -2\tilde{c}$. For better comparison with the methods that will be shown later, we also report $\tilde{c}$.

Indeed, Table 7.1.1 confirms the theoretical bound on the convergence rate for higher order elements as all values for $k > 1$ are close to 1.5.

Surprisingly though, the piecewise linear $\mathcal{Q}^1$ elements perform better than predicted for the given model problem.

Figure 7.1.1.: Convergence for fixed p and uniformly refined grid



Degree k	**Convergence Rate** c	**Convergence Rate** $\tilde{c}$
1	1.2755	-0.6377
2	1.5318	-0.7659
3	1.5448	-0.7724
4	1.5191	-0.7596

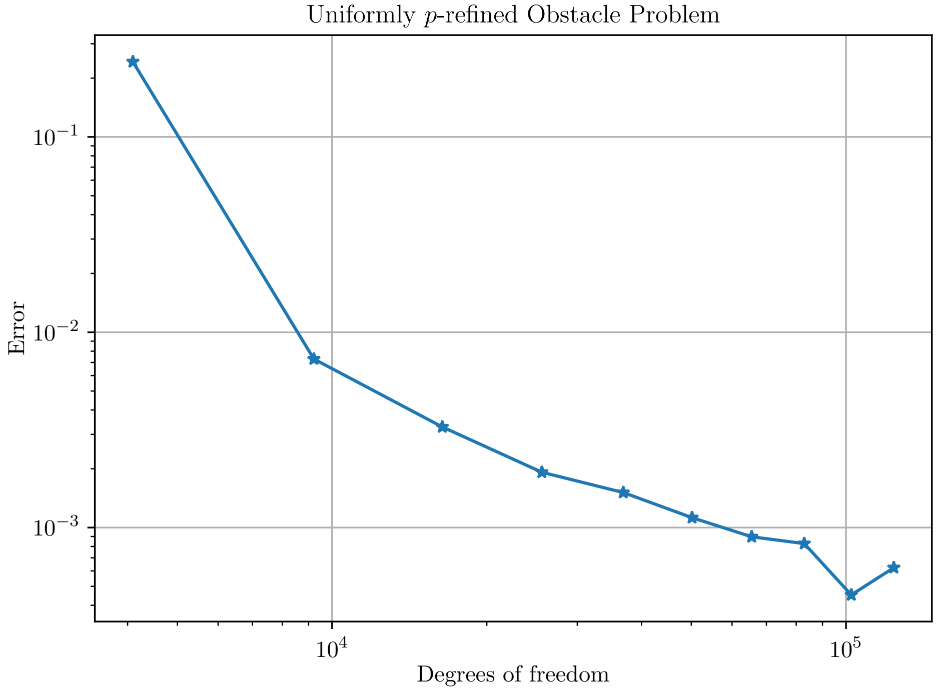
Table 7.1.1.: Ansatz Degree and Corresponding Convergence Rates

Fixed h, Uniform p Refinement

The next logical step in a non-adaptive setting is to perform what is sometimes called p-FEM, i. e. fixing a given grid (and thus h) and increasing the polynomial degree to increase accuracy. In [78] it has been proven that such a method converges eventually for continuous finite element spaces. The authors control the obstacle conditions in the Gauss–Lobatto nodes as well.

We will now numerically investigate the behavior for our DG discretization. Figure 7.1.2 suggests that while increasing the polynomial degree indeed decreases the error,

Figure 7.1.2.: Convergence for fixed grid and uniformly increased degree



the speed at which this happens decreases simultaneously, i. e. convergence slows down when using higher and higher ansatz degrees.

There is an outlier towards the end of the plotted line (at $k = 9$). The error value is actually lower than the following at $k = 10$ besides $\mathcal{Q}^{10}$ being the larger space. We attribute this to the fact that the discretizations do not coincide for different p: We use $\sigma_k = \overline{\sigma} k^2$ as penalty parameter for a fixed value of $\overline{\sigma}$ throughout all degrees k. While the optimal penalty value scales approximately as k^2, we do not optimize for penalty values independently. Hence, the specific value $\sigma_9 = \overline{\sigma} \cdot 9^2$ in SIPG might just be a better discretization for DG-$\mathcal{Q}^9$ than $\sigma_{10} = \overline{\sigma} \cdot 10^2$ is for DG-$\mathcal{Q}^{10}$. Moreover, even if we chose the same penalty value, we discretize the admissible set $\mathcal{K}$ by checking the obstacle condition in the Gauss–Lobatto nodes. Obviously, this will generate different sets $\mathcal{K}_{hp}$ for different values of p.

Fixed p, Adaptive h Refinement

In this section, we will numerically investigate how an adaptive DG discretization will perform. Performing grid refinement on continuous finite element spaces for obstacle problems using hierarchical error estimators has been thoroughly tested before, see

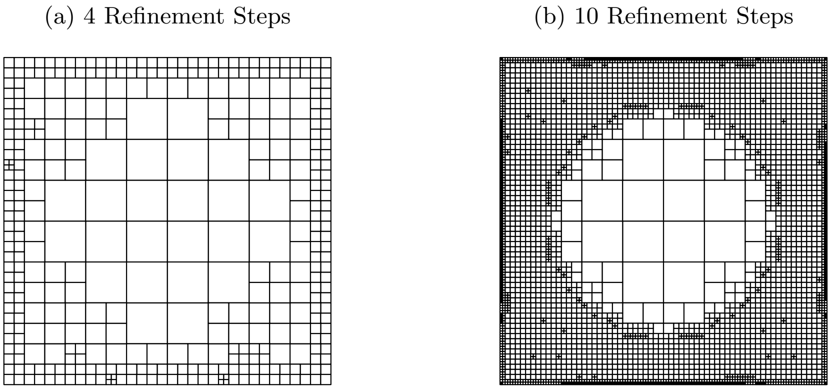


Figure 7.1.3.: Grid after Adaptive h-Refinement, $p = 1$.

e. g. [77]. The difference here is that we use discontinuous Galerkin spaces and the preconditioned error hierarchical estimator differs in the sense that we do not consider e. g. quadratic bubble functions as space enrichment but rather approximately solve on the full extended space, cf. Section 5.1.1.

The fine space $\mathcal{Q}$ in the hierarchical error estimator is constructed by increasing the local polynomial degree of each ansatz function in the coarse space $\mathcal{S}$ by one. Other choices (e. g. increasing by a higher number or refining the grid) are of course possible. Moreover, we allow adjacent elements to have a level difference of at most two, i. e. for each edge there are at most three hanging nodes. Technically, this is not necessary for DG discretizations. Indeed, we found the impact of this safety measure to be rather small for this problem because the adaptive algorithm will try do reduce the discontinuities arising from hanging nodes due to the penalty terms in the DG formulation.

Remark 7.3. Classical conforming adaptive refinements (e. g. red–green refinements) are not possible in our framework, since we restrict ourselves to $\mathcal{Q}^k$ elements which live on square reference elements.

The plots in Figure 7.1.3 show that for piecewise linear finite elements, the error estimator is able to correctly identify the contact region (which is the disc $\{x \in \mathbb{R}^2 : |x| = 1\}$) and refine the grid accordingly. Moreover, since the obstacle is constant, we have that the solution u is constant in the contact region. The adaptive algorithm acknowledges this fact by applying only very few refinements in the corresponding area (i. e. inside the disc). Outside the contact region, the grid is almost uniformly refined.

Albeit the solution being rotationally symmetric, we observe that the grid does not always represent this fact. This can be due to the fact that the discrete solution can only be obtained to a given accuracy when solving the discrete problem or because of rounding errors in the marking process. However, the non-symmetric parts are getting

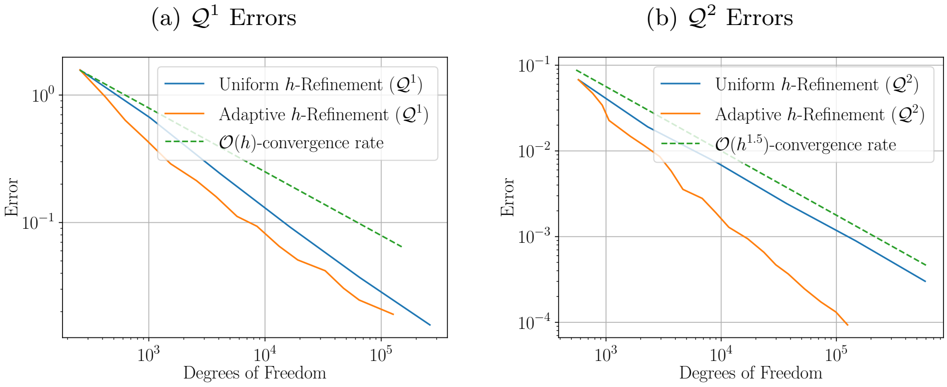


Figure 7.1.4.: Uniform versus Adaptive Refinement Errors

less severe the more refinement steps are applied.

We will now investigate how the convergence rate with respect to *the number of degrees of freedom*, n, behaves.

Remark 7.4. We do not compute the convergence rate with respect to the maximal mesh width since in an adaptive algorithm, some elements might be deliberately left large and thus the notion of h is skewed. As mentioned before, for convergence rates with respect to h on uniform grids, one can multiply the slope by -2.

Again, we computed the approximate slope of the error plot using a linear regression, i. e. we compute $\tilde{c}$ such that

$$e_i \approx n_i^{\tilde{c}}.$$

As observed earlier, Table 7.1.2 shows the piecewise linear elements exceed the theoretical expectation (namely $\mathcal{O}(h)$, i. e. $\tilde{c} = -0.5$) for this model problem, while for the $\mathcal{Q}^2$ elements the measured slope is very close to the theoretical value of -0.75.

For the piecewise linear elements, we observe that while the adaptive method needs less degrees of freedom to achieve a given error threshold, the *rate* at which it converges only slightly differs from the uniform refinement. This is not surprising as the parts of the solution that are not in contact with the obstacle have quadratic and logarithmic contributions and hence will always gain accuracy from refining the mesh when using piecewise linear elements. The contact region, however, which makes up about a third of the domain, is constant and can be approximated exactly even with a very coarse grid. Thus, the adaptive method can save about a third of the degrees of freedom the uniform refinement invests.

For the quadratic $\mathcal{Q}^2$ finite elements, we observe that the adaptive method actually leads to a higher convergence rate, as can be seen in Figure 7.1.4b and in Table 7.1.2. While we have not used the option of p-refining at this point, this already gives an outlook to truly hp-adaptive methods. Loosely speaking, the $\mathcal{Q}^2$ elements can often approximate the non-constant parts of the solution well enough even on rather coarse

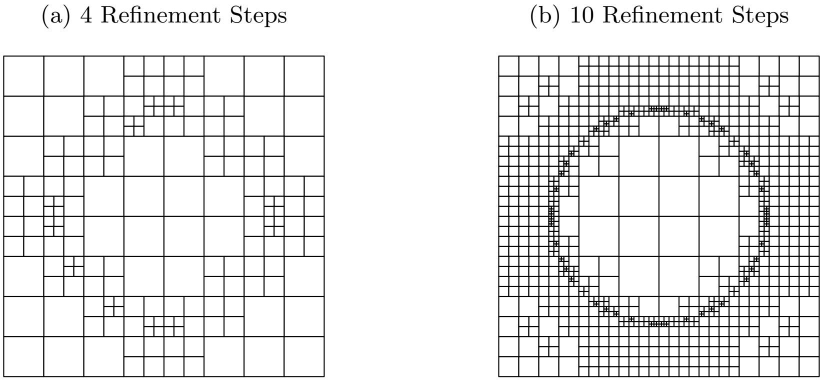


Figure 7.1.5.: Grid after Adaptive h-Refinement, $p = 2$.

elements. Meanwhile, the h-refinement can now “focus” on the nonsmooth parts at the free boundary as argued extensively in Section 3.3. This can also be seen in the exemplary plots in Figure 7.1.5.

Degree k	**Refinement Type**	**Convergence Rate $\tilde{c}$**
1	Uniform	-0.6729
1	Adaptive	-0.7072
2	Uniform	-0.7700
2	Adaptive	-1.2015

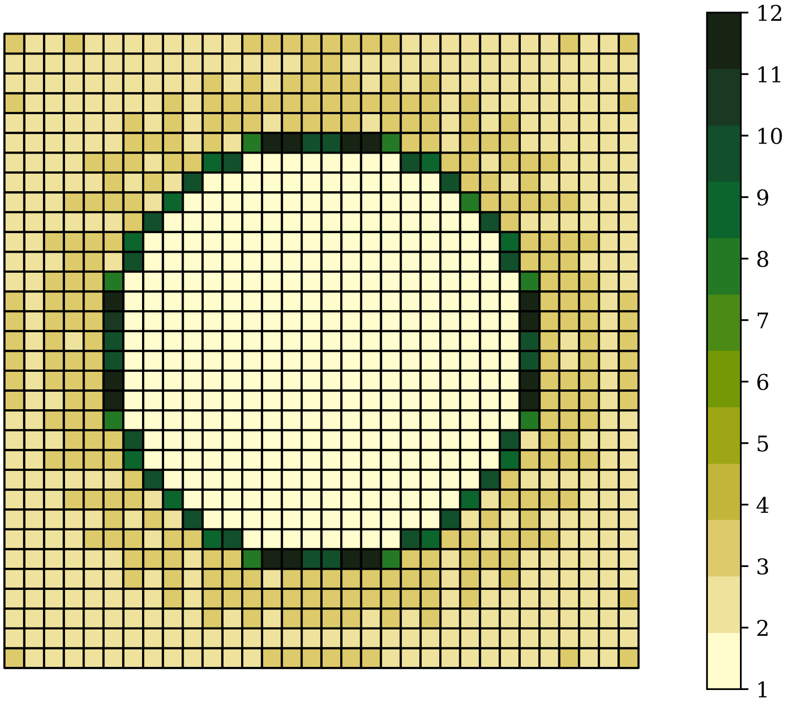
Table 7.1.2.: Convergence Rates for Uniform and Adaptive h-Refinement

Fixed h, Adaptive p Refinement

Following the scheme, we present the results for a fixed grid with optional adaptive p refinement. More precisely, we employ the same error estimator as in the previous section and increase the ansatz degree on marked elements while leaving the grid unchanged.

We found that this method has some practical limitations. The error estimator is correctly identifying the free boundary (or rather the fact that the nonsmooth solution is not accurately represented on coarse elements even when using higher order polynomials). Therefore, the corresponding elements are repeatedly marked in the refinement steps. Figure 7.1.6 shows the distribution of polynomial degrees on each element after 20 refinement steps. We can see that the very high order functions cluster around the free boundary. Many finite element software frameworks, however, only allow for polynomial degrees up to a certain degree. Hence the employable ansatz degrees might saturate while the error estimator still finds significant errors on these

Figure 7.1.6.: Distribution of Degrees on p-Adaptively Refined Space



elements. Moreover, very high order polynomials might lead to high run-time costs or even numerical instabilities.

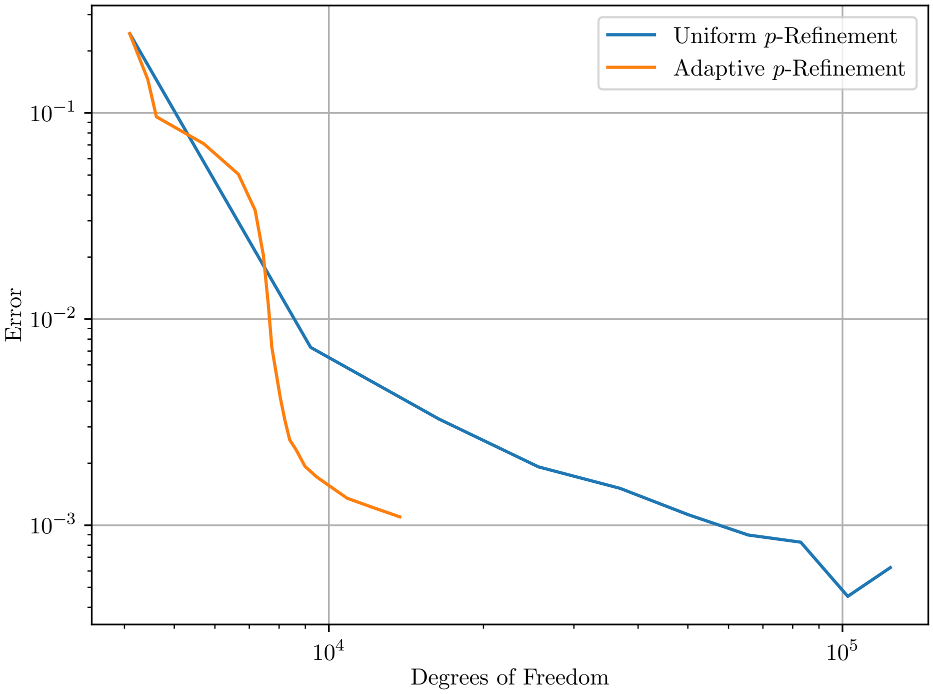
In Figure 7.1.7, we see that our method seems to be unsuited for a purely p-adaptive refinement for this problem. In some cases, the uniform refinement even outperforms the adaptive method. Again, note that for high polynomial degrees, the penalty factor on the adjacent edges will also be very large, rendering the arising system ill-conditioned.

Moreover, we could not get to a high number of degrees of freedom without exceeding the bounds of polynomial degrees that are available in our implementation. Since neither of the (logarithmic) error lines for the adaptive and uniform method seem to follow a linear trend, we omit convergence rates in this section.

Adaptive hp-Refinement

Finally, we will put together all the methods we collected so far: We make use of the fact that DG spaces allow locally changing the grid or increasing the ansatz degree in a straightforward way and apply what is called "hp-adaptivity". As before, we use our hierarchical error estimator as presented in Section 5.1. After marking the elements

Figure 7.1.7.: Errors for Adaptive and Uniform p-Refinements



following the Dörfler marking described earlier, we have to decide whether to h- or p-refine on a given element. We will employ the strategy suggested in [71] which was explained in Section 5.2.2. This involves picking another parameter, namely choosing $\delta \in [0, 1]$ where values closer to 0 prefer h and values closer to 1 prefer p-refinement. We found that the particular choice of δ has some influence on the measured convergence rates, but also on the runtime of the algorithm. Thus, a trade-off has to be made for a particular problem. For the numerical example here, we chose a value $\delta = 0.3$.

Figure 7.1.8 shows that the refinement process evolves according to our expectations: The regions that are close to the free boundary (and hence the less smooth parts of the solution) are resolved by significantly smaller grid elements while the regions in the smooth parts have higher order polynomials on rather coarse elements. Note how this fits nicely to the assumptions on the discretization of the obstacle problem when we discussed a priori estimates for a hp-DG discretization of the obstacle problem in Section 3.3. The difference here, however, is that no a priori information about the solution's structure was supplied. The adaptive procedure was able to separate the smooth and nonsmooth parts of the solution in an automatic way and enriched the DG spaces accordingly.

In the following, we investigate the evolution of the error in the adaptive process

Figure 7.1.8.: *hp*-Adaptively Refined Grid with Ansatz Degrees

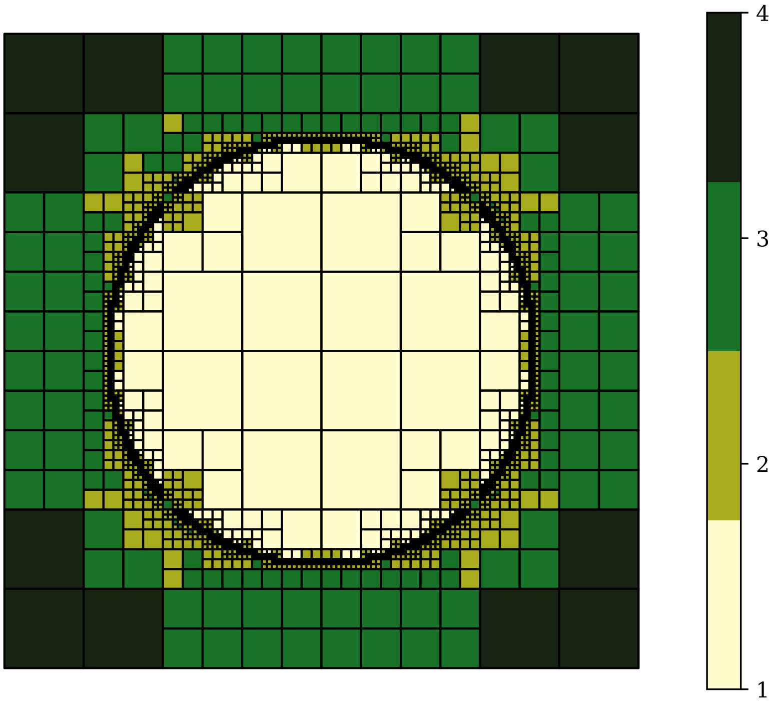
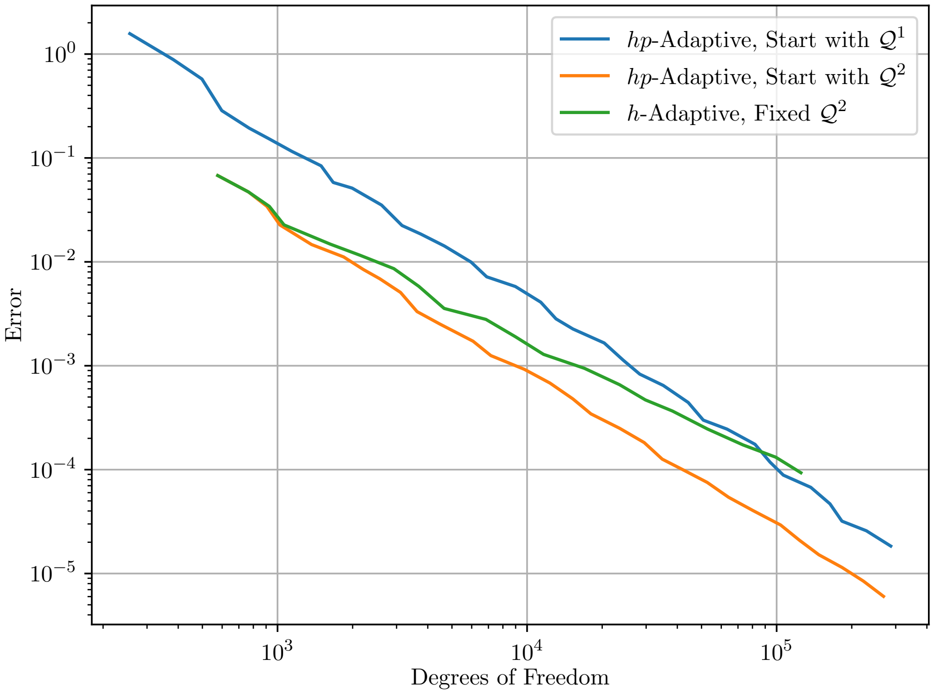


Figure 7.1.9.: Error Plots for Different Adaptive Strategies



for different strategies. We will compare our hp-adaptive algorithm with the best method of the previous sections, namely using $\mathcal{Q}^2$ finite elements and applying h-adaptivity, i. e. refining the grid where necessary. The hp-adaptive algorithm on the other hand also starts with a given mesh and $\mathcal{Q}^k$ elements put can decide between grid refinement and increasing the polynomial degree as argued before. Since for the h-adaptive case we found that starting with $\mathcal{Q}^2$ elements was superior to starting with piecewise linear elements, we also tested the hp-adaptive algorithm with both scenarios. Figure 7.1.9 shows that while the former champion (namely $\mathcal{Q}^2$ with adaptive grid refinements) has a lesser error for a given number of unknowns compared to the hp-adaptive algorithm that started on piecewise linear elements, the latter algorithm exhibits a faster convergence *rate* and its error/unknowns line surpasses the former algorithm eventually. Not surprisingly, the hp-adaptive algorithm starting from $\mathcal{Q}^2$ finite elements combines the best of both worlds: Not only does it show almost the same convergence rate as the same algorithm starting from piecewise linear functions but it also seems to have a lower constant in the asymptotic, expressed through the lower amount of unknowns needed for a given error. This can also be verified by consulting the computed convergence rates in Table 7.1.3.

Degree k	**Refinement Type**	**Convergence Rate $\tilde{c}$**
1	hp-Adaptive	-1.5837
2	hp-Adaptive	-1.4914
2	h-Adaptive	-1.2015

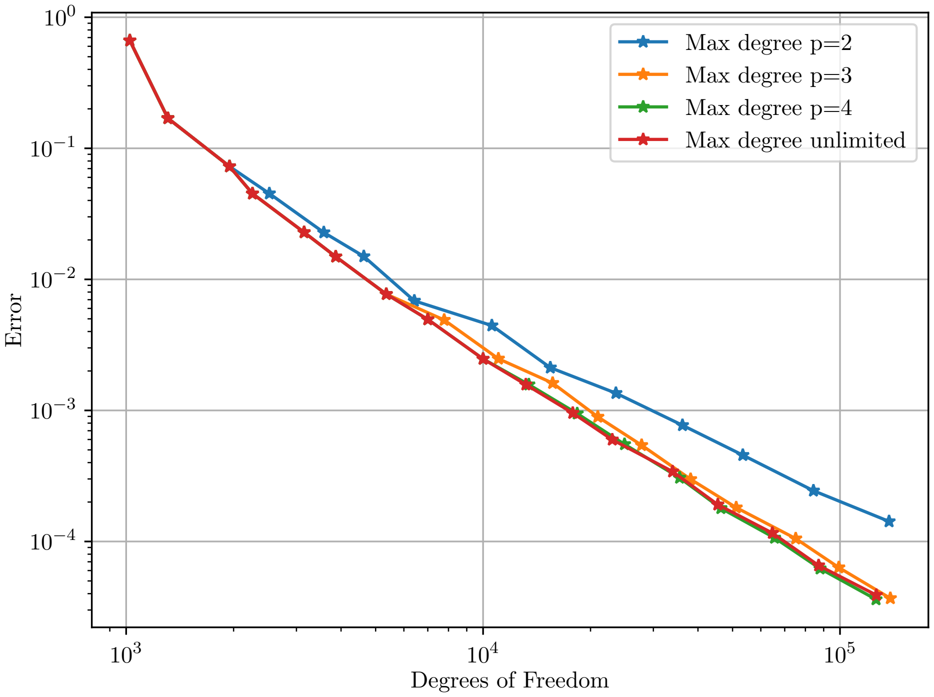
Table 7.1.3.: Convergence Rates for Adaptive Strategies

7.1.3. Limiting Polynomial Degree in hp-Adaptivity

Motivated by the results of Section 7.1.2, one might ask if it is worth to employ higher order polynomial degrees beyond a certain point. The argument could be, that we observed a convergence rate $\tilde{c} \approx -1.58$ which would roughly correspond to a convergence rate of $O(h^3)$ on a uniform grid. Therefore one might be tempted to stop p-refinement after degree 3 or 4 for this problem.

We tested this approach by applying h-refinement on marked elements which already have the maximal degree, even if the hp-criterion suggests to p-refine. Indeed, Figure 7.1.10 shows that using degrees beyond 3 do not gain much for this particular problem.

Figure 7.1.10.: Error Plots for Limited Degrees



This is not too surprising if we remember the result of Theorem 3.50: The local mesh-width h_F in areas where the solution has reduced smoothness (e. g. near the free boundary) would have to scale like h_C^p where h_C is the mesh-width and p the polynomial degree in smoother regions. For larger p this would indeed require h_F to be *very* small compared to h_C.

7.1.4. Accuracy of Error Estimator

Given that Problem 7.1 has a known solution, we are in the comfortable situation to actually test the error estimator in the context of a nonlinear problem. We apply the method suggested in sections 5.1.1 and 5.1.1, namely solving the extended problem in the fine space $\mathcal{Q}$ only approximately by applying a block-Jacobi-like method to each element separately and using matrix-free techniques to evaluate matrix-vector products. Note that the behavior was only analyzed in detail for a linear model problem.

For an *hp*-adaptive approximation of the obstacle problem as outlined in Chapter 4, for every stage of the grid (and corresponding basis), we compute not only the estimated error but also compute the *actual* error in the SIPG norm (7.1) using the analytical solution. In Figure 7.1.11, we can clearly see that the estimated error is very close to the real one and seems to differ mostly by a constant factor. Thus, we gathered evidence that the method which was shown to be effective (under the saturation assumption) for the linear problem can also be a viable choice for more complicated nonlinear problems.

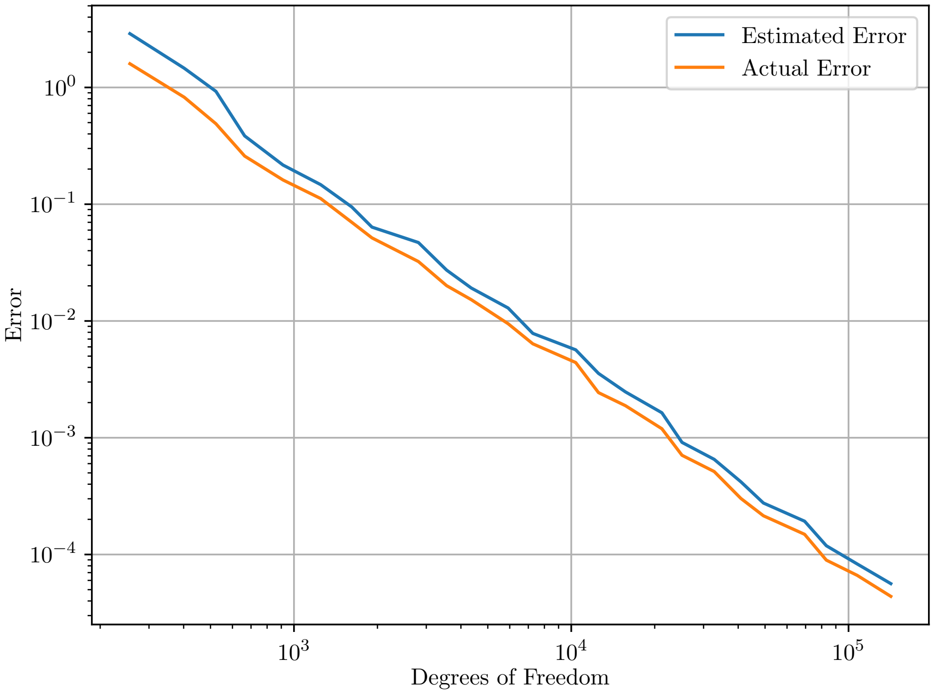
7.1.5. Comparison with Continuous Finite Elements

An often used approach in Finite Element modeling is to use continuous, piecewise linear ansatz functions on a simplex grid (here called $\mathcal{P}^1$). Since DG spaces use more degrees of freedom on a given grid (even when using only piecewise linear finite elements) we are interested whether our *hp*-adaptive approach leads to a discretization which needs less unknowns for a given error bound. To this end we discretized Problem 7.1 on a grid consisting of simplices both with a uniform and an *h*-adaptive refinement strategy and compared it with the results of the previous sections. The *h*-adaptive refinement was performed with a preconditioned hierarchical error estimator as described, e. g. in [76].

In Figure 7.1.12, we see that indeed the continuous variants offer a better ratio of error to number of unknowns compared to the DG spaces with piecewise linear functions, which is hardly surprising as we have more degrees of freedom on a DG space compared to a continuous finite element space defined on the same grid. However, our *hp*-adaptive approach clearly outperforms both the uniform and the *h*-adaptively refined continuous $\mathcal{P}^1$ discretizations after a few iterations of the algorithm. Thus, the DG based method is actually giving a practical advantage over the "classical" approach using continuous, piecewise linear elements.

Remark 7.5. Another aspect would of course be to compare the actual runtime of the algorithms. However, this is naturally an implementation-dependent property.

Figure 7.1.11.: Behavior of Error Estimator and Actual Error



Therefore, we will not compare runtimes here.

7.2. Obstacle Problem with Corner Singularity

In addition to the obstacle problem which we discussed in Section 7.1, we will compute another obstacle solution whose solution has a singularity and in particular has a lower global regularity than the first example. This example was also used in [16].

Problem 7.6. *Consider the L-shaped domain* $\Omega = (-2,2)^2 \setminus [0,2] \times [-2,0]$. *Using Dirichlet data* $u_D = 0$ *and a lower obstacle* $\underline{\psi} \equiv 0$, *we have*

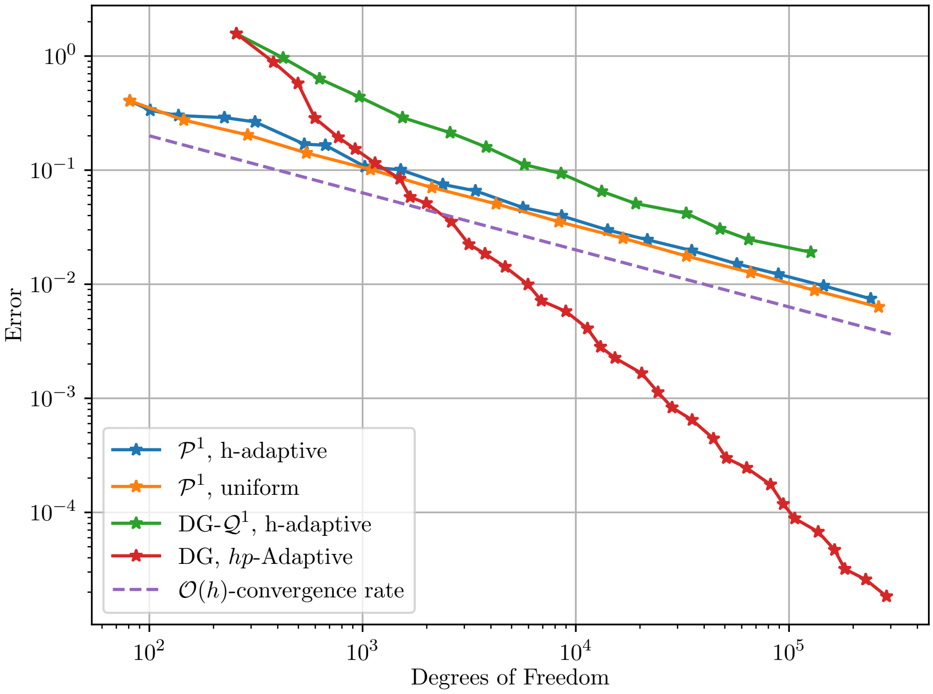
$$\mathcal{K} = \left\{ v \in H_0^1(\Omega) : v \geq \underline{\psi} \ a.\,e. \right\},$$

and we are looking for $u \in \mathcal{K}$ *such that*

$$a(u, v-u) \geq \ell(v-u) \quad \forall v \in \mathcal{K}.$$

Here, we have again $a(v,w) = (\nabla v, \nabla w)$ *and* $\ell(\cdot) = \langle f, \cdot \rangle$ *with* f *given in polar coor-*

Figure 7.1.12.: Continuous vs. Discontinuous Galerkin



dinates by

$$f(r,\theta) = -r^{2/3}\sin(2\theta/3)\left(\gamma_1'(r)/r + \gamma_1''(r)\right) - \frac{4}{3}r^{-1/3}\gamma_1'(r)\sin(2\theta/3) - \gamma_2(r).$$

γ_1 *and* γ_2 *are defined through*

$$\gamma_1(r) = \begin{cases} 1 & \textit{if } \overline{r} < 0, \\ -6\overline{r}^5 + 15\overline{r}^4 - 10\overline{r}^3 + 1 & \textit{if } 0 \le \overline{r} < 1, \\ 0 & \textit{if } 1 \le \overline{r}, \end{cases}$$

with $\overline{r} = 2(r - 1/4)$ *and*

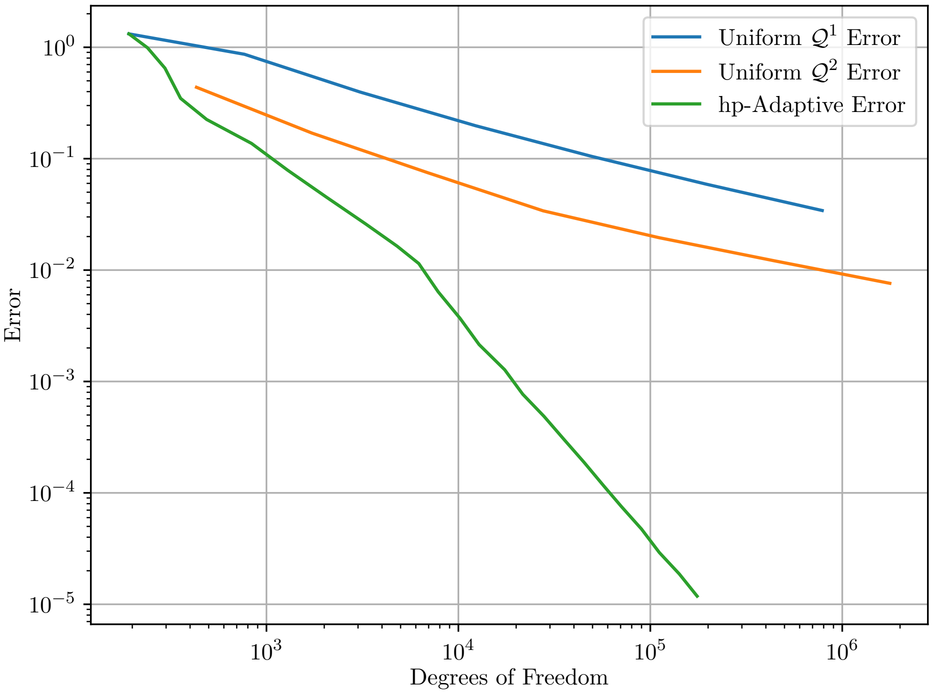
$$\gamma_2(r) = \begin{cases} 0 & \textit{if } r \le 5/4, \\ 1 & \textit{else.} \end{cases}$$

The exact solution u to Problem 7.6 is

$$u(r,\theta) = r^{2/3}\gamma_1(r)\sin(2\theta/3),$$

see [16]. The solution has a singularity at the origin [16] and it holds $u \in H^{5/3-\varepsilon}(D)$ for every $\varepsilon > 0$ and every open neighborhood D of the origin [32]. Hence, this problem is an interesting addition to the examples we computed before since its solution is even less regular than the typical $H^{2.5-\varepsilon}$-regularity we discussed thoroughly in Chapter 3.

Figure 7.2.1.: Discretization Error



Discretization	**Convergence Rate $\tilde{c}$**
$\mathcal{Q}^1$ Uniform	-0.4548
$\mathcal{Q}^2$ Uniform	-0.4837
hp-Adaptive	-1.6674

Table 7.2.1.: Convergence Rates

We discretized the problem the same way we did before, i. e. applying a SIPG scheme for the quadratic part and controlling the obstacle condition in the Lagrange nodes. Due to the reduced regularity, we cannot expect that higher order methods (e. g. $\mathcal{Q}^2$ elements) lead to better convergence rates. Indeed, Figure 7.2.1 and Table 7.2.1 show that a DG discretization with $\mathcal{Q}^2$ elements converges only at a $\mathcal{O}(h)$ rate when using

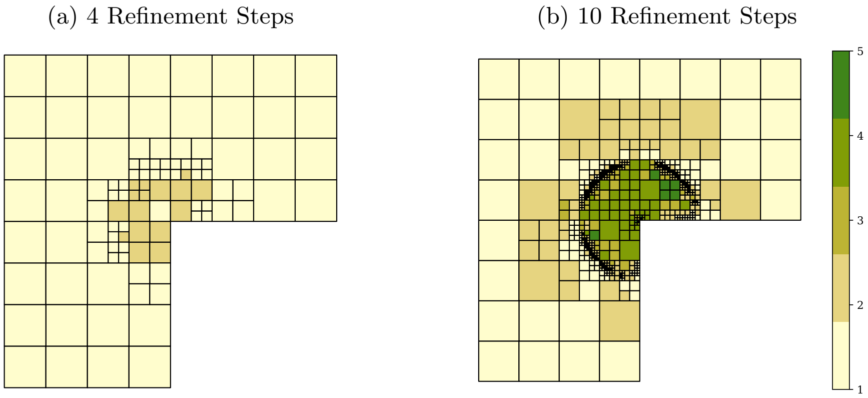


Figure 7.2.2.: Grid after Adaptive *hp*-Refinement.

uniform grid refinements, just as piecewise linear $\mathcal{Q}^1$ elements. Remarkably though, our *hp*-adaptive algorithm is still able to leverage higher order elements at the right places and achieves similar convergence rates as for the smoother problem of Section 7.1.

Figure 7.2.2 indicates that our algorithm resolves both the free boundary where the active and inactive set meet as well as the singularity at the origin with smaller elements, i. e. favoring *h*-refinements. The region where the solution is not constant and smooth on the other hand is discretized with higher order elements.

7.3. Allen–Cahn Phase Field Models

After having discussed the methods introduced in this thesis for the obstacle problem, we will briefly have a look at an application for a phase field model. As discussed in Section 2.4, the Allen–Cahn equation is one of the simplest attempts at modeling phase fields.

7.3.1. Obstacle Potential

Choosing the obstacle potential $\Phi = \chi_{[-1,1]}$ and using an implicit Euler time discretization with timestep τ leads us to an obstacle problem of finding $u_{k+1} \in \mathcal{K}$ such that

$$(\frac{1}{\tau} - \frac{1}{\varepsilon^2})\,(u_{k+1}, v - u_{k+1}) + (\nabla u_{k+1}, \nabla v - \nabla u_{k+1}) \geq \frac{1}{\tau}\,(u_k, v - u_{k+1}) \quad \forall v \in \mathcal{K},$$

see also equation (3.1). The admissible set is the set of $H^1(\Omega)$ functions which are pointwise in the interval $[-1, 1]$ almost everywhere.

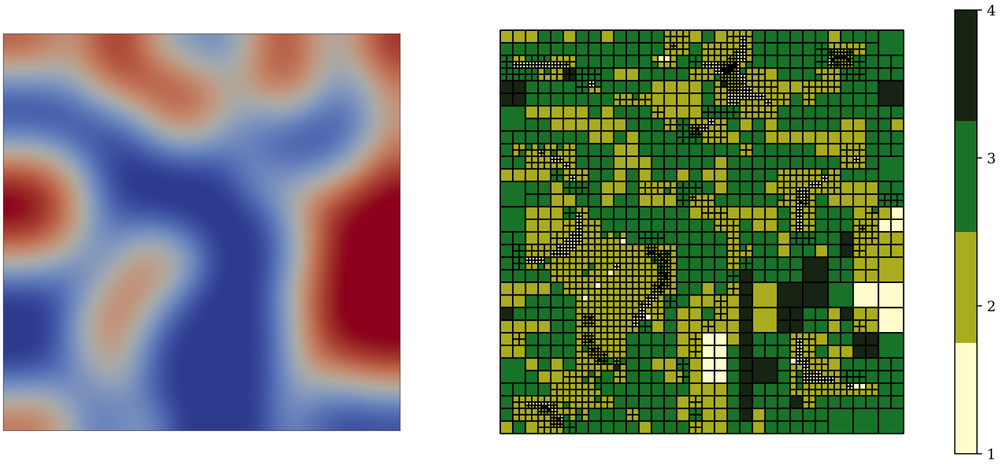


Figure 7.3.1.: Solution and Discretization - Allen–Cahn with Obstacle Potential

The spatial discretization is very similar to the one from the previous section. We replace the $(\nabla v, \nabla w)$ part with the SIPG bilinear form $a_h\,(v, w)$ and get

$$b(u_{k+1}, v - u_{k+1}) \geq \frac{1}{\tau}(u_k, v - u_{k+1}) \quad \forall v \in \mathcal{K}_{hp},$$

with $b(v, w) = \left(1/\tau - 1/\varepsilon^2\right)(v, w) + a_h\,(v, w)$. The admissible set $\mathcal{K}_{hp}$ is once again the set of discrete functions whose nodal values do not exceed the lower and upper obstacle, i. e. the coefficients of the discrete functions are in the interval $[-1, 1]$ for this particular model. For more details, consult Section 7.1.1.

We chose to look at a single timestep, i. e. we approximate the solution of the (elliptic) variational inequality (3.1) for a given u_k. The time step u_k is obtained by computing a few timesteps into the evolution to make sure it has the typical phase field profile. Unfortunately, we cannot rely on an analytic solution for this problem. Therefore, the discretization error has to be approximated. We do so by computing a DG solution u^* on a very fine uniform grid with uniform degree of $p = 5$. Due to the uniform structure we assume that discontinuities and other problems will be negligible. In our adaptive algorithm, we compute the solutions $\tilde{u}^0_{k+1}, ..., \tilde{u}^m_{k+1}$ on a sequence of discrete spaces $V^0, \ldots, V^m$ determined through the algorithms described in Chapter 5. A plot of a single solution $\tilde{u}^i_{k+1}$ along with its discretization (that is, the mesh and the employed polynomial degrees) can be seen in Figure 7.3.1.

As long as the discrete space V^i at hand is still coarser (in the sense that the mesh is everywhere at least one level coarser than for the reference solution and the local polynomial degree is not greater than $p = 5$), we can compute the approximated error

$$|||\tilde{u}^i_{k+1} - u^*||| \approx |||\tilde{u}^i_{k+1} - u_{k+1}|||. \tag{7.2}$$

Another information about the error we have is the estimated error computed during the adaptive process. In Section 7.1.4 we have seen that it captures the error quite well for the obstacle problem. Since the estimated error does need a reference solution, the corresponding trajectory in Figure 7.3.2 is much longer.

Figure 7.3.2.: Allen–Cahn Obstacle Discretization Error

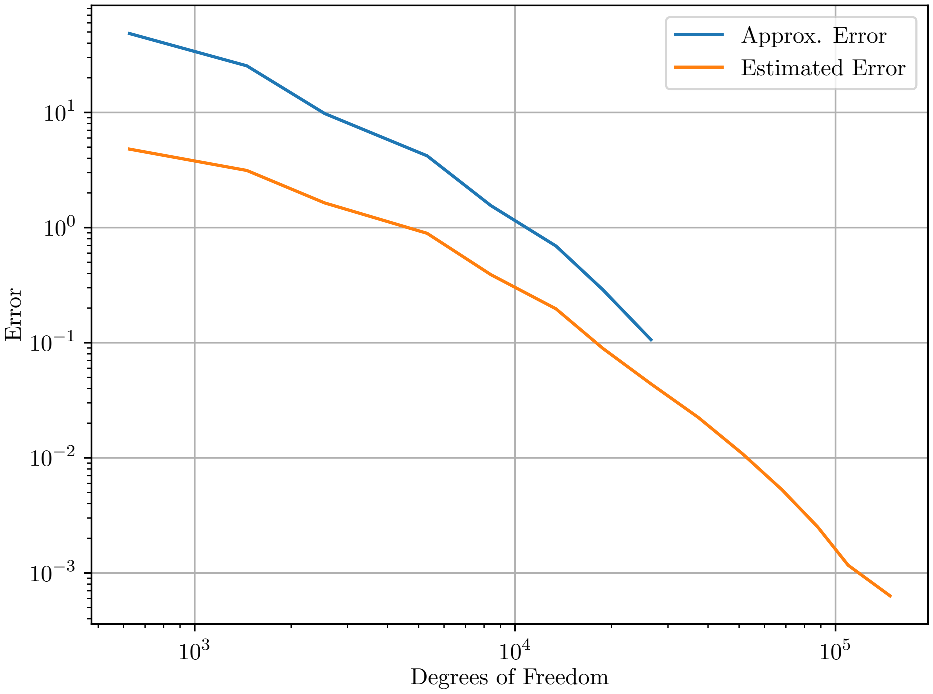


Figure 7.3.2 shows that for the Allen–Cahn problem our error estimator is at first not as good as for the simpler obstacle problem but still within one order of magnitude. It seems that for higher number of degrees of freedom, the approximated error from (7.2) and the estimated error are getting closer.

Remark 7.7. Of course, it is not surprising that the estimated error is not too precise for a small number of unknowns: When the grid is still very coarse, even the finer space $\mathcal{Q}$ will be too coarse to catch the small scale features of the phase field.

With respect to convergence rates, we observe very similar rates to the ones observed for the stationary obstacle problem in earlier sections, see Table 7.3.1.

Error	**Convergence Rate $\tilde{c}$**
Approximated Error	-1.6246
Estimated Error	-1.7101

Table 7.3.1.: Convergence Rates

7.3.2. Logarithmic Potential

In this section, we will investigate a discretized Allen–Cahn equation once more. This time, however, we employ a different type of potential, namely the logarithmic potential, cf. Section 2.4. More precisely, we have

$$\Phi(\xi) = \chi[-1,1](\xi) + \frac{\theta}{2}\left[(1+\xi)\ln(1+\xi) + (1-\xi)\ln(1-\xi)\right].$$

In the concave part, i. e. $-\frac{\theta_c}{2}\xi^2$, we choose $\theta_c = 1$ for simplicity and to stay consistent with the obstacle potential model we investigated before. For the time discretization, we use again the fully implicit scheme from Section 3.1.1. In this case, this gives us variational inequality of the *second kind* in each time step, cf. (3.1),

$$\begin{aligned}&\left(\frac{\varepsilon}{\tau} - \frac{1}{\varepsilon}\right)\left(u^{m+1}, v - u^{m+1}\right) + \varepsilon\left(\nabla u^{m+1}, \nabla(v - u^{m+1})\right)\\ &+\frac{1}{\varepsilon}(\phi(v) - \phi(u^{m+1})) \geq \frac{\varepsilon}{\tau}\left(u^m, v - u^{m+1}\right) \quad \forall v \in H^1(\Omega).\end{aligned}$$

As usual, we have $\phi(v) = \int_\Omega \Phi(\xi)\,\mathrm{d}\xi$. Using a DG space $V^p_{\mathcal{T}}$ with the SIPG method, we can discretize the problem as before. To do so, we replace the integral $\int_\Omega \Phi(v(\xi))\,\mathrm{d}\xi$ by a quadrature rule $\sum_i \omega_i \Phi(v(x_i))$, cf. Section 3.4. Thus, in the $(m+1)$th time step, we solve

$$b(u_{m+1}, v - u_{m+1}) + \sum_i \omega_i \Phi(v(x_i)) - \sum_i \omega_i \Phi(u^{m+1}(x_i)) \geq \frac{1}{\tau}(u_m, v - u_{m+1}) \quad \forall v \in \mathcal{K}_{hp},$$

using the notation from the previous section.

In our numerical example, we again compare the adaptively computed solutions to a reference solution which was computed on a fine uniform grid. A plot of the solution in one time step and an illustration of the discretization can be found in Figure 7.3.3. The corresponding convergence rates, shown in Figure 7.3.4, which uses a value of $\theta = 0.1$ in the potential, exhibit a similar behavior as we observed for the obstacle potential: While the estimator seems to underestimate the error by a factor, we have that asymptotically the estimated and the approximated errors seem to behave similarly, thus fueling our confidence in the error estimator. The convergence rates (both for the estimated and the approximated errors) are reported in Table 7.3.2. It seems that the convergence is slightly worse for the logarithmic potential than it was for the obstacle potential.

Error	**Convergence Rate** $\tilde{c}$
Approximated Error	-1.4026
Estimated Error	-1.3729

Table 7.3.2.: Convergence Rates

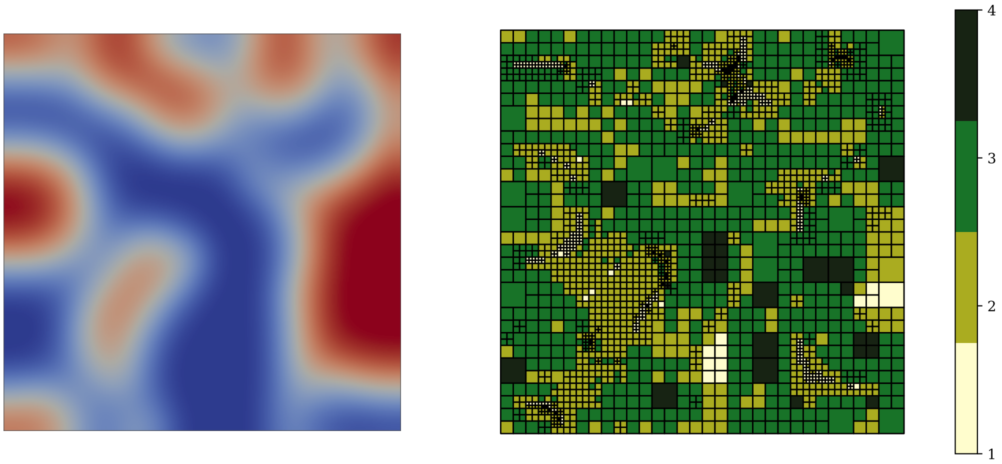


Figure 7.3.3.: Solution and Discretization - Allen–Cahn with Logarithmic Potential

7.4. Application: Ambrosio–Tortorelli for Image Segmentation

In the area of image segmentation, the goal is to separate regions in an image which are homogeneous in a given sense (e. g. having similar gray scales). The curves separating these regions are called "edge sets"[70]. Many algorithms and approaches to find suitable edge sets have been proposed, some of these are PDE-based or rely on variational methods.

In the following, we follow the notation of [70]. One particular model is obtained by minimizing the Mumford-Shah energy [87],

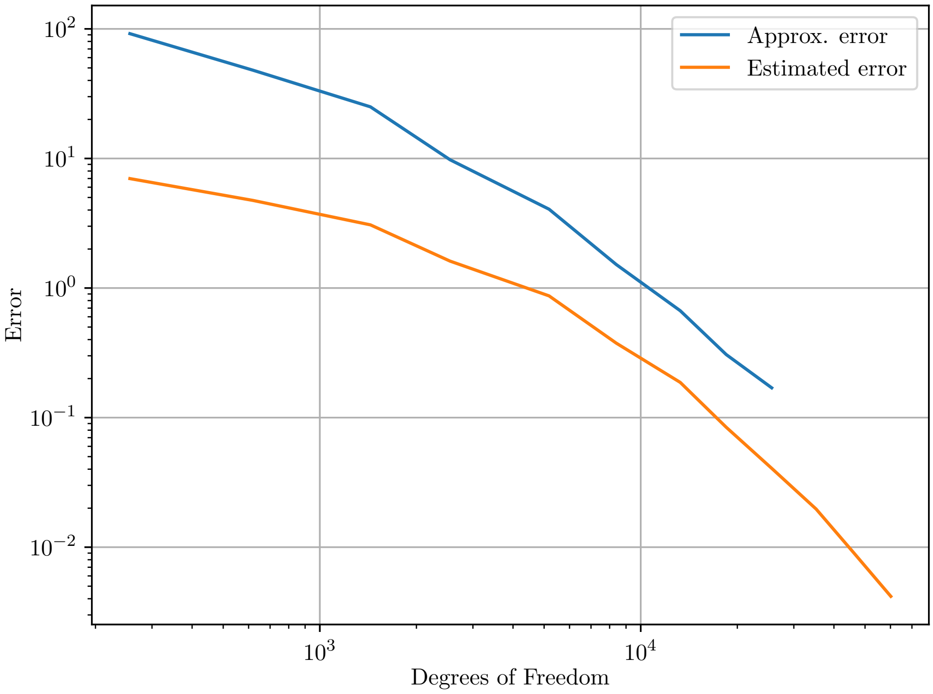
$$\mathrm{MS}(u,\Gamma) = \frac{1}{2}\int_{\Omega\setminus\Gamma} |\nabla u|^2 \,\mathrm{d}x + \frac{\alpha}{2}\mathcal{H}^1(\Gamma) + \frac{\beta}{2}\|u-g\|^2_{L^2(\Omega)}, \tag{7.3}$$

$$(u,\Gamma) \in \mathcal{A} = \Big\{\Gamma \subset \Omega \text{ closed and } u \in H^1(\Omega\setminus\Gamma)\Big\}. \tag{7.4}$$

Here, α and β are positive real numbers and $\mathcal{H}^1$ denotes the one-dimensional Hausdorff-measure. The function g represents the grayscale image we want to segment and Ω is the image domain. It is natural to view $g \in L^\infty(\Omega)$ as a piecewise constant function on the domain $\Omega = (0,m)\times(0,n)$ where the image consists of $m \times n$ pixels. The energy (7.3) penalizes jumps in u while simultaneously penalizing deviations from the image g (weighted by a parameter β). u can thus be viewed as a (piecewise) smooth approximation of g. The geometric part, namely the curve Γ that separates the regions, on the other hand, should not be too long; therefore its length is penalized with a parameter α.

However, the model has some problems both with respect to analysis and numerical treatment. The optimization will be difficult since u depends on Γ and the set of

Figure 7.3.4.: Allen–Cahn Logarithmic Discretization Error, $\theta = 0.1$



possible Γ is not even linear [70]. Moreover, the notion of the curve can be difficult to capture in a discretization.

As a remedy, we will consider a phase field approach via the so-called *Ambrosio–Tortorelli* functional

$$\begin{aligned} \mathrm{AT}(u,z) &= \frac{1}{2}\int_\Omega (z^2+\eta)|\nabla u|^2 \,\mathrm{d}x \\ &\quad + \frac{\alpha}{2}\left(\varepsilon\int_\Omega |\nabla z|^2 \,\mathrm{d}x + \frac{1}{4\varepsilon}\int_\Omega |z-1|^2 \,\mathrm{d}x\right) \\ &\quad + \frac{\beta}{2}\int_\Omega |u-g|^2 \,\mathrm{d}x. \end{aligned} \tag{7.5}$$

Here, the functional is minimized over the set $H^1(\Omega) \times \mathcal{K}$ with

$$\mathcal{K} = \left\{ v \in H^1(\Omega) : 0 \le v \le 1 \text{ a. e.} \right\}.$$

It can be shown [4, 5] that the solutions converge to the solutions of the Mumford–Shah model in the sense of Γ-convergence. Note that the functional is not convex but biconvex, i. e. $\mathrm{AT}(u, \cdot)$ and $\mathrm{AT}(\cdot, z)$ are convex for fixed u and z, respectively. For a fixed u, the minimization in z is an obstacle problem motivating us to apply our method to this particular problem.

Ideally, in the regions separated by the free boundary (approximated through the phase field variable z), u should be reasonably smooth, justifying the use of adaptive methods. Before we introduce the full adaptive algorithm, we explain how we discretize the problem and compute a solution for a given basis.

7.4.1. Discretization and Algebraic Solution

Looking at the Ambrosio–Tortorelli energy (7.5), we observe that all the operators involved can be discretized using the SIPG approach we presented in Chapter 3. More specifically, we have weighted L^2 products and and weighted Laplace terms. Therefore, we can use almost the same methods as before.

Given a grid $\mathcal{T}$ and a distribution of degrees p, we employ for both the image variable u and the phase field variable z the same basis $V_{\mathcal{T}}^{p}$. Let $v, w \in V_{\mathcal{T}}^{p}$. Using the SIPG approach, the following energies are involved:

$$\begin{aligned}
a_h(v) &= \int_\Omega \nabla v \nabla v \,\mathrm{d}x + \sum_{e\in\Gamma} \left(-2 \int_e [\![v]\!] \{\nabla v\} \,\mathrm{d}S + \frac{\sigma_e}{|e|} \int_e [\![v]\!]^2 \,\mathrm{d}S \right), \\
m_h(v) &= \int_\Omega v^2 \,\mathrm{d}x, \\
c_h(v,w) &= \int_\Omega (w^2+\eta)|\nabla v|^2 \,\mathrm{d}x \\
&\quad + \sum_{e\in\Gamma} \left(-2 \int_e [\![v]\!] \left\{ (w^2+\eta)\nabla v \right\} \mathrm{d}S + \frac{\sigma_e}{|e|} \int_e \left\{ w^2 \right\} [\![v]\!]^2 \,\mathrm{d}S \right).
\end{aligned}$$

In the last term of c_h which penalizes jumps in v, we have included $\left\{w^2\right\}$ even though typically one only has an upper bound for the weight function included in the penalty factor. In our case, however, we also want to minimize in the w direction and therefore have to make sure that the bilinear form is also coercive in the w-direction.

The SIPG-modified functional in the discrete space is

$$\begin{aligned}
\mathrm{AT}_{hp}(u,z) &= \frac{1}{2} c_h(u,z) + \frac{\alpha}{2} \left(\varepsilon a_h(z) + \frac{1}{4\varepsilon} m_h(z - \mathbb{1}) \right) \\
&\quad + \frac{\beta}{2} \int_\Omega |u-g|^2 \,\mathrm{d}x,
\end{aligned}$$

where $\mathbb{1}(x) \equiv 1$ is the constant one function.

Finally, we have to discretize the constrained set $\mathcal{K}$ and do so in the established manner by controlling the obstacle condition in the Lagrange nodes:

$$\mathcal{K}_{hp} = \left\{ v \in V_{\mathcal{T}}^{p} : 0 \le v|_K(x_k) \le 1, \quad \forall K \in \mathcal{T} \forall x_k \in \mathbb{X}_K \right\}.$$

Hence, we try to solve the following problem:

Problem 7.8. *Find* $(u, z) \in V^p_{\mathcal{T}} \times \mathcal{K}_{hp}$ *such that*

$$\mathrm{AT}_{hp}(u, z) \to \min\,. \tag{7.6}$$

While optimizing the global problem can be hard, we make use of the fact that the optimization in the individual directions are convex problems and employ an alternating ("operator split") algorithm, see, e. g. [60]. Given initial guesses u^0, z^0 in $V^p_{\mathcal{T}}$, we alternate between minimizing in u and minimizing in z direction while keeping the other direction fixed, respectively. The same method for a (continuous) finite element discretization was used in [30] and other articles. To the best of the author's knowledge, there are no proofs of convergence, though. However, the sequence $\mathrm{AT}_{hp}(u_n, z_n)$ generated by Algorithm 5 is decreasing [30].

Algorithm 5 Alternating Minimization Algorithm

```
procedure AlternateMin($u^0$, $z^0$)
    $n = 0$
    repeat
        $u^{n+1} = \arg\min_{u \in V^p_{\mathcal{T}}} \mathrm{AT}_{hp}(u, z^n)$
        $z^{n+1} = \arg\min_{z \in \mathcal{K}_{hp}} \mathrm{AT}_{hp}(u^{n+1}, z)$
        $n \leftarrow n + 1$
    until $\|u^n - u^{n-1}\| <$ TOL and $\|u^n - u^{n-1}\| <$ TOL
    return $(u^n, z^n)$
end procedure
```

Since the arising convex minimization problems are an unconstrained quadratic problem in u and a quadratic obstacle problem in z, we can use the multigrid solver and the TNNMG algorithm from Chapter 4, respectively.

Remark 7.9. For models of brittle fracture that employ a very similar ansatz (also using the Ambrosio–Tortorelli energy), the TNNMG algorithm has been successfully applied to the global functional instead of using operator split [63].

Remark 7.10. In [30], it is argued that for a function $v \in C(\Omega)$, the function $\min(1, v)$ would have a lower energy than v (and similarly for $\max(0, v)$). Therefore, the author concludes that one can ignore the obstacle restriction and simply solve the unconstrained quadratic problem. It is not so obvious if this also holds for the discrete case, however. In our experiments, we observed that the approximate solution might violate the constraints slightly if they are not imposed. Hence, we still solve the obstacle problem in $\mathcal{K}_{hp}$ as described above.

7.4.2. Adaptive Algorithm

Since the edges in the image are lower dimensional objects which are modeled through the phase field approach in the functional AT_{hp}, we expect that a very fine resolution of these areas is necessary to represent z accurately. Moreover, some regions of the image domain might be rather homogeneous, which lead us to believe an *hp*-adaptive method could also benefit the approximation of the image variable u.

In Chapter 5, we explained how to use hierarchical estimators for DG discretizations of variational inequalities. Hence, we can directly build on that knowledge. In the following, we will briefly explain the ideas and refer the reader to Chapter 5 for more details. The adaptive algorithm is performed by applying the hierarchical estimator to both subproblems we have seen in the alternating minimization algorithm. Having obtained approximate elementwise errors, the marking and the *hp*-decision can be performed as before. Thus, both regions which cause large errors for the image variable u and regions which cause large errors in the edge variable z should be appropriately refined.

Remark 7.11. It has to be noted that the method of Section 5.1.1, namely performing a single step of a block Jacobi method to compute an approximation of the hierarchical error estimator, is of course a very rough estimate: Not only do we replace the full solution of the given subproblem by performing only a single iteration of the method, but we also do not attempt to solve the global problem but restrict ourselves to approximately solving the subproblems in both directions. If the hierarchical error approximation for a given problem is not satisfying, one could try to run an alternating minimization also in the larger space $\mathcal{Q}$.

Starting with a DG space $\mathcal{S}$, the algorithm can be written as follows:

Algorithm 6 Alternating Adaptive Algorithm

```
1. Solve for $u, z$ in $\mathcal{S}$, e. g. using alternating minimization.
2. Estimate error in $u$ and refine:
    a) Choose incremental space $\mathcal{Q} \supset \mathcal{S}$, represent $u$ and $z$ by $u_{\mathcal{Q}}, z_{\mathcal{Q}} \in \mathcal{Q}$.
    b) Compute an approximation $e_u \approx \arg\min_{u \in \mathcal{Q}} \mathrm{AT}_{hp}(u, z_{\mathcal{Q}}) - u_{\mathcal{Q}}$ (see Section 5.1.1) and derive local and global error estimators.
    c) Mark elements for refine, choose $h$ or $p$ refinement for each element.
    d) Obtain new space $\tilde{\mathcal{S}} \supset \mathcal{S}$ accordingly.
3. Analogously, estimate error in $z$ and refine, again:
    a) Choose incremental space $\tilde{Q} \supset \tilde{\mathcal{S}}$, represent $u$ and $z$ by $u_{\tilde{\mathcal{Q}}}, z_{\tilde{\mathcal{Q}}} \in \tilde{\mathcal{Q}}$.
    b) Compute an approximation $e_z \approx \arg\min_{z \in \tilde{\mathcal{Q}}} \mathrm{AT}_{hp}(u_{\tilde{\mathcal{Q}}}, z) - z_{\tilde{\mathcal{Q}}}$ and derive local and global error estimators.
    c) Mark elements for refine, choose $h$ or $p$ refinement for each element.
    d) Obtain new space $\hat{\mathcal{S}} \supset \tilde{\mathcal{S}}$ accordingly.
4. If global errors are small enough, stop. Else, set $\mathcal{S} \leftarrow \hat{\mathcal{S}}$ and go to Step 1.
```

In practice, the stopping criterion in Step 4 of Algorithm 6 might replaced by performing a fixed number of steps.

Remark 7.12. Since the performance of hierarchical error estimators is connected to the saturation assumption 5.11, we might have bad performance if $\beta \approx 1$. In particular, the saturation assumption is linked to severity of the oscillations in the right hand side [50]. In our case, the right hand side represents an image, which might have sharp edges and strong oscillations. To capture these, one can include oscillation terms (which are usually of higher order). Let f be the data of the current problem and f_h its L^2-projection into the finite element space at hand.

$$\operatorname{osc}^2(f) = \sum_{K \in \mathcal{T}} \frac{h_K^2}{p_K^2} \|f - f_h\|_{0,K}^2, \tag{7.7}$$

see also [49] and, for a more DG specific reference, [22]. For the obstacle problem, similar oscillation terms were derived in [77].

Note that the computation of f_h involves solving a linear system containing the L^2 mass matrix of the finite element space. Since the mass matrix of a discontinuous Galerkin space is block diagonal, this can be done in parallel just like the other parts of the hierarchical error estimator.

7.4.3. Examples

For this application oriented example, we will not present error graphs since it is now clear what the analytical solution (or a sufficiently accurate approximation) would look like. Rather, we show a selection of example input images and the respective approximations of the smooth image function u and the edge indicator z.

Boat

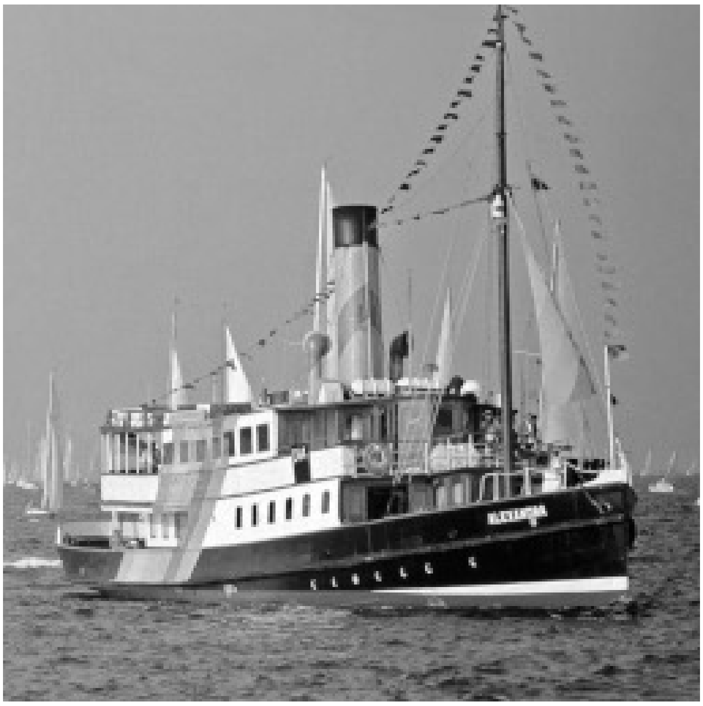
As a first test image, we use the *boat*[1] image, Figure 7.4.1. For this problem, we used the following model parameters:

$$\begin{aligned} \alpha &= 30, \\ \beta &= 10, \\ \varepsilon &= 0.85, \\ \eta &= 0.005. \end{aligned}$$

Note that the value of ε is higher than usual to increase the visibility of the detected edges when printed on this page. As for the discretization, we only implemented a purely h-adaptive refinement process as described in the previous section. No oscillation terms (cf. Remark 7.12) were added to the error estimator in this example.

[1] This work is a derivative of “Alexandra in speziellen Farben zur Windjammerparade 1972 in Kiel” by Wolfgang Fricke, used under CC BY. Source: `https://commons.wikimedia.org/wiki/File:Alexandra_P_Kiel_03-09-1972_(1).jpg`, retrieved 9 April 2022.

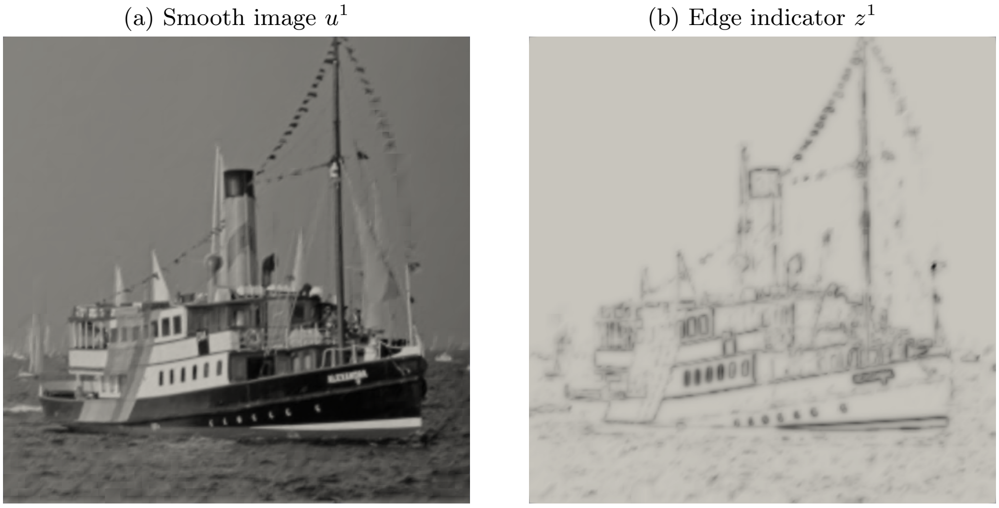
Figure 7.4.1.: Input image *boat*[1]



As we can see in Figure 7.4.2b, the edge variable z can indeed be used to visualize the edges in the picture and thus to segment the image. In Figure 7.4.3, we can see the generated grid. The adaptive algorithm captures the parts where the image exhibits sharp color transitions (in particular the light and dark paint of the boat, its mast, and the lateral windows). Less distinct transitions (as seen, e. g. in the waves on the bottom of the image) are less pronounced with these parameters.

Sharp Edges: Letters

Our second test image Figure 7.4.4 is structurally different from the first one. It contains a few monochromatic letters on a solid background. Thus, in contrast to the boat example, we have few but very pronounced edges. In particular, these might be aligned with the grid axes. Since the input is a 256×256 pixel image (and not a vector graphic), edges which are not parallel to grid axis do not look smooth in close-up. The sharp discontinuities are sometimes not well captured even in the finer space $\mathcal{Q}$ when estimating the error. This might lead to discretizations which do not *look* pleasant, as our human eye is quite capable at recognizing inconsistencies in the pictures. Therefore, a sufficiently fine initial grid might be required, possibly amended with the oscillation terms from (7.7). For this model, the following parameters are



(a) Smooth image u^1

(b) Edge indicator z^1

Figure 7.4.2.: Discrete Solutions for the boat Image

Figure 7.4.3.: Discretization^1

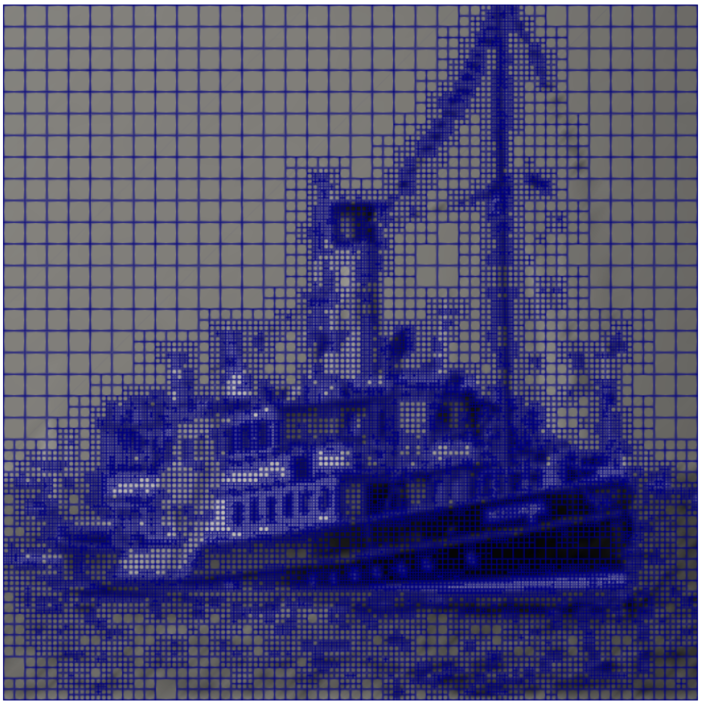
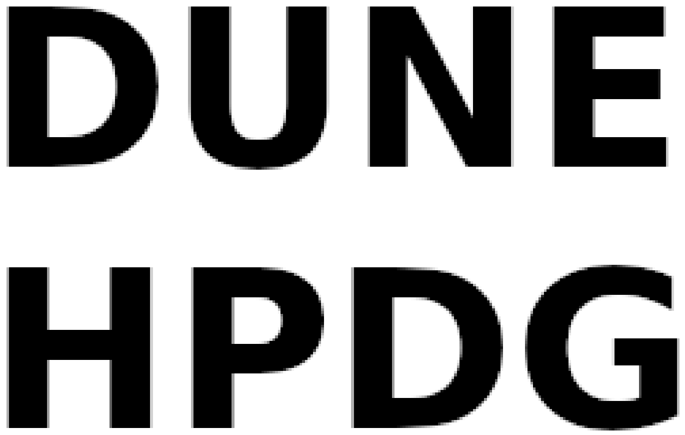


Figure 7.4.4.: Input image *letters*



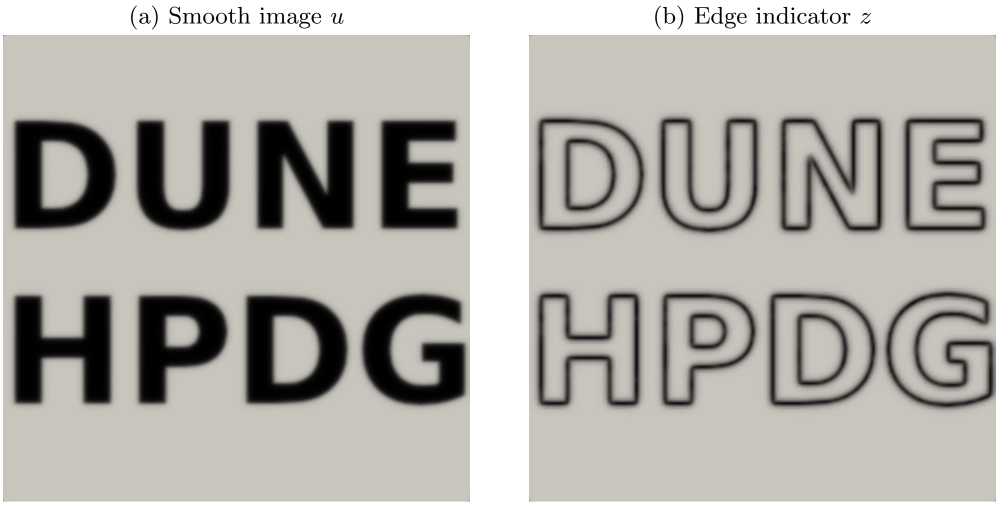
used:

$$\begin{aligned}\alpha &= 75,\\ \beta &= 1,\\ \varepsilon &= 0.85,\\ \eta &= 0.005.\end{aligned}$$

Again, we apply an h-adaptive process using the discretization and algorithms described before.

7.5. Algebraic Solver

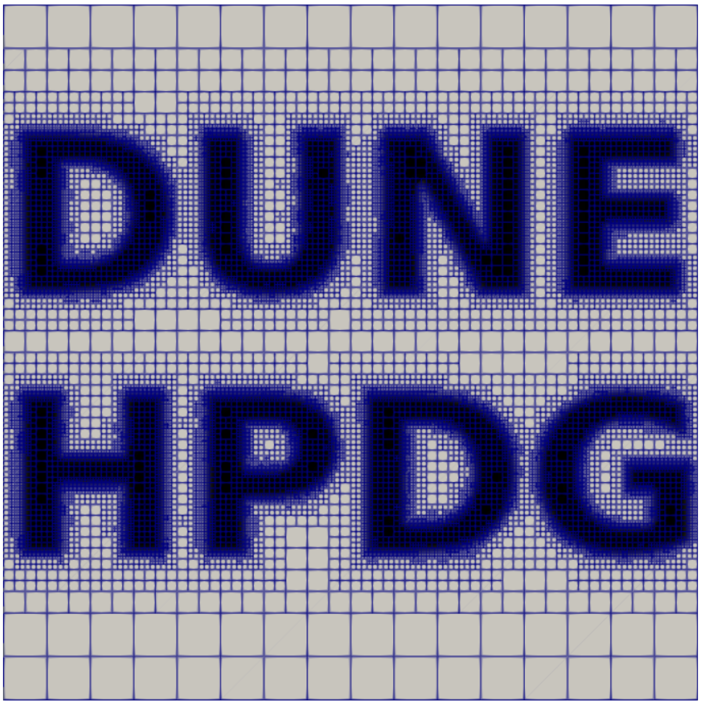
Finally, we want to briefly highlight the quality of the algebraic solvers. In Chapter 4, we suggested to use a geometric multigrid approach with a layer of different p-levels ("p-multigrid") on top as a linear solver, which forms one of the building blocks of the TNNMG method, see Section 4.1. As discussed before, we cannot expect the convergence to be independent from the number of grid and p levels.


(a) Smooth image u

(b) Edge indicator z

Figure 7.4.5.: Discrete Solutions for the text Image

Figure 7.4.6.: Discretization



7.5.1. Linear System

At first, we want to consider the solution of a linear system. We discretize the following linear problem

Problem 7.13.

$$\begin{aligned} (\nabla u, \nabla v) &= \ell(v), && \forall v \in H^1(\Omega), \\ u &= 0 && \text{on } \partial\Omega. \end{aligned}$$

Here, we choose $\Omega = [0,1]^2 \subset \mathbb{R}^2$ *and* $\ell(v) = -10 \int_\Omega v \,\mathrm{d}x$.

For the discretization, we once more use the SIPG approach from Section 3.2, using a penalty parameter of $\sigma = 2p^2$.

Remark 7.14. Note that the choice of the penalty parameter influences the condition number of the stiffness matrix. If we choose a penalty number which is too large, the convergence rate of our solver will suffer.

An extensive discussion about the solver including dependencies on various parameters can be found in [8]. In particular, the authors show how a higher number of pre- and post-smoothing steps can accelerate the convergence.

For our linear numerical example, we use $u_0 = 0$ as the initial iterate or each problem. The convergence rate ρ is calculated as follows:

$$\rho = \left(\frac{\|u^* - u_\nu\|_A}{\|u^*\|_A} \right)^{1/\nu}$$

where k is the number of iterations needed to reduce the error below a threshold of 10^{-8}, $\|\cdot\|_A$ is the energy norm and u^* is a reference solution computed with a much smaller tolerance. Depending on the number of levels k and the polynomial degree p, we solve Problem 7.13 numerically on a uniform grid. We used $m = 3$ pre- and post-smoothing steps, respectively. For our model problem, Table 7.5.1 shows convergence rates of the linear multigrid solver introduced in Section 4.2.

	k=2	**k=3**	**k=4**	**k=5**
p=1	0.131685	0.12166	0.11985	0.241822
p=2	0.15661	0.235043	0.490307	0.696959
p=3	0.311607	0.546574	0.730558	0.849779
p=4	0.400005	0.66417	0.821656	0.906722

Table 7.5.1.: Convergence rates of *hp*-multigrid for linear problem.

We can clearly see that the number of grid levels and the polynomial order affects the convergence rates. This is not surprising due to the use of “inherited” bilinear forms, cf. Section 4.2 (see also [8]). To an extent, one can overcome this level-dependence by removing parts of the penalty terms on coarser levels, see Remark 4.7. This takes a higher computational load and more complex implementations.

Clearly, the convergence rates from Table 7.5.1 are disappointing, in particular if one compares them to the rates for geometric multigrid methods on uniform meshes when a continuous $\mathcal{P}^1$ finite element space is used. Since the convergence speed of the TNNMG method for nonlinear problems is asymptotically governed by the quality of the linear solver [67], we are interested in better convergence rates. Adding a lower obstacle $\underline{\psi} \equiv -\frac{1}{2}$, we turn Problem 7.13 into an obstacle problem. The TNNMG algorithm with the *hp*-multigrid as linear solvers gives the following convergence rates, see Table 7.5.2.

	k=2	**k=3**	**k=4**	**k=5**
p=1	0.0209317	0.0698919	0.0920876	0.12038
p=2	0.0735163	0.0882236	0.108398	0.316817
p=3	0.0876253	0.125077	0.333269	0.561108
p=4	0.0689302	0.213225	0.483802	0.703574

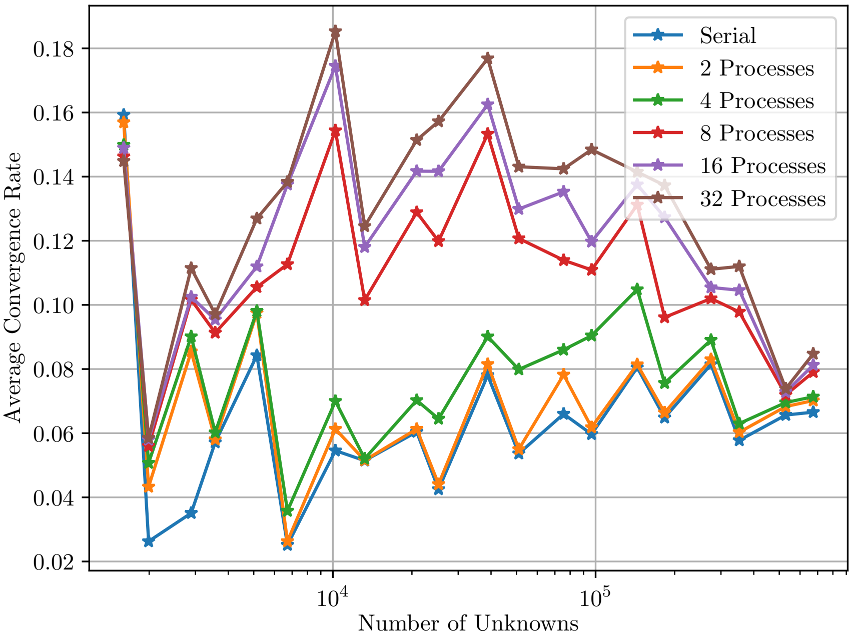
Table 7.5.2.: Convergence rates of TNNMG for an obstacle problem.

As we can see, the TNNMG algorithm for the discretized obstacle problem also admits a level dependency (albeit slightly better than for the linear problem). It has to be noted, though, that we did not account for the fact that the TNNMG method is fastest only once the active set has been determined correctly. Therefore often *nested iterations* (i. e. using the interpolated solutions from coarser spaces as initial iterate) are employed. Given the fact that we focus on adaptive procedures, we can obtain coarse grid solutions naturally: Since the current space was obtained by refining from a coarser space where the (approximate) solution is known, we can use this initial iterate for the next application of the TNNMG method. In particular, we found that this gives far superior convergence rates compared to what we just found for the uniform problems using $u_0 = 0$ as an initial guess. We postpone the extended discussion of the solver performance in the adaptive setting to the following Section 7.5.2, where the serial and parallel solvers are discussed for the obstacle problem. For example, Figure 7.5.1 shows that the convergence rates stay well below 0.2 for the obstacle problem even for a high number of unknowns and a parallel setup if the adaptive procedure with nested iteration is used. Hence, even if the *hp*-multigrid approach might look like it yields unsatisfactory convergence rates, we can indeed use it to efficiently solve the arising algebraic problems with the TNNMG method.

7.5.2. Parallel Solution of the Algebraic Problem

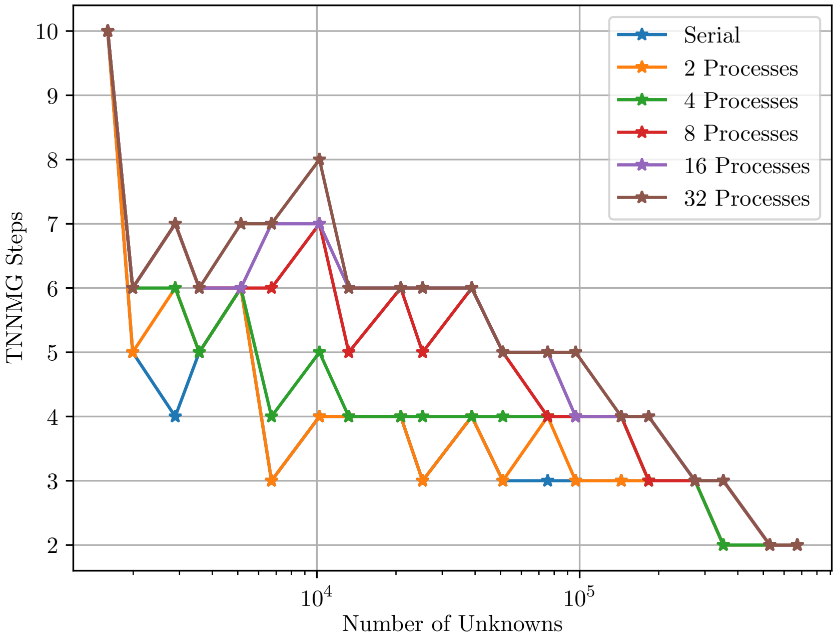
In the following, we will briefly investigate how the parallelization approaches from Section 4.3 influence the convergence of the algebraic solver. To do so, we compare different amounts of parallel nodes by solving the same problem ("strong scaling"). More precisely, we solve once more Problem 7.1 by the *hp*-adaptive procedure described before. The initial domain is split across several nodes and the resulting systems are solved in parallel. After each refinement step the new system on the greater space is solved by using the interpolated solution from the previous step as initial iterate. This helps in particular our (nonsmooth) Newton method to converge more rapidly. In contrast to the previous section, we will not address the question of convergence of the discretization here (after all, we compute the same trajectory of solutions as in the serial case) but rather investigate how fast the algebraic solver converges and how this changes with respect to the number of processes. This boils down to the question to which extend the modified smoothers degrades the solver's performance (we use the ℓ_1 smoothers described in Section 4.3, both in a nonlinear way as the first step of the TNNMG algorithm and in the linear way as part of the multigrid step). Note that we did not implement load-balancing, yet. Therefore, in particular in later stages of the adaptive algorithm, the number of unknowns per process might be unevenly distributed. In Figure 7.5.1, we computed the average convergence rate each time

Figure 7.5.1.: Average Convergence Rates



we had to solve an algebraic system. For each of those systems, we started with an initial guess u_0 (as explained before, initialized from interpolated solutions of coarser grids, if available) and we computed iterates $u_1, \ldots, u_k$ by applying TNNMG steps until a threshold $\|u_k - u^*\| < 10^{-8}$ was met. The average convergence rate c_{avg} was

Figure 7.5.2.: Number of TNNMG Steps k



computed by

$$c_{\text{avg}} = \sqrt[k]{\frac{\|u_k - u^*\|}{\|u_0 - u^*\|}}.$$

Here, u^* denotes an "exact" algebraic solution, which we approximate by a precomputed solution with a much lower tolerance. Figure 7.5.1 shows that a higher number of cores (and thus higher number of parallel blocks) does indeed have a negative impact on the convergence rate. The number of steps k needed to achieve the given accuracy is shown in Figure 7.5.2. This plot confirms our finding that a higher number of processes slows down the algebraic solver and we cannot claim "strong scaling" in this respect. It has to be noted, though, that the number of TNNMG steps suffer only moderately, therefore the algorithm is still useful, in particular if performance gains can be achieved.

Remark 7.15. Strong scaling is often considered with respect to run times, i. e. a perfectly strong scaling algorithm would take only $1/N$-th of the time when using N processes compared to a single process.

We do not think that run time is a suitable measure here since it depends on a lot of external factors like hardware, the specific software used, network architecture etc.

These are beyond our control and can only be a snapshot of the capabilities at a given time.

8. Conclusion

We have seen that the discretization of variational inequalities with higher order DG elements is indeed possible and (using the *hp*-adaptive approach) gives higher order convergence rates with respect to the number of unknowns. Hardly surprising, the higher order methods lead to a better convergence rate compared to piecewise linear finite element functions which did not yield convergence rates beyond the $\mathcal{O}(n)$ rate. Interestingly, we can obtain convergence rates better than the $\mathcal{O}(n^{1.5})$ rates from piecewise quadratic finite element functions. These are well known for the obstacle problem [38, 111, 112]. The *hp*-adaptive method is both theoretically (cf. Chapter 3) and practically (cf. Chapter 7) able to beat these in terms of error per unknown. Beyond the obstacle problem, we can use the methods also for variational inequalities of the second kind, which we demonstrated for a timestep of the time-discretized Allen–Cahn equation with logarithmic potential. Other nonlinearities can be treated in the same manner, cf. Section 3.4. Thus we developed an efficient framework which covers discretization, solution of the algebraic system and an adaptive process to approximate solutions to variational inequalities. Since the arising systems for DG discretizations (in particular with higher order) are denser than for a continuous piecewise linear finite element discretization, we have to make sure that the implementation can leverage the better convergence rates to make up for the higher arithmetic loads per unknown. Luckily, the blocked structures of DG bases allow for many optimizations such as fast arithmetic, good use of caches and easier parallelization. In conclusion, we can say that (given a suitable implementation), adaptive discontinuous Galerkin methods can be used as an efficient way to solve variational inequalities numerically.

Appendices

A. Details

In the following, we will derive a condition under which Assumption 3.49 is true. For simplicity, we will restrict ourselves to the case of a lower obstacle. The criterion is based on bounding the difference of the residual $f - L\underline{\psi}$ to its integral mean in a L^p norm, making it similar to classical oscillation assumptions.

Note that, however, this condition is in general stronger than required. A simple counter example would be $f - L\underline{\psi}$ being a linear function.

Lemma A.1. *For $K_0 \in \mathcal{T}^l$, assume there is $s \in [1, \infty]$, such that*

$$\left\| |K_0| \left((f - L\underline{\psi} - P^0_{K_0}(f - L\underline{\psi}) \right) \right\|_{L^s(K_0)} \leq \beta^s \left\| (f - L\underline{\psi}) \right\|_{L^1(K_0)}. \tag{A.1}$$

The basis-dependent constant β^s is defined through

$$\beta^s = \min_{\varphi_p} \frac{\int_{K_0} \varphi_p \,\mathrm{d}x}{\left\| \varphi_p \right\|_{L^t(K_0)}} \tag{A.2}$$

where the minimum is taken over the basis functions φ_p defined on K_0 and t is defined through $1/t + 1/s = 1$ (using the usual convention that $t = \infty$ if $s = 1$ and vice versa).

Then, Assumption 3.49 is true, i. e.

$$\int_{K_0} \left(f - L\underline{\psi} \right) \varphi_p \,\mathrm{d}x \leq 0.$$

Proof. First, note that since $(f - L\underline{\psi}) \leq 0$, we have

$$P^0_K(f - L\underline{\psi}) = \frac{1}{|K|} \int_K f - L\underline{\psi} \,\mathrm{d}x = -\frac{1}{|K|} \int_K |f - L\underline{\psi}| \,\mathrm{d}x = -\frac{1}{|K|} \left\| f - L\underline{\psi} \right\|_{L^1(K)}.$$

Let t again denote the conjugate exponent to s. Observing that $\omega_p = \int_K \varphi_p \,\mathrm{d}x > 0$

by construction of the basis functions, we deduce

$$\begin{aligned}\int_K (f - L\underline{\psi})\varphi_p \,\mathrm{d}x &= \int_K P_K^0(f - L\underline{\psi})\varphi_p + R_K^0(f - L\underline{\psi})\varphi_p \,\mathrm{d}x \\ &= P_K^0(f - L\underline{\psi})\omega_p + \int_K R_K^0(f - L\underline{\psi})\varphi_p \,\mathrm{d}x \\ &\le P_K^0(f - L\underline{\psi})\omega_p + \left\| R_K^0(f - L\underline{\psi}) \right\|_{L^s(K)} \|\varphi_p\|_{L^t(K)} \\ &= P_K^0(f - L\underline{\psi})\omega_p + |K|^{-1}\left\| |K| R_K^0(f - L\underline{\psi}) \right\|_{L^s(K)} \|\varphi_p\|_{L^t(K)} \\ &\le P_K^0(f - L\underline{\psi})\omega_p + |K|^{-1}\beta^s \left\| f - L\underline{\psi} \right\|_{L^1(K)} \|\varphi_p\|_{L^t(K)} \quad \text{(by (A.1))} \\ &\le P_K^0(f - L\underline{\psi})\omega_p + |K|^{-1} \left\| f - L\underline{\psi} \right\|_{L^1(K)} \omega_p \quad \text{(by (A.2))} \\ &= P_K^0(f - L\underline{\psi})\omega_p - P_K^0(f - L\underline{\psi})\omega_p \\ &= 0.\end{aligned}$$

□

Lemma A.2. *Let $u \in H^s(\Omega)$ with $s \in (0,1)$. Then, we have*

$$u^+ = \max(u, 0) \in H^s(\Omega).$$

Proof. Let f be a smooth function on Ω. Clearly, $f^+ \in L^2(\Omega)$ as f^+ is dominated by f. Let Ω^+ and Ω^0 be the parts where f is positive and nonpositive, respectively. Now, we have

$$\begin{aligned}\|f^+\|_{H^s(\Omega)}^2 &= \|f^+\|_0^2 + \int_\Omega \int_\Omega \frac{|f^+(x) - f^+(y)|^2}{|x-y|^{N+2s}} \,\mathrm{d}x \,\mathrm{d}y \\ &= \|f^+\|_0^2 + \int_{\Omega^+} \int_{\Omega^+} \frac{|f^+(x) - f^+(y)|^2}{|x-y|^{N+2s}} \,\mathrm{d}x \,\mathrm{d}y \\ &\quad + 2\int_{\Omega^0} \int_{\Omega^+} \frac{|f^+(x)|^2}{|x-y|^{N+2s}} \,\mathrm{d}x \,\mathrm{d}y.\end{aligned}$$

The first double integral is clearly dominated by $|f|_s^2$. For the second double integral, we note that we that for $x \in \Omega^+$ and $y \in \Omega^0$, we have $u(y) \le 0$ and $u(x) > 0$ and thus we have the pointwise estimate

$$|u(x)|^2 \le |u(x) - u(y)|^2,$$

showing that the second double integral is indeed also bounded by $|f|_s^2$. Using the usual densitiy argument, the proof is finished. □

Remark A.3. Similar to the preceding lemma, it can be shown, that for $u \in W^{m,p}$ with $m \in (0, 1 + 1/m)$, one has $|u| \in W^{m,p}$ [29]. In [102], the case for u^+ is discussed for several spaces.

Continuity of Preconditioned Nonlinear Gauss–Seidel

In the following, we want to lay ground for the proof of Lemma 4.10. We are minimizing the following functional: Let $j(v) = \sum j_i(v_i)$.

$$\tilde{D}_x(v) = \frac{1}{2}B(v,v) - \langle b - Ax, v\rangle + j(x+v).$$

Using $B(v,v) = \langle Bv, v\rangle$ for $B \in \mathbb{R}^{n\times n}$ s. p. d. Here, we have replaced the matrix A by B in the quadratic part. Equivalently, we solve the variational inequality

$$B(u, v-u) - \langle b - Ax, v-u\rangle + j(x+v) - j(x+u) \geq 0.$$

We see, the shift in the linear part is with respect to x but using A, not B. When applying Gauss–Seidel, we have that intermediate iterates will be shifted by x plus the former corrections. Say $y = \sum_{j=1}^{i-1} y_j e_j$ contains all corrections made up to the i-th subspace, then we have

$$y_i = \operatorname*{arg\,min}_{u\in\mathbb{R}} \tilde{D}_x(y + Pu),$$

where $P : \mathbb{R} \to \mathbb{R}^n$, $v \mapsto ve_i$ is the prolongation operator. Similarly, define $R : \mathbb{R}^n \to \mathbb{R}$, $Rv = v_i$, as the restriction to the subspace of the i-th component.

As a variational inequality, this reads

$$\begin{aligned} B(Pu, P(v-u)) &- \langle b - Ax - By, P(v-u)\rangle \\ &+ j_i(Rx + Ry + v) - j_i(Rx + Ry + u) \geq 0. \end{aligned} \tag{A.3}$$

Lemma A.4. *Let u and $\tilde{u}$ be solutions of* (A.3) *with respect to the shifts* (x, y) *and* $(\tilde{x}, \tilde{y})$ *respectively. Then, we have there is a C such that*

$$\|u - \tilde{u}\| \leq C \left(\|x - \tilde{x}\| + \|y - \tilde{y}\|\right).$$

Proof. First, define $z = y + B^{-1}Ax$ and $\tilde{z} = \tilde{y} + B^{-1}A\tilde{x}$. For shorter notation, introduce $d_u = \tilde{u} - u$, $d_x = \tilde{x} - x$, $d_y = \tilde{y} - y$ and $d_z = \tilde{z} - z$.

Insert $v = R(\tilde{x} - x) + R(\tilde{y} - y) + \tilde{u}$ in the VI for u and $v = R(x - \tilde{x}) + R(y - \tilde{y}) + u$ in the VI for $\tilde{u}$.

This gives

$$\begin{aligned} B(Pu, PRd_x + PRd_y + Pd_u) &- \langle b - Bz, PRd_x + PRd_y + Pd_u\rangle \\ &+ j_i(R\tilde{x} + R\tilde{y} + \tilde{u}) - j_i(Rx + Ry_i + u) \qquad \geq 0 \\ -B(P\tilde{u}, PRd_x + PRd_y + Pd_u) &+ \langle b - B\tilde{z}, PRd_x + PRd_y + Pd_u\rangle \\ &- j_i(R\tilde{x} + R\tilde{y} + \tilde{u}) + j_i(Rx + Ry + u) \qquad \geq 0. \end{aligned}$$

Adding both, we have

$$B(-Pd_u, PRd_x + PRd_y + Pd_u) - \langle B(\tilde{z} - z), PRd_x + PRd_y + Pd_u\rangle \geq 0.$$

Or, using $B(\cdot,\cdot) = \langle B\cdot, \cdot\rangle$,

$$B(-Pd_u - d_z, PRd_x + PRd_y + Pd_u) \geq 0.$$

Adding $0 = -PRd_x - PRd_y + PRd_x + PRd_y$ on the left hand side, we get

$$B(PR(d_x+d_y)-d_z, P(R(d_x+d_y)+d_u)) \geq B(P(R(d_x+d_y)+d_u), P(R(d_x+d_y)+d_u)).$$

Thus

$$\begin{aligned}\gamma_B\|P(R(d_x+d_y)+d_u)\|^2 &\leq B(P(R(d_x+d_y)+d_u), P(R(d_x+d_y)+d_u))\\ &\leq B(PR(d_x+d_y)-d_z, P(R(d_x+d_y)+d_u))\\ &\leq \Gamma_B\|PR(d_x+d_y)-d_z\|\|P(R(d_x+d_y)+d_y)\|.\end{aligned}$$

Dividing by $\gamma_B\|P(R(d_x+d_y)+d_u)\|$ on both sides, we get

$$\|P(R(d_x+d_y)+d_u)\| \leq \frac{\Gamma_B}{\gamma_B}\|PR(d_x+d_y)-d_z\|. \tag{A.4}$$

Now, we can finally estimate:

$$\begin{aligned}\|d_u\| &= \|RPd_u\|\\ &= \|R\|\|Pd_u\|\\ &\leq \|R\|\left(\|P(R(d_x+d_y)+d_u)\| + \|PR(d_x+d_y)\|\right)\\ &\leq \|R\|\left(\frac{\Gamma_B}{\gamma_B}\|PR(d_x+d_y)-d_z\| + \|PR(d_x+d_y)\|\right)\\ &\leq C(\|d_x\|+\|d_y\|+\|d_z\|)\end{aligned}$$

The last estimate is possible since both P and R are bounded linear operators. It remains to show $\|d_z\| \leq C(\|d_x\|+\|d_y\|)$.

$$\begin{aligned}\|d_z\| &= \|\tilde{z}-z\|\\ &= \|\tilde{y}+B^{-1}A\tilde{x}-y+B^{-1}Ax\|\\ &= \|d_y+B^{-1}Ad_x\|\\ &\leq \|d_y\| + ||B^{-1}A\|\|d_x\|\\ &\leq C(\|d_x\|+\|d_y\|).\end{aligned}$$

□

B. Zusammenfassung

In dieser Arbeit wird dargestellt, wie sogenannte Discontinuous Galerkin (DG) Methoden für die numerische Lösung von Variationsungleichungen eingesetzt werden können. Dazu werden zunächst die mathematischen Grundlagen eingeführt, insbesondere ebendiese Variationsungleichungen sowie Anwendungen im Bereich von Phasenfeldgleichungen. Danach werden die Discontinuous Galerkin Methoden erklärt sowie gezeigt, wie man diese nutzen kann, um Hindernisprobleme (welche einen wichtigen Spezialfall von Variationsungleichungen darstellen und hier als Modellproblem dienen) zu diskretisieren. Hindernisprobleme haben die Eigenschaft, dass ihre Lösungen selbst für glatte Daten nur eine beschränkte Regularität besitzen. Somit ist der Einsatz von Methoden mit höherer Ordnung zumindest fragwürdig, da deren Effizienz maßgeblich an der Glattheit der zu approximierenden Lösung hängt. Es wird jedoch gezeigt, wie durch geschickte Wahl des Gitters der Einsatz von DG Funktionen höherer Ordnung zu einer Diskretisierung führt, deren theoretische Konvergenzordnung höher ist als die bisher bekannten Schranken. Ferner wird erklärt, wie die für das Hindernisproblem eingeführte Diskretisierung auf allgemeinere Variationsungleichungen verallgemeinert werden kann.

Im praktischen Einsatz müssen die entstehenden algebraischen Probleme natürlich erst einmal gelöst werden. Dazu wird beschrieben, wie die Truncated Nonsmooth Newton Multigrid (TNNMG) Methode für die betrachteten Probleme eingesetzt werden kann, um möglichst schnell zu einer numerischen Lösung zu kommen. Ein wichtiger Baustein des TNNMG Verfahrens ist ein schneller linearer Löser, der mittels eines geometrischen Mehrgitterverfahrens für DG Systeme realisiert wird. Darüber hinaus wird in der Arbeit ein nichtlinearer Glätter entwickelt, der für die Konvergenz des TNNMG Verfahrens in einer parallelisierten Implementierung geeignet ist.

Da die eingangs erwähnten "geschickt gewählten" Gitter jedoch von der (unbekannten) analytischen Lösung abhängig sind, wird eine adaptive Prozedur beschrieben, welche die glatten und nicht glatten Teile der Lösung identifiziert und jeweils mit Funktionen höherer Ordnung bzw. einem feineren Gitter auflöst (man spricht von hp-Adaptivität). Um die Fehler der aktuellen Zwischenlösung zu approximieren und lokalisieren, wird ein hierarchischer Fehlerschätzer benutzt, der auf den DG Fall angepasst wurde und dessen Effizienz sowie Verlässlichkeit für ein lineares Modellproblem bewiesen wird.

Am Ende werden die eingeführten Methoden numerisch an einer Reihe von Beispielen getestet. Dabei wird auf verschiedene Aspekte des Prozesses eingegangen und gezeigt, dass die hp-adaptive DG Variante einer stückweise linearen oder stückweise quadratischen Diskretisierung überlegen ist. Insbesondere kann die Konvergenz mit höherer Ordnung experimentell nachgewiesen werden, obwohl die Regularität der analytischen Lösung dieses nicht direkt erwarten lässt.

Bibliography

[1] Robert A. Adams and John J. F. Fournier. *Sobolev Spaces.* Academic Press, second edition, 2003.

[2] Samuel M. Allen and John W. Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095, June 1979.

[3] Marreddy Ambati, Tymofiy Gerasimov, and Laura De Lorenzis. A review on phase-field models of brittle fracture and a new fast hybrid formulation. *Computational Mechanics*, 55(2):383–405, 2015.

[4] Luigi Ambrosio and Vincenzo Maria Tortorelli. Approximation of functional depending on jumps by elliptic functional via Γ-convergence. *Communications on Pure and Applied Mathematics*, 43(8):999–1036, 1990.

[5] Luigi Ambrosio and Vincenzo Maria Tortorelli. On the approximation of free discontinuity problems. 1992.

[6] Paola F. Antonietti and Paul Houston. A class of domain decomposition preconditioners for hp-discontinuous Galerkin finite element methods. *Journal of Scientific Computing*, 46(1):124–149, Jan 2011.

[7] Paola F. Antonietti, Paul Houston, and Iain Smears. A note on optimal spectral bounds for nonoverlapping domain decomposition preconditioners for hp-version discontinuous Galerkin methods. *International Journal of Numerical Analysis and Modeling*, 2015.

[8] Paola F. Antonietti, Marco Sarti, and Marco Verani. Multigrid algorithms for hp-discontinuous Galerkin discretizations of elliptic problems. *SIAM Journal on Numerical Analysis*, 53(1):598–618, 2015.

[9] Paola F. Antonietti, Marco Sarti, and Marco Verani. Multigrid algorithms for high order discontinuous Galerkin methods. In *Lecture Notes in Computational Science and Engineering*, pages 3–13. Springer International Publishing, 2016.

[10] Douglas N. Arnold. *An interior penalty finite element method with discontinuous elements.* PhD thesis, University of Chicago, 1979.

[11] Douglas N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19(4):742–760, 1982.

[12] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.

[13] C Baiocchi, V Comincioli, E Magenes, and GA Pozzi. Free boundary problems in the theory of fluid flow through porous media: existence and uniqueness theorems. *Annali di Matematica Pura ed Applicata*, 97(1):1–82, 1973.

[14] Allison H. Baker, Robert D. Falgout, Tzanio V. Kolev, and Ulrike Meier Yang. Multigrid smoothers for ultraparallel computing. *SIAM Journal on Scientific Computing*, 33(5):2864–2887, January 2011.

[15] Lothar Banz and Ernst P. Stephan. A posteriori error estimates of hp-adaptive IPDG-FEM for elliptic obstacle problems. *Applied Numerical Mathematics*, 76:76 – 92, 2014.

[16] S. Bartels and C. Carstensen. Averaging techniques yield reliable a posteriori finite element error control for obstacle problems. *Numerische Mathematik*, 99(2):225–249, 2004.

[17] Peter Bastian. *Parallele adaptive Mehrgitterverfahren.* Vieweg Teubner Verlag, 1996.

[18] Peter Bastian, Markus Blatt, Andreas Dedner, Nils-Arne Dreier, Christian Engwer, René Fritze, Carsten Gräser, Christoph Grüninger, Dominic Kempf, Robert Klöfkorn, et al. The DUNE framework: basic concepts and recent developments. *Computers & Mathematics with Applications*, 81:75–112, 2021.

[19] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Ralf Kornhuber, Mario Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. part ii: implementation and tests in DUNE. *Computing*, 82(2):121–138, Jul 2008.

[20] Peter Bastian, Markus Blatt, Andreas Dedner, Christian Engwer, Robert Klöfkorn, Mario Ohlberger, and Oliver Sander. A generic grid interface for parallel and adaptive scientific computing. part i: abstract framework. *Computing*, 82(2):103–119, Jul 2008.

[21] Peter Bastian, Markus Blatt, and Robert Scheichl. Algebraic multigrid for discontinuous Galerkin discretizations of heterogeneous elliptic problems. *Numerical Linear Algebra with Applications*, 19(2):367–388, 2012.

[22] Robert E. Bird, William M. Coombs, and Stefano Giani. A posteriori discontinuous Galerkin error estimator for linear elasticity. *Applied Mathematics and Computation*, 344-345:78–96, March 2019.

[23] Luise Blank, Harald Garcke, Lavinia Sarbu, and Vanessa Styles. Primal-dual active set methods for Allen–Cahn variational inequalities with nonlocal constraints. *Numerical methods for partial differential equations*, 29(3):999–1030, 2013.

[24] Markus Blatt and Peter Bastian. The iterative solver template library. In Bo Kågström, Erik Elmroth, Jack Dongarra, and Jerzy Waśniewski, editors, *Applied Parallel Computing. State of the Art in Scientific Computing*, pages 666–675, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[25] Markus Blatt and Peter Bastian. On the generic parallelisation of iterative solvers for the finite element method. *Int. J. Comput. Sci. Engrg.*, 4:56–69, 01 2008.

[26] James F. Blowey and Charles M. Elliott. Curvature dependent phase boundary motion and parabolic double obstacle problems. In Wei-Ming Ni, L. A. Peletier, and J. L. Vazquez, editors, *Degenerate Diffusions*, pages 19–60, New York, NY, 1993. Springer New York.

[27] Folkmar A. Bornemann. An adaptive multilevel approach to parabolic equations I. General theory and 1D implementation. *IMPACT of Computing in Science and Engineering*, 2(4):279–317, Dec 1990.

[28] Folkmar A. Bornemann, Bodo Erdmann, and Ralf Kornhuber. A posteriori error estimates for elliptic problems in two and three space dimensions. *SIAM J. Numer. Anal.*, 33(3):1188 – 1204, 1996.

[29] Gérard Bourdaud and Yves Meyer. Fonctions qui opèrent sur les espaces de sobolev. *Journal of Functional Analysis*, 97(2):351 – 360, 1991.

[30] Blaise Bourdin. Image segmentation with a finite element method. *ESAIM: Mathematical modelling and numerical analysis*, 33(2):229–244, 1999.

[31] Dietrich Braess. *Finite Elemente.* Springer-Verlag Berlin Heidelberg, 4 edition, 2007.

[32] Dietrich Braess, Carsten Carstensen, and Ronald HW Hoppe. Convergence analysis of a conforming adaptive finite element method for an obstacle problem. *Numerische Mathematik*, 107(3):455–471, 2007.

[33] Achi Brandt and Colin W. Cryer. Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems. *SIAM Journal on Scientific and Statistical Computing*, 4(4):655–684, December 1983.

[34] Susanne C. Brenner, J. Cui, T. Gudi, and L.-Y. Sung. Multigrid algorithms for symmetric discontinuous Galerkin methods on graded meshes. *Numerische Mathematik*, 119(1):21–47, April 2011.

[35] Susanne C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods.* Springer, third edition, 2008.

[36] Haïm Brezis. *Opérateurs maximaux monotones et semi-groupes de contractions dans les espaces de Hilbert.* North-Holland Pub. Co., Amsterdam, 1973.

[37] Haïm R. Brezis and Guido Stampacchia. Sur la régularité de la solution d'inéquations elliptiques. *Bull. Soc. Math. France*, 96:153–180, 1968.

[38] Franco Brezzi, William W. Hager, and P. A. Raviart. Error estimates for the finite element solution of variational inequalities. *Numerische Mathematik*, 28(4):431–443, Dec 1977.

[39] Kolja Brix, Martin Campos Pinto, and Wolfgang Dahmen. A multilevel preconditioner for the interior penalty discontinuous Galerkin method. *SIAM J. Numer. Anal.*, 46(5):2742–2768, July 2008.

[40] Markus Bürg and Willy Dörfler. Convergence of an adaptive hp finite element strategy in higher space-dimensions. *Applied numerical mathematics*, 61(11):1132–1146, 2011.

[41] Xinfu Chen and Charles M. Elliott. Asymptotics for a parabolic double obstacle problem. *Proceedings.*, 444(1922):429–445, 1994.

[42] Philippe G Ciarlet. *The finite element method for elliptic problems / Philippe G. Ciarlet.* Society for Industrial and Applied Mathematics, Philadelphia, PA, unabridged republ. of the work first publ. by North-Holland, 1978 edition, 2002.

[43] Klaus Deckelnick, Gerhard Dziuk, and Charles M Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta numerica*, 14:139–232, 2005.

[44] Andreas Dedner, Robert Klöfkorn, Martin Nolte, and Mario Ohlberger. A generic interface for parallel and adaptive discretization schemes: abstraction principles and the dune-fem module. *Computing*, 90(3-4):165–196, August 2010.

[45] Leszek Demkowicz, Waldemar Rachowicz, and Ph Devloo. A fully automatic hp-adaptivity. *Journal of Scientific Computing*, 17(1):117–142, 2002.

[46] Peter Deuflhard, Peter Leinen, and Harry Yserentant. Concepts of an adaptive hierarchical finite element code. *IMPACT of Computing in Science and Engineering*, 1(1):3 – 35, 1989.

[47] JK Djoko. Discontinuous Galerkin finite element methods for variational inequalities of first and second kinds. *Numerical Methods for Partial Differential Equations: An International Journal*, 24(1):296–311, 2008.

[48] W Dörfler and V Heuveline. Convergence of an adaptive hp finite element strategy in one space dimension. *Applied numerical mathematics*, 57(10):1108–1124, 2007.

[49] Willy Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, June 1996.

[50] Willy Dörfler and Ricardo H Nochetto. Small data oscillation implies the saturation assumption. *Numerische Mathematik*, 91(1):1–12, 2002.

[51] Christian Engwer, Carsten Gräser, Steffen Müthing, and Oliver Sander. The interface for functions in the dune-functions module. *Archive of Numerical Software*, Vol 5:No 1 (2017), 2017.

[52] Christian Engwer, Carsten Gräser, Steffen Müthing, and Oliver Sander. Function space bases in the dune-functions module. 2018.

[53] Yekaterina Epshteyn and Béatrice Rivière. Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *Journal of Computational and Applied Mathematics*, 206(2):843–872, sep 2007.

[54] Lawrence C. Evans. *Partial Differential Equations.* American Mathematical Society, 1998.

[55] Thomas Fankhauser, Thomas P. Wihler, and Marcel Wirz. The hp-adaptive FEM based on continuous sobolev embeddings: Isotropic refinements. *Computers & Mathematics with Applications*, 67(4):854–868, March 2014.

[56] Xiaobing Feng and Andreas Prohl. Numerical analysis of the Allen-Cahn equation and approximation for mean curvature flows. *Numerische Mathematik*, 94(1):33–65, Mar 2003.

[57] Roland Glowinski. *Numerical Methods for Nonlinear Variational Problems.* Springer Series in Computational Physics. Springer-Verlag, 1984.

[58] Roland Glowinski, Jacques-Louis Lions, and Raymond Trémolières. *Numerical Analysis of Variational Inequalities*, volume 8 of *Studies in Mathematics and Its Applications.* Elsevier, 1981.

[59] Jayadeep Gopalakrishnan and Guido Kanschat. A multilevel discontinuous Galerkin method. *Numerische Mathematik*, 95(3):527–550, 2003.

[60] Jochen Gorski, Frank Pfeuffer, and Kathrin Klamroth. Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical methods of operations research*, 66(3):373–407, 2007.

[61] Carsten Gräser. *Convex minimization and phase field models.* PhD thesis, Freie Universität Berlin, 2011.

[62] Carsten Gräser, Max Kahnt, and Ralf Kornhuber. Numerical approximation of multi-phase Penrose-Fife systems. *Comput. Meth. in Appl. Math.*, 16:523–542, 2016.

[63] Carsten Gräser, Daniel Kienle, and Oliver Sander. Truncated nonsmooth Newton multigrid for phase-field brittle-fracture problems. 2020.

[64] Carsten Gräser and Ralf Kornhuber. Multigrid methods for obstacle problems. *Journal of Computational Mathematics*, pages 1–44, 2009.

[65] Carsten Gräser, Ralf Kornhuber, and Uli Sack. Time discretizations of anisotropic Allen–Cahn equations. *IMA Journal of Numerical Analysis*, 33(4):1226–1244, March 2013.

[66] Carsten Gräser and Oliver Sander. The dune-subgrid module and some applications. *Computing*, 86(4):269, 2009.

[67] Carsten Gräser and Oliver Sander. Truncated nonsmooth Newton multigrid methods for block-separable minimization problems. *IMA Journal of Numerical Analysis*, 39(1):454–481, 11 2018.

[68] Wolfgang Hackbusch. *Iterative solution of large sparse systems of equations*, volume 95. Springer, 1994.

[69] Michael Hintermüller, Kazufumi Ito, and Karl Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.

[70] Michael Hintermüller, Steven-Marian Stengl, and Thomas M. Surowiec. Uncertainty quantification in image segmentation using the Ambrosio–Tortorelli approximation of the Mumford–Shah energy. *Journal of Mathematical Imaging and Vision*, 63(9):1095–1117, July 2021.

[71] Paul Houston and Endre Süli. A note on the design of hp-adaptive finite element methods for elliptic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 194:229–243, February 2005.

[72] Ohannes A. Karakashian and Frederic Pascal. A posteriori error estimates for a discontinuous Galerkin approximation of second-order elliptic problems. *SIAM J. Numerical Analysis*, 41:2374–2399, 01 2003.

[73] Noboru Kikuchi and John Tinsley Oden. *Contact problems in elasticity: a study of variational inequalities and finite element methods.* SIAM, 1988.

[74] David Kinderlehrer and Guido Stampacchia. *An Introduction to Variational Inequalities and their Applications.* Academic Press, Inc, 1980.

[75] Ralf Kornhuber. Monotone multigrid methods for elliptic variational inequalities i. *Numerische Mathematik*, 69(2):167–184, December 1994.

[76] Ralf Kornhuber. A posteriori error estimates for elliptic variational inequalities. *Computers & Mathematics with Applications*, 31(8):49–60, 1996.

[77] Ralf Kornhuber and Qingsong Zou. Efficient and reliable hierarchical error estimates for the discretization error of elliptic obstacle problems. *Mathematics of Computation*, 80(273):69–88, jun 2010.

[78] Andreas Krebs and Ernst P. Stephan. A p-version finite element method for nonlinear elliptic variational inequalities in 2D. *Numerische Mathematik*, 105(3):457–480, 2007.

[79] Martin Kronbichler and Katharina Kormann. A generic interface for parallel cell-based finite element operator application. *Computers & Fluids*, 63:135 – 147, 2012.

[80] Martin Kronbichler and Katharina Kormann. Fast matrix-free evaluation of discontinuous Galerkin finite element operators. *ACM Transactions on Mathematical Software*, 45(3):1–40, September 2019.

[81] Martin Kronbichler, Katharina Kormann, Igor Pasichnyk, and Momme Allalen. Fast matrix-free discontinuous Galerkin kernels on modern computer architectures. In Julian M. Kunkel, Rio Yokota, Pavan Balaji, and David Keyes, editors, *High Performance Computing*, pages 237–255, Cham, 2017. Springer International Publishing.

[82] J. L. Lions and G. Stampacchia. Variational inequalities. *Communications on Pure and Applied Mathematics*, 20(3):493–519, 1967.

[83] Jan Mandel. A multilevel iterative method for symmetric, positive definite linear complementarity problems. *Applied Mathematics & Optimization*, 11(1):77–95, February 1984.

[84] Brendan S. Mascarenhas, Brian T. Helenbrook, and Harold L. Atkins. Application of p-multigrid to discontinuous Galerkin formulations of the Euler equations. *AIAA Journal*, 47(5):1200–1208, May 2009.

[85] Jens Markus Melenk and Barbara I Wohlmuth. On residual-based a posteriori error estimation in hp-fem. *Advances in Computational Mathematics*, 15(1):311–331, 2001.

[86] William F Mitchell and Marjorie A McClain. A comparison of hp-adaptive strategies for elliptic partial differential equations. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1–39, 2014.

[87] David Bryant Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 1989.

[88] Steffen Müthing, Marian Piatkowski, and Peter Bastian. High-performance implementation of matrix-free high-order discontinuous Galerkin methods. arXiv:1711.10885, 2017.

[89] Eleonora Di Nezza, Giampiero Palatucci, and Enrico Valdinoci. Hitchhiker's guide to the fractional Sobolev spaces. *Bulletin des Sciences Mathématiques*, 136(5):521–573, July 2012.

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

[91] Muhammad Aslam Noor. Some developments in general variational inequalities. *Applied Mathematics and Computation*, 152(1):199–277, 2004.

[92] Elias Pipping, Oliver Sander, and Ralf Kornhuber. Variational formulation of rate-and state-dependent friction problems. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 95(4):377–395, 2015.

[93] Alfio Quarteroni and Alberto Valli. *Numerical Approximation of Partial Differential Equations.* Springer Science & Business, 2009.

[94] Beatrice Rivière. *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation.* Society for Industrial and Applied Mathematics, 2008.

[95] Béatrice Riviere and Mary F Wheeler. A posteriori error estimates for a discontinuous Galerkin method applied to elliptic problems. log number: R74. *Computers & Mathematics with Applications*, 46(1):141–163, 2003.

[96] José-Francisco Rodrigues. *Obstacle Problems in Mathematical Physics.* North-Holland, 1987.

[97] Erich Rothe. Zweidimensionale parabolische Randwertaufgaben als Grenzfall eindimensionaler Randwertaufgaben. *Mathematische Annalen*, 102(1):650–670, 1930.

[98] Jacob Rubinstein, Peter Sternberg, and Joseph B. Keller. Fast reaction, slow diffusion, and curve shortening. *SIAM Journal on Applied Mathematics*, 49(1):116–133, February 1989.

[99] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[100] Uli Sack. *Numerical Simulation of Phase Separation in Binary and Multicomponent Systems.* PhD thesis, Freie Universität Berlin, 2014.

[101] Oliver Sander. *DUNE—The Distributed and Unified Numerics Environment*, volume 140. Springer Nature, 2020.

[102] Giuseppe Savaré. On the regularity of the positive part of functions. *Nonlinear Analysis: Theory, Methods & Applications*, 27(9):1055 – 1074, 1996.

[103] Jie Shen and Xiaofeng Yang. Numerical approximations of Allen-Cahn and Cahn-Hilliard equations. *Discrete & Continuous Dynamical Systems - A*, 28(4):1669–1691, 2010.

[104] Iain Smears. Nonoverlapping domain decomposition preconditioners for discontinuous Galerkin approximations of Hamilton–Jacobi–Bellman equations. *Journal of Scientific Computing*, 74(1):145–174, Jan 2018.

[105] Ingo Steinbach. Phase-field models in materials science. *Modelling and simulation in materials science and engineering*, 17(7):073001, 2009.

[106] Jörg Stiller. Robust multigrid for high-order discontinuous Galerkin methods: A fast Poisson solver suitable for high-aspect ratio cartesian grids. *Journal of computational physics*, 327, 2016-12.

[107] Xue-Cheng Tai. Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numerische Mathematik*, 93(4):755–786, February 2003.

[108] Mark A Taylor, Beth A Wingate, and Rachel E Vincent. An algorithm for computing Fekete points in the triangle. *SIAM Journal on Numerical Analysis*, 38(5):1707–1720, 2000.

[109] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*. Springer Series in Computational Mathematics. Springer Berlin Heidelberg, 2009.

[110] Fei Wang. Discontinuous Galerkin methods for solving double obstacle problem. *Numerical Methods for Partial Differential Equations*, 29(2):706–720, June 2012.

[111] Fei Wang, Weimin Han, and Xiao-Liang Cheng. Discontinuous Galerkin methods for solving elliptic variational inequalities. *SIAM Journal on Numerical Analysis*, 48(2):708–733, 2010.

[112] Lie-Heng Wang. On the quadratic finite element approximation to the obstacle problem. *Numerische Mathematik*, 92(4):771–778, Oct 2002.

[113] Mary Fanett Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM Journal on Numerical Analysis*, 15(1):152–161, February 1978.

[114] Thomas P. Wihler. An hp-adaptive strategy based on continuous Sobolev embeddings. *Journal of Computational and Applied Mathematics*, 235(8):2731–2739, February 2011.

[115] Joseph Wloka. *Partielle Differentialgleichungen*. B. G. Teubner Stuttgart, 1982.

[116] Qingsong Zou, Andreas Veeser, Ralf Kornhuber, and Carsten Gräser. Hierarchical error estimates for the energy functional in obstacle problems. *Numerische Mathematik*, 117(4):653–677, 2011.