# A Multigrid Method for the Efficient Numerical Solution of Optimization Problems Constrained by Partial Differential Equations

**Dissertation**

zur

Erlangung des Doktorgrades (Dr. rer. nat.)

der

Mathematisch–Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich–Wilhelms–Universität Bonn

vorgelegt von

Martin Engel

aus

Brühl

Bonn 2009

Angefertigt mit Genehmigung der Mathematisch–Naturwissenschaftlichen Fakultät der Rheinischen Friedrich–Wilhelms–Universität Bonn

1. Gutachter: Prof. Dr. Michael Griebel

2. Gutachter: Prof. Dr. Sören Bartels

Tag der Promotion: 15.05.2009

Erscheinungsjahr: 2009

Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn **http://hss.ulb.uni-bonn.de/diss_online** elektronisch publiziert.

# Abstract

We study the minimization of a quadratic functional subject to constraints given by a linear or semilinear elliptic partial differential equation with distributed control. Further, pointwise inequality constraints on the control are accounted for.

In the linear-quadratic case, the discretized optimality conditions yield a large, sparse, and indefinite system with saddle point structure. One main contribution of this thesis consists in devising a coupled multigrid solver which avoids full constraint elimination. To this end, we define a smoothing iteration incorporating elements from constraint preconditioning. A local mode analysis shows that for discrete optimality systems, we can expect smoothing rates close to those obtained with respect to the underlying constraint PDE.

Our numerical experiments include problems with constraints where standard pointwise smoothing is known to fail for the underlying PDE. In particular, we consider anisotropic diffusion and convection-diffusion problems. The framework of our method allows to include line smoothers or ILU-factorizations, which are suitable for such problems. In all cases, numerical experiments show that convergence rates do not depend on the mesh size of the finest level and discrete optimality systems can be solved with a small multiple of the computational cost which is required to solve the underlying constraint PDE. Employing the full multigrid approach, the computational cost is proportional to the number of unknowns on the finest grid level.

We discuss the role of the regularization parameter in the cost functional and show that the convergence rates are robust with respect to both the fine grid mesh size and the regularization parameter under a mild restriction on the next to coarsest mesh size. Incorporating spectral filtering for the reduced Hessian in the control smoothing step allows us to weaken the mesh size restriction. As a result, problems with near-vanishing regularization parameter can be treated efficiently with a negligible amount of additional computational work. For fine discretizations, robust convergence is obtained with rates which are independent of the regularization parameter, the coarsest mesh size, and the number of levels.

In order to treat linear-quadratic problems with pointwise inequality constraints on the control, the multigrid approach is modified to solve subproblems generated by a primal-dual active set strategy (PDAS). Numerical experiments demonstrate the high efficiency of this approach due to mesh-independent convergence of both the outer PDAS method and the inner multigrid solver.

The PDAS-multigrid method is incorporated in the sequential quadratic programming (SQP) framework. Inexact Newton techniques further enhance the computational efficiency. Globalization is implemented with a line search based on the augmented Lagrangian merit function. Numerical experiments highlight the efficiency of the resulting SQP-multigrid approach. In all cases, locally superlinear convergence of the SQP method is observed. In combination with the mesh-independent convergence rate of the inner solver, a solution method with optimal efficiency is obtained.

# Contents

# 1 Introduction

> In that Empire, the craft of Cartography attained such Perfection that the Map of a Single province covered the space of an entire City, and the Map of the Empire itself an entire Province. In the course of Time, these Extensive maps were found somehow wanting, and so the College of Cartographers evolved a Map of the Empire that was of the same Scale as the Empire and that coincided with it point for point.
>
> *(On Rigor In Science, Jorge Luis Borges)*

Constrained optimization is an ubiquitous task in numerous areas of science, engineering, and economics: network routing problems involve maximizing bandwidth utilization while sustaining a designated throughput, portfolio optimization includes the task of maximizing expected return at a given level of risk exposure, and engineers seek the optimal shape of an aircraft wing or automobile with the objective of minimizing fuel consumption while maintaining speed and stability. In many other applications we find ourselves searching the optimal state of a given system subject to additional constraints.

For many processes in the natural sciences or economics the appropriate mathematical model is given by a partial differential equation (PDE) or systems thereof. A prominent example which appears in an abundance of different applications is given by Poisson's equation on a domain $\Omega \subset \mathbb{R}^d$

$$-\Delta y = u \text{ in } \Omega, \quad y = 0 \text{ on } \partial\Omega. \tag{1.1}$$

A classical problem is to obtain the electrostatic potential $y$ from a given source $u = \rho/\varepsilon_0$ with the charge density $\rho$ and the vacuum permittivity $\varepsilon_0$. A different point of view is as follows: the source function $u$ is at our disposal to *control* the solution $y$, and conforming with optimal control parlance, we call $u$ the control and the corresponding output $y$ of (1.1) the *state*. Using the control $u$ in order to reach some desired objective can formally be written as

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}(y,u) && \text{(1.2a)} \\ \text{subject to} \quad & \mathcal{C}(y,u) = 0. && \text{(1.2b)} \end{aligned}$$

Here, $\mathcal{J}(y,u)$ is the *objective functional* which typically measures quantities like energy or cost which depend on $y$ or $u$. The abstract operator equation (1.2b) embodies the PDE which now acts as a constraint. Problem (1.2) is the abstract formulation of a *PDE constrained optimization* problem, which in turn is a particular instance of

an equality-constrained optimization problem. Note that the PDE-based structure imposes a natural partitioning of the optimization variables $x = (y, u)$.

A concrete example for (1.2) can be found in optimal design of semiconductor devices. Here, one specific task is to determine an optimal doping profile, which enters the source term of (1.1) in a coupled drift-diffusion model [44, 97]. Another example is given by optimization processes arising in the context of tumor treatment by regional hyperthermia [106]. One objective is to noninvasively generate an optimal heat distribution in tissue by microwave or radio frequency energy with the aim of making a tumor more susceptible to accompanying treatments. The control $u$ represents the distributed heat source and controls the temperature distribution in tissue according to the bioheat equation, which is a diffusion equation with an additional nonlinear Helmholtz term. An abundance of problems exist which can be cast in the abstract form (1.2) and we can give only a few pointers to the available literature. We refer to [26] for examples in inverse problems, to [27, 28] for applications from optimal control, to [88, 122] for optimal design and shape optimization and to [79] for optimization problems with a special emphasis on flow control.

If the control $u$ appears in (1.2b) as the source term, we speak of *distributed* control. However, although we focus on distributed control problems in this thesis, other cases are possible and common. In several applications, $u$ enters the boundary conditions of (1.2b). To give an example, a long-time objective in flow control is reducing vorticity in a flow field by injection or suction on parts of the boundary. In the field of optimal shape design, $u$ often represents a certain set of parameters defining (part of) the boundary of the domain on which (1.2b) is defined.

In many applications, additional inequality constraints with respect to (a subset of) the optimization variables have to be satisfied. In this thesis, we will consider pointwise inequality constraints on $u$. To this end, the optimization problem (1.2) is extended by the additional requirement that

$$u \in \mathcal{U}_{\text{ad}}, \tag{1.3}$$

where $\mathcal{U}_{\text{ad}}$ is a given set of *admissible* controls.

In this thesis, we consider distributed control problems of the form (1.2), (1.3), where $\mathcal{J}$ is a quadratic functional and the constraints (1.2b) are given by a linear or semilinear elliptic boundary value problem. The first class of problems is referred to as the linear-quadratic case and has been addressed systematically for the first time in the seminal work of Lions [111]. Existence and uniqueness of solutions to (1.2), (1.3) follow from the well-posedness of the constraints and convexity conditions for $\mathcal{J}$ and $\mathcal{U}_{\text{ad}}$. In the non-convex and singular cases, the theory is far from being mature. Here, semilinear elliptic constraints are the most accessible class of problems and cover many important applications such as a simplified scalar Ginzburg-Landau model in superconductivity [80]. Important theoretical contributions are due to Lions [112], Casas and Tröltzsch [6] and Ito and Kunisch [100].

In almost all practical cases, neither for the constraints (1.2b) nor for the complete problem (1.2), (1.3) is a closed-form solution available. Therefore, one has to resort to numerical simulation. To this end, the infinite-dimensional problem (1.2), (1.3) has to be discretized, which is usually done with finite element or finite difference methods [6, 64, 94, 121]. If $h$ denotes a discretization parameter related to the mesh width, discretizing (1.2b) yields a linear or nonlinear system of equations with $n_y = \mathcal{O}(h^{-d})$ unknowns in $d$ dimensions. When treating the optimization problem (1.2), the controls $u$ are treated as unknowns as well, and naturally the number $n_u$ of discrete control unknowns is of particular interest. In the case of optimal shape design often a priori $u$ is a discrete quantity with $n_u = \mathcal{O}(1)$ and a small constant $< 100$. On the other hand, if $u_h$ arises from the discretization process for (1.2b) as well, we obtain $n_u = \mathcal{O}(h^{-(d-1)})$ for boundary control problems and $n_u = \mathcal{O}(h^{-d})$ for distributed control problems. Thus, for fine discretizations not only $n_y$ but also $n_u$ will be large. In any case, a large system of linear or nonlinear equations arises from the discretization, and solving these systems in optimal complexity $\mathcal{O}(n)$, where $n$ is the total number of unknowns, is a nontrivial task.

The objective of this work consists in devising a fast and efficient algorithm for the solution of discretized PDE constrained optimization problems. In particular, we aim at computing a discrete solution with an accuracy proportional to the discretization error and a cost which is at most $\mathcal{O}(n \log n)$. For problems arising from discretized second-order elliptic PDEs a solver with this (near-optimal) efficiency is the multigrid method, and in combination with nested iteration the optimal complexity $\mathcal{O}(n)$ is obtained. As it will turn out, the multigrid method can be adapted to solve problems of the type (1.2) with the same asymptotic complexity. One possible way to picture the multigrid method is as follows: solutions obtained on coarser grids are utilized to remove smooth components from the error, whereas a *smoothing iteration* on each level reduces the oscillatory error components on that level. A major task we are concerned with is to develop a smoothing iteration which is suitable to be applied in the context of PDE constrained optimization. For more details on the multigrid method we refer to [40, 41, 85, 154] as well as [147] and the numerous references therein.

Incorporating multigrid ideas in optimization algorithms can be carried out in a number of different ways. In order to reach our desired objective, we have to consider the following question: how does the computational effort needed for the solution of (1.2) scale in relation to the cost for the solution of (1.2b)? In other words, if we desire to reduce the discretization error and, say, halve the mesh size $h$, which impact will this have on the computational cost for the solution of the overall optimization problem? Subsequently we will discuss this and other important issues and outline our own approach in more detail. To this end, we proceed by giving some necessary details on the inner workings of optimization techniques appropriate for the solution of (1.2), (1.3).

A key issue for any fast optimization algorithm is to exploit gradient and curvature

information. Due to the PDE-based nature of the optimization problems considered, under suitable assumptions, (Fréchet–) derivatives up to second-order exist. Newton's method proves to be particularly attractive here due to the following properties: first, the speed of convergence is locally superlinear or quadratic, and second, Newton's method for discretized operator equations often exhibits a mesh-independence principle [3]. In combination with an optimal solver for the Newton systems an overall mesh-independent solution method results. However, we do not hesitate to mention a well-known drawback of Newton's method: convergence can be guaranteed only in a neighborhood of a local minimizer. Thus, an initial iterate which is sufficiently close to a solution is required, and in practice globalization by line search, filter or trust-region techniques as well as hybridization with a globally convergent gradient-scheme should be considered. We stress the point that the objective of these globalization measures is to assure convergence from remote starting points only, and not to find a globally optimal solution.

Let us briefly digress here to comment on the relation to the vast field of *global optimization*. Here, common strategies are based on a reduction of the search space by heuristics or random sampling, thus giving a leverage to trade in the accuracy of the approximately globally optimal solution for a reduced computational effort. An example for a probabilistic search method is simulated annealing. In contrast, branch-and-bound algorithms are an example for deterministic meta-strategies. They follow the divide-and-conquer principle, building up a tree of the search space with branching steps and removing subtrees using a bounding criterion. In *continuous optimization* however, which is the focus of this thesis, local gradient-descent and Newton-type methods are the principal tools. However, these methods may be of use for the solution of subproblems which arise in global optimization. In [110], a mixed integer nonlinear programming problem has been solved with a branch-and-bound search which uses a Newton-based algorithm to solve nonlinear programming problems arising at the nodes.

Let us now look at Newton-like methods in optimization in some more detail. We ignore the side constraints (1.3) for the moment, as they will be considered explicitly at a later stage. A first approach, related to the generalized reduced gradient method [69], is to eliminate $y = y(u)$ (tacitly assuming local invertibility of $\mathcal{C}(y, u)$) and apply Newton's method to minimize the reduced functional

$$\hat{\mathcal{J}}(u) = \mathcal{J}(y(u), u). \tag{1.4}$$

Due to the elimination of the constraints, every iterate naturally satisfies the constraints. Methods with this property are called *feasible*. Furthermore, the dimension of the problem is reduced to that of the control space. Therefore, this approach is also termed a *reduced* Newton method. Such methods might be attractive if the dimension of the control space is small or if it is of high priority to reuse an existing implementation for the elimination of the constraints. Some optimization codes build around this concept, using PDE solvers as a black-box within an outer optimization loop [28].

Several drawbacks are associated with the reduced Newton approach. In each Newton iteration, a full nonlinear solve of the constraint PDE is required for the gradient computation. Furthermore, the solution of the Newton system requires inverting the Hessian $D^2_{uu}\hat{\mathcal{J}}(u)$. In PDE constrained optimization, the operator $D^2_{uu}\hat{\mathcal{J}}(u)$ involves the solution operators of the linearized constraints and would be represented by a dense matrix. For large-scale problems it is therefore not possible to explicitly assemble $D^2_{uu}\hat{\mathcal{J}}(u)$ and solve the related system with direct factorization. Instead, iterative methods such as conjugate gradient are applied. Computing the matrix-vector product with $D^2_{uu}\hat{\mathcal{J}}(u)$ requires computing two inversions of the linearized constraints. Unacceptably large computational costs are the consequence, since the number of the conjugate gradient iterations in general depends on $N_u$ and furthermore, $D^2_{uu}\hat{\mathcal{J}}(u)$ can be quite ill-conditioned. Devising preconditioners for the non-assembled reduced Hessian is a largely unanswered question (but see [125] for an exception).

A first attempt to overcome this difficulty with the help of multigrid is due to Hackbusch [83]. He observed that, in the context of a linear-quadratic problem constrained by (1.1), the reduced Hessian $D^2_{uu}\hat{\mathcal{J}}(u)$ is an operator of Fredholm type and a suitable multigrid method [84] could be applied, where the smoothing iteration is given by a fixed point iteration. Although the scaling behavior with respect to $n_u$ is improved, the computational cost is still quite large since the smoothing step requires the evaluation of $D^2_{uu}\hat{\mathcal{J}}(u)$.

A further disadvantage of the reduced Newton method, which has been reported in [27] in the context of flow control (see also [96]), is an adverse effect on the outer Newton iteration. Due to the nonlinear elimination in fact more Newton steps might be required, compared to other second-order methods.

Quasi-Newton methods eliminate the need to compute $D^2_{uu}\hat{\mathcal{J}}(u^l)$ by maintaining an approximation $W^l$ based on low-rank updates in each Newton iteration $l$. The convergence speed reduces to superlinear at most, but a substantial amount of computational cost can be saved. However, in large-scale applications the $W^l$ tend to become dense, unless one resorts to limited-memory variants, which come at the expense of a further reduction in convergence speed. A delicate issue for ill-conditioned problems is how to choose the initial approximation $W^0$. On the other hand, a distinct advantage of most quasi-Newton methods is that by construction the matrices $W^l$ are guaranteed to be positive definite. Quasi-Newton methods have been the methods of choice if second-order information is not available or expensive to compute, however in these cases automatic differentiation often provides a viable alternative. Further details and references can be found for instance in [127]. Here instead, we focus on methods which employ "true" second-order information.

A substantial gain in performance may be obtained when the feasibility requirement is dropped. For many purposes, it is sufficient if the *final* iterate of an optimization algorithm is feasible[1] within the desired accuracy. This idea leads to *infeasible* algo-

[1]There are exceptions, e.g. in singular control problems, a constraint violation might prevent ob-

rithms such as Lagrange-Newton and SQP methods [4], which are derived by applying Newton's method to the necessary first-order optimality condition for the Lagrangian

$$\mathcal{L}(y,u,p) = \mathcal{J}(y,u) + p^T \mathcal{C}(y,u), \tag{1.5}$$

with Lagrange multipliers $p$. This yields a sequence of linear systems

$$\begin{pmatrix} D^2_{xx}\mathcal{L}(x^l,p^l) & (D_x\mathcal{C}(x^l))^T \\ D_x\mathcal{C}(x^l) & 0 \end{pmatrix} \begin{pmatrix} \delta x^l \\ \delta p^l \end{pmatrix} = - \begin{pmatrix} D_x\mathcal{J}(x^l) + (D_x\mathcal{C}(x^l))^T p^l \\ \mathcal{C}(x^l) \end{pmatrix}. \tag{1.6}$$

Now the key in deriving SQP is noting that, provided $D^2_{xx}\mathcal{L}(x^l,p^l)$ is positive definite on $\ker D_x\mathcal{C}(x^l)$, the Newton increments $\delta x^l, \delta p^l$ defined by (1.6) solve the first-order optimality system of the linear-quadratic optimization problem

$$\text{minimize} \quad \frac{1}{2}\delta x^T D^2_{xx}\mathcal{L}(x^l,p^l)\delta x + (D_x\mathcal{L}(x^l,p^l))^T \delta x \tag{1.7a}$$

$$\text{subject to} \quad D_x\mathcal{C}(x^l)\delta x = -\mathcal{C}(x^l). \tag{1.7b}$$

Thus, a solution of the nonlinearly constrained problem (1.2) can be obtained by solving a sequence of linear-quadratic problems (1.7). Under suitable assumptions, the fast local convergence and the mesh-independence of Newton's method are inherited by the SQP method. Furthermore, the inequality constraints (1.3), which are difficult to handle with Newton's method due to a lack of differentiability[2] can be enforced by explicitly adding them to (1.7).

The basic algorithm outlined so far corresponds to a local SQP method. Devising a robust and efficient SQP method involves several additional issues such as globalization, the choice of a suitable merit function to control the progress of the iteration[3], and inexactness issues derived from inexact Newton methods [55]. Further variants result if the Hessian $D^2_{xx}\mathcal{L}(x^l,p^l)$ in (1.7) is approximated by quasi-Newton methods or is chosen as the Hessian of an augmented Lagrangian. We refer to [32] for a survey on SQP methods and to [127] for further details on practical issues and related techniques such as augmented Lagrangian and quasi-Newton-SQP methods.

The outlined SQP algorithm gives rise to a number of variants, depending on the solution method for (1.6). The system (1.6) has saddle point structure and as such is an indefinite system which has $n_x = n_y + n_u$ positive and $n_p$ negative eigenvalues. Saddle point systems and their efficient solution present a large and active area of research, and since the advent of the prominent Uzawa iteration, numerous algorithms have been proposed [22]. Examples for applications leading to saddle point systems

taining a solution of intermediate subproblems.

[2] It is possible to consider inequality constraints using the concept of semismooth Newton methods [91, 95], however the resulting algorithm in finite dimension will again be of SQP-type.

[3] In constrained optimization, a merit function is needed to balance the objectives of minimizing $\mathcal{J}$ and satisfying the constraints.

are discretizations using mixed finite element methods and fluid flow equations, e.g. the Stokes- and Navier-Stokes-equations. A substantial part of research is directed towards these problems. Despite the common structure in an abstract setting, the characteristics of these saddle point systems are in general quite different. For instance, in steady state Stokes flow, applying a certain block elimination to the saddle point matrix yields a Schur complement which is spectrally equivalent to an $L^2$ identity. In contrast, the Schur complement arising from an analogous elimination strategy applied in the context of optimal control is precisely $D^2_{uu}\hat{\mathcal{J}}(u^l)$. This shows that there is little hope to obtain a generic solver for saddle point systems in a "black-box" sense (with the exception of direct factorizations).

Three approaches are predominant in the context of optimization problems: direct factorization, iterative solution and reduced SQP methods. Arguably the most prevalent SQP variant in PDE constrained optimization (but not limited to such problems) are reduced SQP-methods, which are obtained by employing a reduction to the control space, much like in the reduced Newton method, only here applied for the linear-quadratic control problem at each SQP step. Essentially, the reduced SQP methods differ from the reduced Newton method in that they employ linearized PDE operators only, leading to a reduction in computational cost, since the nonlinear constraint is solved only once during the course of the optimization process. Reduced SQP methods have been applied in PDE constrained optimization for example in [29, 71, 96, 108], and many more examples can be found in the references given at the beginning of this introduction. Until recent, most multigrid-related efforts in PDE constrained optimization also fell into this category. In fact, most published methods are closely related to Hackbusch's approach, following that strategy in each SQP iteration. We mention [71] for a parabolic boundary control problem and [58] for an optimal shape design problem. A reduced quasi-Newton method using BFGS updates and incorporating a two-grid method has been applied in [155] to a parameter identification problem in groundwater flow.

Due to the large similarities of the reduced Newton and SQP methods, most of the sketched drawbacks apply to the latter as well. Therefore, recently there has been an increasing interest in all-at-once or coupled approaches for the solution of (1.6). For a better distinction, such methods are sometimes called *full SQP*, in contrast to the reduced variants. Sparse factorization techniques for indefinite matrices have reached a high degree of sophistication [59] and enjoy a considerable popularity in the optimization community, largely due their robustness and their black-box character. However, they are not well suited in the situation of PDE constrained optimization due to their high demand on computational resources.

Iterative methods such as Krylov subspace iterations can be applied rather straightforwardly, since only matrix-vector products with the saddle point matrix are needed. However, convergence of Krylov methods for indefinite problems is slow, making effective preconditioning a prerequisite. Full SQP variants employing inexact block factorizations as preconditioner for Krylov methods have been developed in an ab-

stract optimization context [48, 68], for parameter estimation [82] and for optimal control problems [20, 66, 67, 140].

Our approach is best described as a full SQP method with a coupled multigrid method for the solution of (1.6). Before we point out the key features and contributions of our work, let us briefly summarize existing multigrid approaches which make use of coupled approaches in some sense.

In [9], which is the earliest reference for a fully coupled multigrid method in the optimization context, the term "One-Shot" approach has been coined. There, a simultaneous multigrid solution of a boundary control problem with constraints given by (1.1) has been proposed, employing a variant of a gradient descent scheme as the smoothing iteration. In [58], an inexact null-space iteration has been employed as a smoothing iteration within a full space multigrid method. Formally, the method is presented as an SQP approach, however it is applied to a linear programming problem from topology optimization, where only the Hessian changes due to the log-barrier terms of an interior point method. The PDE constraints are linear and do not change.

Several publications due to Borzi [33–35] address the multigrid solution of linear and semilinear elliptic control problems. The method uses the full approximation storage (FAS) variant of multigrid, which is applied directly to the nonlinear system of the first-order optimality conditions. Hence, the linearization, as well as an optional projection step in the presence of inequality constraints, are performed locally in the smoothing iteration. As smoother a collective Gauss-Seidel iteration is chosen. The same approach has been applied to a parameter identification problem for the diffusion equation in [14].

A more general framework which is closer related to the "One-Shot" approach is proposed in [124]. Here, the multigrid method plays the role of an outer iteration, similar to the FAS approach, while on each grid level the smoothing iteration may be implemented by any optimization method. This methodology offers flexibility accompanied by a satisfying convergence theory, but until now has not been on par with the more classical multigrid approach, performance-wise [33].

Our own contributions in this context are as follows:

- In a preliminary step, we devise a preconditioned GMRES iteration for the solution of linear-quadratic equality-constrained problems. Our preconditioner is a block triangular variant of a constraint preconditioner [102]. Numerical experiments are given and the robustness of the approach is discussed.

- In a second step, a coupled multigrid method for linear-quadratic problems is developed. The smoothing effect of the smoothing iteration, which incorporates elements from constraint preconditioning, is described in detail. This approach is related to [58] but differs in the approximation of the inexact reduced Hessian. For a particular variant, a quantitative local Fourier analysis is given which shows that for a wide range of parameters, the smoothing rate of the coupled approach is close to the smoothing rate governed by the underlying PDE constraint.

These findings are affirmed in detailed numerical experiments. The implementation is extended to the full multigrid method and numerical results are given which demonstrate that indeed the PDE constrained optimization problem can be solved with optimal complexity $\mathcal{O}(n)$. So far most results obtained with coupled multigrid approaches found in the literature are restricted to the model problem. Here, we give detailed additional numerical experiments for more complex PDE constraints. In particular, we consider constraints which can not be treated with the standard pointwise smoothers employed in the scalar case. We consider the full diffusion equation, the anisotropic diffusion equation on non-uniform meshes, and the convection-diffusion equation with circular wind. The results indicate that our approach can be adapted to constraints which require non-standard smoothing techniques. We believe that this is a benefit not found in other multigrid approaches so far.

- We provide a detailed discussion of the robustness properties with respect to the regularization parameter in the standard $L^2$-regularized functional. Based on concepts related to regularized Fredholm operators of the first kind, a modification leading to increased robustness in the presence of near-vanishing regularization parameter is proposed.

- We extend our approach for the linear-quadratic case in order to handle the inequality constraints (1.3). To this end, we apply the primal-dual active set (PDAS) strategy [23, 24, 91]. The PDAS generates a sequence of equality constrained problems which can be solved efficiently with a modified variant of our multigrid method. This approach is novel in the sense that we solve the full primal-dual system generated by the PDAS iteration with the multigrid method. Previously, the inner systems have been solved with reduced space methods only. We remark that combinations of active set strategies and multigrid methods exist in the context of variational inequalities resulting from obstacle problems [42, 87, 98].

- Finally we employ the PDAS-multigrid method as solver for the linear-quadratic, inequality constrained systems generated by the SQP iteration. We improve the efficiency by allowing for inexactness of the SQP substeps in the sense of inexact Newton methods [55]. This avoids "oversolving" in the case where Newton directions yield too little progress. At the same time, the superlinear convergence of the SQP iteration is preserved. The efficiency of the method is demonstrated on several numerical examples. To our knowledge, no full SQP-multigrid method has been applied to nonlinear problems before. The only other coupled multigrid approach known to us which treats semilinear constrained problems is the FAS method [35].

**Thesis Outline** The remainder of this thesis is organized as follows. In *Chapter 2* we present the theoretical foundation for the numerical solution of linear-quadratic PDE-constrained optimization problems. The discretization of the first-order conditions then yields a large indefinite and symmetric system of equations and we conclude the chapter by stating some properties important for the iterative solution of such saddle point systems. In *Chapter 3* we present an iterative one-level method based on the concept of constraint preconditioning. Detailed numerical results serve to demonstrate the potential as well as the shortcomings of the single-level approach. In *Chapter 4* we devise a multigrid method for the fast and efficient solution of discretized linear-quadratic optimization problems. The smoothing iteration smoothing, which incorporates elements from constraint preconditioning is described in detail. An extensive discussion of detailed numerical experiments accounts for the efficiency and flexibility of the proposed method. In *Chapter 5* the multigrid method will be adapted to handle the equality-constrained problems generated by a PDAS strategy. Several numerical experiments show that the combined PDAS-multigrid strategy yields an efficient solver for control-constrained optimal control problems. In *Chapter 6* we bring together the previous results in a full SQP approach including inequality constraints. Numerical experiments for several semilinearly constrained optimization problems are given to demonstrate the efficiency of the SQP-multigrid approach. In *Chapter 7* we will summarize our findings and provide an outlook on further development and promising extensions of our proposed method.

**Acknowledgments** At this point I would like to express my gratitude to all the people who have contributed to the completion of this thesis. First of all I would like to thank my adviser Prof. Dr. Michael Griebel for his commitment as well as many helpful and inspiring discussions. The excellent working conditions at the Institute for Numerical Simulation provided the sound foundation on which this piece of research could be conducted. Also I would like to thank the committee member Prof. Dr. Sören Bartels for providing the second opinion on this dissertation.

A sincere thank you is directed to all colleagues at the INS for the nice work atmosphere. In particular, a big thanks goes to Roberto Croce, without whom everything would have been not even half the fun. Furthermore, I am very grateful to Christian Feuersänger, who patiently provided support for his excellent LaTeX package PGFPlots. Last but not least Dr. Marc Alexander Schweitzer deserves a special mention since he always willingly provides valuable insight in issues ranging from petty technical details to challenging mathematical questions. A special note of gratitude goes to Susanne Lenz, Melanie Reiferscheid and Ralf Wildenhues for the tedious and unrewarding task of proof-reading.

On a personal level, I will always be thankful for the encouragement and unconditional understanding I experienced from Melanie, my family and my friends.

A sizable part of this thesis originates from a research stay at the School of Com-

putational Science at Florida State University in fall 2006. I am very much obliged to Prof. Max Gunzburger for the kind hospitality and generosity I was allowed to experience during my visit.

This work would not have been possible without the financial support of the Deutsche Forschungsgemeinschaft, which was granted through project C2 of the collaborative research center 611 "Singular Phenomena and Scaling in Mathematical Models", this is also gratefully acknowledged.

Bonn, February 2009 Martin Engel

# 2 Optimal Control of Elliptic Boundary Value Problems

This chapter covers the theoretical background as well as the finite-dimensional approximation for the linear-quadratic case of PDE constrained optimization problems given in the abstract formulation

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}(z(y), u) \\ \text{subject to} \quad & \mathcal{C}(y, u) = 0. \end{aligned} \tag{OP}$$

The functional $\mathcal{J} : Z \times U \to \mathbb{R}$, which we seek to minimize, is the *objective functional*, which in general involves suitable norms of $u$ and $y$. The operator $\mathcal{C} : Y \times U \to W$ specifying the constraints represents a partial differential equation (PDE). We call the solution $y$ of the PDE the *state variable*. The *control variable* $u$ is at our disposal to influence the system $\mathcal{C}$. The variable $z(y)$ is denoted the *observation* of the state and is given by $z(y) = Ey$ with a linear and continuous observation operator $E : Y \to Z$. A concrete choice of appropriate spaces will be given below. Most choices are in fact dictated by the cost functional and the variational formulation of the constraints. The notion "linear-quadratic" refers to the class of problems where $\mathcal{J}$ is a quadratic functional and the constraints $\mathcal{C}$ are linear. The case of nonlinear $\mathcal{C}$ will be discussed in the first section of Chapter 6.

In many applications additional constraints on the control variable are modeled by requiring that $u \in \mathcal{U}_{\text{ad}}$ holds, where $\mathcal{U}_{\text{ad}} \subset U$, the set of *admissible controls*, is a proper closed and convex subset of the control space $U$. We stress the fact that, even in the case of linear $\mathcal{C}$, if $\mathcal{U}_{\text{ad}}$ is a proper subset of $U$, the optimization problem itself is nonlinear, which requires special numerical techniques. Therefore, the numerical treatment of the case $\mathcal{U}_{\text{ad}} \subset U$ is deferred to Chapter 5. The theoretical setting in the case of linear constraints however includes the case of $\mathcal{U}_{\text{ad}} \subset U$. Imposing analogous constraints on the state $y$ leads to Lagrange multipliers with low regularity (measures, in general). We do not discuss this case here and instead refer to e.g. [49].

After substantiating the setting and stating the relevant results on existence and uniqueness of solutions to (OP) we continue with deriving the necessary first-order conditions. In the convex case, these are also sufficient and thus completely characterize the unique solution. The first-order conditions constitute a system of coupled PDEs termed the optimality system and provide one possible way to derive a finite-dimensional approximation of the continuous optimal control problem. The linear system resulting after discretization is of saddle point type, the matrix representing

the linear operator is commonly denoted the KKT (Karush-Kuhn-Tucker) matrix. The chapter concludes with some important properties of these KKT matrices and their impact on the numerical solution. Efficient numerical methods for the linear-quadratic case are the topic of Chapters 3 and 4.

Let us now give a concrete instance of (OP), which frequently serves as a model problem for theoretical and numerical investigations.

**Example 2.1** (Optimal Control of the Poisson equation (The Linear-Quadratic Model Problem (LQMP)))**.** Here and throughout we assume that $\Omega \subset \mathcal{R}^d, d = 2, 3$ is a convex polygonal domain or a domain with boundary of class $C^{1,1}$. Let $\bar{y} \in L^2(\Omega)$ be a given function, the so-called *target state* and define the *tracking type* functional

$$\mathcal{J}(y,u) = \frac{1}{2}\|Ey - \bar{y}\|_Z^2 + \frac{\sigma}{2}\|u\|_U^2, \tag{2.1}$$

with $\sigma > 0$ a regularization parameter. The constraints $\mathcal{C}$ are given by the Poisson equation with homogeneous Dirichlet boundary conditions,

$$\begin{aligned} -\Delta y &= f + u \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \Gamma \end{aligned} \tag{2.2}$$

with a given source term $f \in L^2(\Omega)$. In this case we have $\mathcal{U}_{\text{ad}} = U = L^2(\Omega)$, $Z = L^2(\Omega)$, $Y = H_0^1(\Omega)$ and $W = L^2(\Omega)$ (or $H^{-1}(\Omega)$). The observation operator $E$ is the continuous injection $H_0^1(\Omega) \hookrightarrow L^2(\Omega)$. The aim of minimizing (2.1) subject to (2.2) is to find a control function $u$ such that the solution $y$ of Poissons's problem minimizes the $L^2$-distance to the target state $\bar{y}$. Functionals similar to (2.1) are frequently encountered in parameter identification and inverse problems. In this context, (2.1) corresponds to a Tikhonov-regularized output least-squares functional.

*Remark* 2.2. The LQMP Example 2.1 is an example of a *distributed* control problem, due to the appearance of $u$ on the right hand-side of (2.2). Another type of control arising in practical applications is given by *boundary control*, that is, the control $u$ enters the constraints through the boundary conditions. Also various other generalizations to Example 2.1 are possible. For instance, the output state $y$ can be observed in a subdomain of $\Omega$ only (*partial observation*), or on the boundary (*boundary observation*).

A comprehensive exposition of the theory for the case of linear constraints was given for the first time in the seminal work of Lions [111]. In our presentation we will adopt this approach and focus on constraints which are given by well-posed boundary value problems (BVP). Different approaches, which are suitable for the control of singular systems are discussed in e.g. [65, 112]. The results given in Section 2.1 can be found in the monographs [111], [65] and [145].

## 2.1 Existence of Solutions

In order to substantiate the framework for the abstract minimization problem (OP) we need to introduce some notation. For further details on the functional analytic setting we refer to e.g. [2, 157] and Appendix A.

Let $Y$ be a real Hilbert space with its dual $Y'$ and let $a(\cdot,\cdot)$ be a continuous and coercive bilinear form on $Y$. We denote by $A : Y \to Y'$ the linear and continuous operator which is defined by

$$\langle Ay, v\rangle_{Y',Y} = a(y,v) \quad \text{for } y, v \in Y. \tag{2.3}$$

Due to the Lax-Milgram theorem, $A$ is bijective and has a continuous inverse $A^{-1} : Y' \to Y$. We introduce a linear and continuous operator $B : U \to Y'$, which describes the action of the control (in Example 2.1, $B$ is the continuous injection $L^2(\Omega) \hookrightarrow H^{-1}(\Omega)$). For a given source $f \in Y'$, the linear constraints are then given by

$$Ay = f + Bu \quad \text{in } Y', \tag{2.4}$$

and for each control $u$ we denote the corresponding state by $y(u)$. We introduce the *reduced* functional

$$\hat{\mathcal{J}}(u) = \mathcal{J}(Ey(u), u). \tag{2.5}$$

*Remark* 2.3. Several numerical solution procedures are based on the reduced functional $\hat{\mathcal{J}}$, these approaches are grouped into the category of the *reduced space* and *feasible* methods. In contrast to this, there is a family of algorithms called *full space* methods. We will return to this issue in Chapters 3 and 4 where we discuss the numerical solution in detail.

The abstract optimization problem (OP) can now be equivalently stated as

$$\min_{u \in \mathcal{U}_{\text{ad}}} \hat{\mathcal{J}}(u), \tag{2.6}$$

where the solution of (2.6) is the optimal control, denoted with $u^*$. Existence of a solution is stated in (cf. [111, Ch. II], or [65])

**Theorem 2.4.** *Let $\mathcal{U}_{ad}$ be closed and convex and $\hat{\mathcal{J}}(u) : \mathcal{U}_{ad} \to \mathbb{R}$ be weakly lower semicontinuous over $\mathcal{U}_{ad}$. Then an optimal solution of* (2.6) *exists, if either $\mathcal{U}_{ad}$ is bounded or $\hat{\mathcal{J}}$ is radially unbounded, i.e. $\hat{\mathcal{J}} \to \infty$ for $\|u\| \to \infty$.*

*Remark* 2.5. The lower semicontinuity is obtained in particular if $\hat{\mathcal{J}}$ is Gâteaux-differentiable and convex. From now on we assume these properties to hold. Furthermore, if $\hat{\mathcal{J}}$ is strictly convex, the optimal control is unique.

A useful property is established by

**Proposition 2.6.** *Let $\hat{\mathcal{J}}$ be twice Gateaux-differentiable on a convex subset $X_1$ of a linear normed space $X$. If for any $x \in X_1$ we have that $\hat{\mathcal{J}}''(x)(v,v) \geq 0$ for all $v \in X$, then $\hat{\mathcal{J}}$ is a convex functional. If furthermore for a constant $\alpha > 0$ independent of $v$ the condition $\hat{\mathcal{J}}''(x)(v,v) \geq \alpha\|v\|$ holds for all $v \in X$, then $\hat{\mathcal{J}}$ is strictly convex.*

From this we immediately conclude that if $X$ is a Hilbert space with norm $\|.\|$ and $\bar{x} \in X$, then the functional $\mathcal{J}(x) = \|x - \bar{x}\|^2$ is strictly convex. All assumptions are satisfied by the tracking type functional (2.1).

*Remark* 2.7 (The Case of $\sigma = 0$ in (2.1)). Under certain conditions, the existence of solutions can be established also in the case of a vanishing regularization term in the objective functional (2.1). For unbounded $\mathcal{U}_{\text{ad}}$, the coercivity of $\mathcal{J}$ can be restored by measuring the tracking error in a stronger norm, e.g. the $H^1$-norm would be suitable in the case of Example 2.1, cf. [65, 111]. Evidently this also requires a higher regularity of the target state $\bar{y}$. Note however, that only for $\sigma > 0$ higher regularity than $u^* \in L^2(\Omega)$ can be expected. For $\sigma = 0$ one usually obtains a so-called *bang-bang control*, that is, $u^*$ attains almost everywhere values which are on the boundary of $\mathcal{U}_{\text{ad}}$, cf. [111, Ch.II, Remark 2.3].

## 2.2 Characterization of Solutions and Optimality Systems

We will now derive a convenient characterization of solutions in terms of the gradient of $\hat{\mathcal{J}}$, following the classical text [111]. In the abstract setting we obtain variational inequalities which transform to the necessary first-order conditions constituting the *optimality system*. In the case of a strictly convex functional and linear constraints, e.g. the setting of Example 2.1, the optimality system also provides the sufficient condition for the solution of the optimization problem. For $\mathcal{U}_{\text{ad}} = U$, the variational inequality reduces to an equality. In the general case, Lagrange multiplier or Karush-Kuhn-Tucker theory transforms the variational inequalities into a system of equations with additional complementarity conditions that is at the heart of many numerical optimization methods for the solution of (OP), including active-set strategies, which are discussed in Chapter 5.

The central result is

**Theorem 2.8.** *Let $f : \mathcal{U}_{ad} \to \mathbb{R}$ be a Gâteaux-differentiable functional on the closed and convex set $\mathcal{U}_{ad}$. Then if $u^*$ is a solution of*

$$f(u^*) = \inf_{u \in \mathcal{U}_{ad}} f(u) \tag{2.7}$$

*the* variational inequality

$$f'(u^*)(v - u^*) \geq 0 \quad \textit{for} \quad v \in \mathcal{U}_{ad} \tag{2.8}$$

*holds. If $f$ is convex and $u^*$ satisfies* (2.8) *then $u^*$ is a solution of* (2.7).

*Remark* 2.9. If $\mathcal{U}_{\text{ad}} = U$, we can set $v = u^* \pm \varphi$ for $\varphi \in U$ and (2.8) reduces to

$$f'(u^*) = 0. \tag{2.9}$$

Furthermore, if $f$ is strictly convex, (2.8) or (2.9) are necessary and sufficient conditions for $u^*$ to be a solution of (2.7).

We will now apply Theorem 2.8 to the reduced version $\hat{\mathcal{J}}$ of the tracking functional (2.1) in order to deduce concrete optimality systems. From (2.4) we obtain

$$Ey(u) - \bar{y} = EA^{-1}f + EA^{-1}Bu - \bar{y} \tag{2.10}$$

and the Gâteaux-derivative with respect to $u$ is then given by

$$D_G(Ey(u) - \bar{y})v = EA^{-1}Bv \quad \text{for } v \in U. \tag{2.11}$$

Using (2.11) and (2.1), we obtain

$$D_G\hat{\mathcal{J}}(u)v = (Ey(u) - \bar{y}, EA^{-1}Bv)_Z + \sigma(u,v)_U, \quad v \in \mathcal{U}_{\text{ad}}. \tag{2.12}$$

Implicitly using the Riesz map $\Lambda : Z \to Z'$ yields

$$\begin{aligned}(Ey(u) - \bar{y}, EA^{-1}Bv)_Z &= \langle Ey(u) - \bar{y}, EA^{-1}Bv\rangle_{Z',Z} \\ &= \langle E'(Ey(u) - \bar{y}), A^{-1}Bv\rangle_{Y',Y},\end{aligned} \tag{2.13}$$

where $E' : Z' \to Y'$ is the dual operator of $E$. We define the dual operator $A' : Y \to Y'$ using the relation

$$(A'v, y) = (Ay, v) = a(y, v), \quad y, v \in Y, \tag{2.14}$$

and identify $Y$ with its bidual space $(Y')'$ in the canonical way. Using that $A'$ is an isomorphism[1], for $u \in U$ we introduce the *adjoint variable* $p = p(u) \in Y$ as the unique solution of

$$A'p = E'(\bar{y} - Ey(u)). \tag{2.15}$$

Substituting (2.15) into (2.13) and using relation (2.14) for the dual operator, we obtain

$$\begin{aligned}D_G\hat{\mathcal{J}}(u)v &= \sigma(u,v)_U - (A'p, A^{-1}Bv)_Y \\ &= \sigma(u,v)_U - (p, Bv)_Y \\ &= \sigma(u,v)_U - (B'p, v)_U,\end{aligned} \tag{2.16}$$

where $B' : Y \to U'$ is the dual of the control action operator $B$. We would like to point out that the previous computation shows in particular that the gradient of $\hat{\mathcal{J}}$ can be identified with $\sigma u - B'p \in U'$. Evaluating the *reduced gradient* defined by (2.16) requires evaluation of the solution operators of the adjoint equation (2.15) and the

[1] $A'$ is defined by the adjoint of the bilinear form $a(y, v)$.

state equation (2.4), i.e. the solution of two PDEs. Computing (2.16) is a crucial step of reduced space methods and accounts for the larger part of the computational work.

Let us now assume that $\mathcal{U}_{\text{ad}} = U$. Collecting the previous computations and using (2.8), we obtain the *optimality system*

$$\begin{aligned} Ay(u^*) &= f + Bu^*, \\ A'p(u^*) &= E'(\bar{y} - Ey(u^*)), \\ \sigma u^* - B'p(u^*) &= 0. \end{aligned} \tag{OS}$$

The first equation in (OS) is the *state equation*, the second equation is called the *adjoint equation* and the last equation is the *optimality condition*. In the case that $\mathcal{U}_{\text{ad}} \subset U$ is a proper subset, the last equality has to be replaced by

$$(\sigma u^* - B'p(u^*), v - u^*) \geq 0, \quad v \in \mathcal{U}_{\text{ad}}. \tag{2.17}$$

**Example 2.10** (Optimality System for Distributed Control of the Poisson Equation)**.** The concrete instance of (OS) for Example 2.1 is given by

$$\begin{aligned} -\Delta y &= f + u \quad \text{in } \Omega \\ y &= 0 \quad \text{on } \Gamma, \end{aligned} \tag{2.18a}$$

$$\begin{aligned} -\Delta p &= \bar{y} - y \quad \text{in } \Omega \\ p &= 0 \quad \text{on } \Gamma, \end{aligned} \tag{2.18b}$$

$$\sigma u - p = 0. \tag{2.18c}$$

Again, if $\mathcal{U}_{\text{ad}} \subset U$, instead of (2.18c) we obtain

$$(\sigma u - p, v - u) \geq 0, \quad v \in \mathcal{U}_{\text{ad}}. \tag{2.19}$$

For the remainder of this chapter and throughout Chapters 3 and 4 we assume that $\mathcal{U}_{\text{ad}} = U$. The inequality constrained case will be treated in Chapter 5.

*Remark* 2.11 (Regularity of the Optimal Control). For the case of $\mathcal{U}_{\text{ad}} = U$ and distributed control with homogeneous Dirichlet boundary conditions, $u^* \in H^2(\Omega)$ follows.

In view of the numerical examples, we remark that analogous results are obtained for

**Example 2.12** (General Second-Order Elliptic Operator)**.** In Example 2.1, the Laplace operator can be replaced by the general second-order elliptic operator

$$Ly = -\sum_{i,j=1}^{d} D_i(a_{ij} D_j y) + cy \tag{2.20}$$

where $c, a_{ij} \in C^{1,1}$, $c \geq 0, a_{ij} = a_{ji}$, and for the coefficients the uniform strong ellipticity

$$\sum_{i,j=1}^{d} a_{ij} x_i x_j \geq \alpha |x|^2 \tag{2.21}$$

holds for all $x \in \mathcal{R}^d$ with a constant $\alpha > 0$ independent of $x$.

Further results including Neumann boundary control and boundary observation as well as constraints given by systems of equations can be found in the given references. Inhomogeneous Dirichlet control on the boundary is complicated both theoretically and computationally by the appearance of fractional order Sobolev norms, e.g. the $H^{1/2}$-norm appearing in the classical variational formulation of the BVP. In [111] the inhomogeneous Dirichlet problem is treated theoretically with the transposition method. Also in [111], the concept of very weak solutions is introduced, which allows to employ $L^2$-norms and ensures $u \in L^2$ (whereas in the general case, $u$ is only a measure). See [81] for a flow control problem in a finite element setting using fractional norms. For a wavelet-based approach to the numerical solution involving approximations of fractional order Sobolev norms we refer to [45, 46].

We note that when deriving the system (OS), we have in fact derived a concrete application of the Lagrange multiplier rule. Here, we explicitly *defined* the adjoint variable $p$ through (2.15) and exploited the well-posedness of the dual problem. For details on Lagrange multiplier theory in Banach spaces and application to optimal control we refer to [65, 119], see also Section 6.1. Here, without elaborating on the technical details, we remark that optimality systems can be conveniently derived utilizing the Lagrange functional

$$\mathcal{L}(y, u, \lambda) = \mathcal{J}(y, u) + (\lambda, Ay - f - Bu)_{(Y')', Y'}, \tag{2.22}$$

with Lagrange multipliers $\lambda \in (Y')'$. Under suitable assumptions, (in particular a constraint qualification), the Lagrange theory in Banach spaces guarantees the existence of a Lagrange multiplier $\lambda^*$, for which

$$D_{(y,u,\lambda)} L(y^*, u^*, \lambda^*) = 0 \tag{2.23}$$

holds, where $(y^*, u^*)$ is the minimum solution of $\mathcal{J}$, cf. [119]. Using reflexivity of $Y$, the multipliers $\lambda^*$ can be identified with the adjoint variable $p \in Y$. The optimality system (OS) follows from (2.23), differentiating with respect to $\lambda$ recovers the state equation, $D_y L(y^*, u^*, \lambda^*) = 0$ corresponds to the adjoint equation and setting $D_u L(y^*, u^*, \lambda^*) = 0$ gives the optimality condition.

We conclude this section with the remark that the derivation of the optimality system (OS) is a crucial step for the numerical approximation of the optimization problem (OP). In the case that $\mathcal{J}$ is strictly convex and the constraints $\mathcal{C}$ are linear, i.e. a unique solution exists, we conclude from Theorems 2.4 and 2.8 that, in order

to obtain the solution of (OP), in fact we can solve the corresponding optimality system given by (OS). The linear-quadratic, equality-constrained case also provides the basis for nonlinear problems and inequality-constrained problems, since linearization processes yield a sequence of linear systems of the type (OS).

## 2.3 Finite Dimensional Approximation

In the previous exposition we have formulated optimization problems and first-order characterizations of solutions in function spaces. The next stage in the development of a numerical solution method is the construction of a finite dimensional approximation of the optimization problem. Besides the choice of the particular discretization technique, another issue arises in the field of PDE constrained optimization, namely that of the order of discretization and optimization.

### 2.3.1 Optimize-Then-Discretize vs. Discretize-Then-Optimize

At this point, a natural way to proceed is to discretize the optimality system, i.e. to discretize all three coupled equations in (OS). This approach is referred to as *optimize-then-discretize* (OTD), where "optimize" means the derivation of the first-order conditions. However, the order of discretization and derivation of the necessary first-order conditions is by no means given a priori. It is also a perfectly viable approach to first discretize the optimization problem (OP), i.e. derive approximations $\mathcal{J}_h$, $\mathcal{C}_h$ as well as $y, u_h$ and only then formulate the first-order conditions for the finite dimensional optimization problem, see Figure 2.1 for a schematic representation. The second approach is termed *discretize-then-optimize* (DTO). For both the OTD and the DTO approach we obtain a finite dimensional system of equations. For the OTD approach, this system is given by the discretized state equation, the discretized adjoint equation and the discretized optimality condition, see Figure 2.1 bottom left. The system for the DTO approach consists of the discretized state equation, the discrete adjoint equation and the discrete optimality condition, Figure 2.1 bottom right. In the DTO approach one approximates the continuous optimization problem with a sequence of finite-dimensional ones for a discretization parameter $h > 0$. In contrast, the sequence of finite-dimensional systems generated by the OTD approach does not necessarily permit their interpretation as optimality systems of a finite-dimensional optimization problem. For a pure Ritz-Galerkin approach the discretized and the discrete optimality system coincide, however this need not be the case in general. We refer to [53] for a detailed discussion in the context of optimal control problems constrained by a linear convection-diffusion equation. There, a streamline upwind Petrov-Galerkin (SUPG) method is employed for the discretization of the state equation. For the DTO approach, the SUPG-related terms in the discretized state equation induce corresponding terms in the discrete adjoint equation and discrete optimality

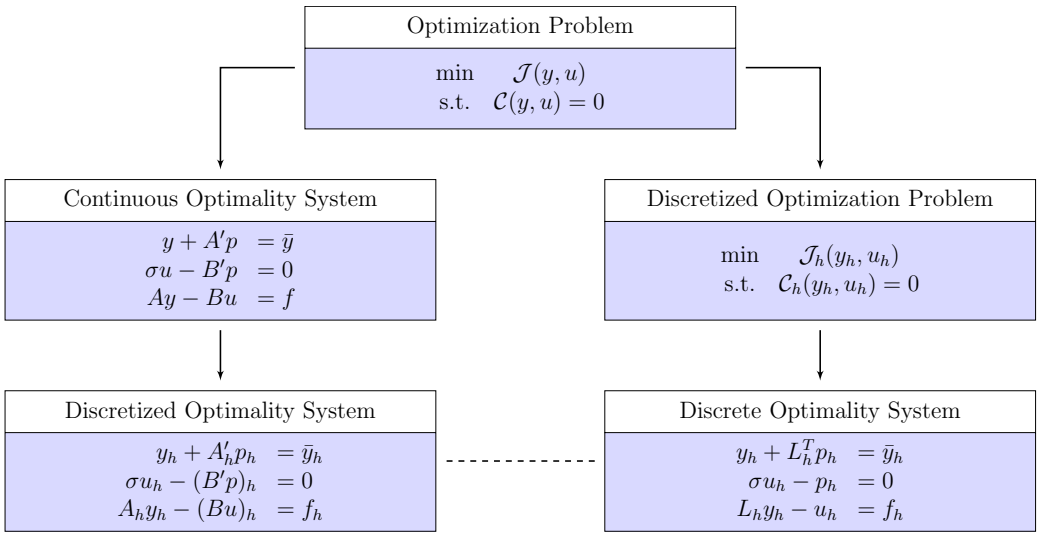



Figure 2.1: Schematic view on the issue of Optimize-Then-Discretize as opposed to Discretize-Then-Optimize.

condition. Thus, the discrete adjoint equation is not a consistent approximation of the continuous adjoint equation in (OS).

On the other hand, the OTD approach yields inconsistent gradients in the sense that the discretized adjoint equation is not the adjoint of the discretized state equation. Thus, the discretized optimality system derived with the OTD approach is not symmetric and can not be the optimality system for some finite dimensional optimization problem. Using the notation in Figure 2.1, under the generally valid assumption that $A_h = L_h$, this means that $A_h' \neq L_h^T$.

No authoritative answer can be given to the question which approach should be favored and some minor issues are regarded by some authors to be of a more philosophical nature, see [79] to which we refer for an in-depth discussion. For now, let us just point out the most important issues. An advantage of the OTD approach is that a suitable discretization method can be chosen independently for each of the three equations in (OS). Consequently, in some cases the approximation quality with respect to the adjoint $p$ is better for the OTD approach [53]. An often cited argument in favor of the DTO approach is the deployment of automatic differentiation tools [77, 122]. Furthermore, the discrete gradients are always consistent and the discrete optimality system is always symmetric, which might enlarge the pool of suitable numerical solution methods.

Being meticulous, one should clearly distinguish between the finite-dimensional approximation of the optimality system derived by the OTD approach and the discrete optimality system derived by the DTO approach. In most numerical examples we consider, both approaches lead to the same discrete system, therefore for reasons of simplicity, we do not distinguish between the discretized and the discrete optimality

system. In the cases where the resulting systems do not coincide, we point this out and employ the DTO approach, obtaining always a symmetric system.

### 2.3.2 Discretization of the State Equation

Example 2.12 will provide the basis for several of our numerical experiments. Therefore, the first step for the finite-dimensional approximation of the corresponding optimality system is the discretization of the state equation, regardless whether the DTO or the OTD approach are employed. The constraints $\mathcal{C}$ are given by the diffusion equation

$$\begin{aligned} -\nabla \cdot (D\nabla y) &= f + u \quad \text{in } \Omega \\ y &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{2.24}$$

where we assume that the domain $\Omega \in \mathbb{R}^d, d = 2, 3$ either is a convex polygon or has a boundary of class $C^{1,1}$. The $d \times d$ - tensor $D$ is assumed to be symmetric with coefficient functions $d_{ij} \in L^\infty(\Omega)$ satisfying the uniform strong ellipticity condition

$$\sum_{i,j=1}^{d} d_{ij} x_i x_j \geq \alpha |x|^2 \quad \forall x \in \mathbb{R}^d \text{ a.e. in } \Omega \tag{2.25}$$

with a constant $\alpha > 0$ independent of $x$.

The diffusion equation (2.24) arises for a wide range of interesting applications. It is extensively applied in geophysics, in particular in groundwater flow and reservoir simulations. Here, the tensor $D$ represents the conductivity (permeability divided by fluid viscosity) and the state $y$ is the piezometric pressure head.[2] In these and other applications, local conservation of mass is of utmost importance. An abundance of discretization methods is applicable to (2.24). However, some of them achieve only global mass conservation, like the classical Galerkin discretization. Yet a range of discretization methods strives for local mass conservation. We mention the classical finite volume method, the support-operator method [142, 143] and the related mimetic finite differences [114], and the multipoint flux approximation [1, 105], to name but a few. We employ the enhanced cell-centered finite differences derived in [7, 8], which can be considered a particular instance of a mixed finite-element method [43], using the lowest order Raviart-Thomas ($\mathrm{RT}_0$) spaces [43, 131] on a general quadrilateral grid. Approximating integrals with trapezoid-midpoint quadrature reduces the mixed finite element method to a cell-centered finite-difference method with a 9-point stencil in two and a 19-point stencil in three dimensions, respectively. In the following we will sketch the basic procedure for $d = 2$. For further details we refer to [7].

[2] Note that for reasons of consistency with the notation of the optimization problem, we deviate from the common notation in the literature, where the pressure would be denoted by $p$, the flux by $u$ and so on.

The starting point is an expanded mixed formulation of (2.24). By introducing the "adjusted gradient" $\tilde{v} = -G^{-1}\nabla y$, with $G$ a positive definite and symmetric tensor, one obtains in variational form the system

$$(\nabla \cdot v, q) = (f + u, q), \qquad q \in L^2(\Omega), \tag{2.26a}$$
$$(G\tilde{v}, w) = (y, \nabla \cdot w), \qquad w \in H(\mathrm{div}), \tag{2.26b}$$
$$(Gv, \tilde{w}) = (GDG\tilde{v}, \tilde{w}), \quad \tilde{w} \in (L^2(\Omega))^d, \tag{2.26c}$$

with $H(\mathrm{div}) = \{w \in (L^2(\Omega))^d : \nabla \cdot w \in L^2(\Omega)\}$. The tensor $G$ will later be based on the local geometry. Note that for $G = D^{-1}$ we obtain $\tilde{v} = v$ and the standard mixed formulation results.

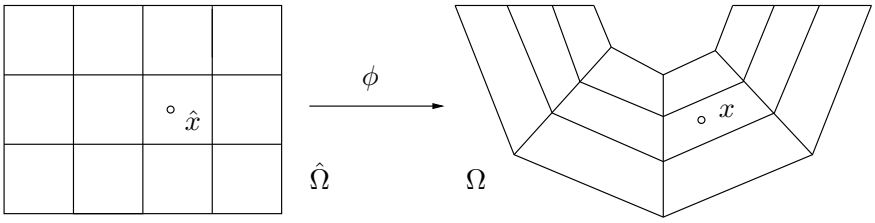



Figure 2.2: Mapping $\phi$ from rectangular computational domain $\hat{\Omega}$ to physical domain $\Omega$.

The discretization is carried out in two steps. First, we define the $\mathrm{RT}_0$ spaces on a rectangular computational grid. Then, a mapping technique is employed to define the approximation spaces on the curvilinear grid, see Figure 2.2.

We denote the computational domain and the corresponding rectangular grid by $\hat{\Omega}$ and $\hat{\mathcal{T}}_h$, respectively. Each element in $\hat{\mathcal{T}}_h$ is a unit square reference element which we denote by $\hat{T}_i,\ i = 1, \ldots, N$. Furthermore, let $\mathcal{T}_h$ be a shape regular partition of the physical domain $\Omega$ into convex quadrilaterals $T_i,\ i = 1, \ldots, N$. The standard pressure and velocity spaces on the reference element are given by

$$\hat{V}_h(\hat{T}_i) = \{\hat{w}_h = (\alpha_1 x_1 + \beta_2, \alpha_1 x_2 + \beta_2, \alpha_3 x_3 + \beta_3)^T : \alpha_j, \beta_j \in \mathbb{R},\ j = 1, 2, 3\} \tag{2.27a}$$

and

$$\hat{W}_h(\hat{T}_i) = \{\alpha : \alpha \in \mathbb{R}\}, \tag{2.27b}$$

respectively. The $\mathrm{RT}_0$ spaces on the rectangular grid $\hat{\mathcal{T}}_h$ are then defined by

$$\hat{V}_h = \{\hat{w}_h : \hat{w}_h \in \hat{V}_h(\hat{T}_i),\ \hat{T}_i \in \hat{\mathcal{T}}_h,\ \hat{w}_h \cdot \nu \text{ continuous}\} \tag{2.28a}$$

and

$$\hat{W}_h = \{\hat{q}_h : \hat{q}_h \in \hat{W}(\hat{T}_i),\ \hat{T}_i \in \hat{\mathcal{T}}_h\}. \tag{2.28b}$$

Now let $\phi : \hat{T}_i \to \mathbb{R}^d$ be a diffeomorphism of a reference element onto an actual element $T_i = \phi(\hat{T}_i)$. We denote its Jacobian matrix by $D\phi$ and the Jacobian determinant by $J\phi$. For a scalar function $\hat{s}(\hat{x})$, we have

$$s(x) = \hat{s} \circ \phi^{-1}(x) \equiv \Phi(\hat{s}). \tag{2.29}$$

For functions $\hat{w} \in \hat{V}_h(\hat{T}_i)$ we employ the Piola transform

$$w(x) = \frac{1}{J\phi} D\phi \hat{w} \circ \phi^{-1}(x) \equiv P\phi(\hat{w}), \tag{2.30}$$

since this preserves the normal components of vector-valued functions across element boundaries. With (2.29) and (2.30), the definition of the $\mathrm{RT}_0$ spaces on $\mathcal{T}_h$ is now straightforward: the local pressure and velocity spaces are given by

$$V_h(T_i) = \{w_h : w_h = P\phi(\hat{w}_h),\ \hat{w}_h \in \hat{V}_h\}, \tag{2.31a}$$

$$W_h(T_i) = \{q_h : q_h = \Phi(\hat{q}_h),\ \hat{q}_h \in \hat{W}_h\}, \tag{2.31b}$$

and from (2.31) we obtain $V_h, W_h$ analogously to (2.28). Now we define the tensor $G$ as

$$G(\phi(\hat{x})) = (J\phi(D\phi^{-1})^T D\phi^{-1})(\hat{x}). \tag{2.32}$$

With this definition follows

$$Gw_h \cdot \tilde{w}_h = (J\phi(D\phi^{-1})^T D\phi^{-1}) P\phi(\hat{w}_h) \cdot P\phi(\hat{\tilde{w}}_h) = \frac{1}{J\phi} \hat{w}_j \cdot \hat{\tilde{w}}_h \tag{2.33}$$

for $w_h, \tilde{w}_h \in V_h$. Approximating the expanded mixed form (2.26) in the spaces $V_h, W_h$ and employing the transformations (2.29), (2.30) and (2.33), we obtain the following finite dimensional problem: find $\hat{v}_h, \hat{\tilde{v}}_h \in \hat{V}_h$ and $\hat{y}_h \in \hat{W}_h$ such that

$$(\hat{\nabla} \cdot \hat{v}_h, \hat{q}_h) = (J(\hat{f}_h + \hat{u}_h), \hat{q}_h), \quad \hat{q}_h \in \hat{W}_h, \tag{2.34a}$$

$$(\hat{\tilde{v}}_h, \hat{w}_h) = (\hat{y}_h, \hat{\nabla} \cdot \hat{w}_h), \qquad \hat{w}_h \in \hat{V}_h, \tag{2.34b}$$

$$(\hat{v}_h, \hat{\tilde{w}}_h) = (\mathcal{D}\hat{\tilde{v}}_h, \hat{\tilde{w}}_h), \qquad \hat{\tilde{w}}_h \in \hat{V}_h, \tag{2.34c}$$

where the tensor $\mathcal{D} = J\phi D\phi^{-1} D D\phi^{-T}$ is the diffusion tensor $D$ modified by the second Piola transform. Note that the transformed diffusion tensor $\mathcal{D}$ is always full on general grids, regardless of the structure of the original diffusion tensor $D$, since it embodies the diffusion coefficients as well as the local geometry.

Without further modification, the linear system associated with (2.34) is an indefinite saddle-point system. Solution methods for such systems will be discussed to some extent in Section 3.1. Here, without further ado, we derive the Schur complement system for the pressure by eliminating (2.34b) and (2.34c) in (2.34a). The integrals appearing on the left side in (2.34b) and (2.34c) are approximated with the

trapezoidal-midpoint rule.[3] This approach, which is related to a lumped mass approximation, yields a diagonal matrix $A_1$ and thus avoids intricate inner-outer iterations for the solution of 2.34. Instead, a single positive definite system for the pressure unknowns is obtained.[4] The integrals on the right side in (2.34c) are approximated with the trapezoidal rule. Denoting the discrete gradient and divergence by $A_2$ and $A_2^T$, respectively, the pressure Schur complement system reads

$$\begin{aligned} M_h(f_h + u_h) &= (A_2^T A_1^{-1} \mathcal{D} A_1^{-1} A_2) y_h \\ &= L_h y_h. \end{aligned} \tag{2.35}$$

Here, $u_h$ are the discrete control unknowns which are computed during the solution of the optimal control problem, whereas $f_h$ is the vector of cell-center evaluations of the given function $f$, thus $M_h f_h$ corresponds to a lumped mass approximation of the right hand side.

We briefly state the relevant error estimates for the scalar pressure unknown. For the detailed convergence analysis including estimates for the flux and the divergence we refer to [7]. We assume that problem (2.24) admits the full $H^2$-regularity (this would be ensured e.g. by requiring $d_{ij} \in C^{0,1}(\bar{\Omega})$, [78, 113]). Furthermore, we require the triangulation $\mathcal{T}_h$ to be $h^2$-uniform, [63]. This condition states that the elements in $\mathcal{T}_h$ are $h^2$-perturbations of parallelograms and therefore is also termed $h^2$-parallelogram in [12]. We denote the $L^2$-projection onto $W_h$ with $\Pi_h$, i.e. for $q \in L^2(\Omega)$ the projection $\Pi_h q$ is defined by

$$(q - \Pi_h q, \tilde{q}) = 0, \ \forall \tilde{q} \in W_h. \tag{2.36}$$

For $\Pi_h$ and $q \in H_0^1(\Omega)$, the well-known estimate

$$\|q - \Pi_h q\| \leq Ch \|q\|_1 \tag{2.37}$$

holds [51]. In view of (2.37), for the discrete pressure solution $y_h$ of (2.34) one can show the optimal $L^2$-estimate

$$\|y - y_h\| \leq Ch, \tag{2.38}$$

where $y$ is the pressure solution of (2.26) and the constant $C$ in (2.38) is independent of the mesh parameter $h$, but depends on $p$, $y$ and $\phi$. In addition, a superconvergence result is obtained with respect to the discrete $L^2$-norm

$$\|q\|_{L^2,h}^2 = \sum_{T_j \in \mathcal{T}_h} |T_j| q(x_j)^2, \tag{2.39}$$

[3] To be precise, for the integration of the $i$-th component of a vector in the $i$-th direction the trapezoidal rule is employed, integrals in the other directions are evaluated with the midpoint rule, cf. [8].

[4] In order to recover the flux, (2.34b) and (2.34c) have to be solved for $v_h$ and $\tilde{v}_h$ after the solution of the pressure system.

where $x_j$ denotes the center of the element $T_j$. The following estimate holds:

$$\|y - y_h\|_{L^2,h} \leq Ch^2. \tag{2.40}$$

Finally we remark that in [7] an enhancement with respect to implementation of boundary conditions is proposed that yields improved approximation results for the flux unknowns, but at the same time renders the finite dimensional system nonsymmetric. In view of the discussion in Section 2.3.1, we employ the DTO approach in case the boundary conditions lead to a nonsymmetric system.

### 2.3.3 The Discrete Optimality System

In order to complete the derivation of a finite-dimensional optimality system, it remains to obtain a discrete adjoint equation and a discrete optimality condition.

**Optimize-Then-Discretize** For the discretization of the adjoint equation in (OS) we again employ the MFE method, following the procedure in Section 2.3.2. This yields

$$L_h p_h = M_h \bar{y}_h - M_h y_h, \tag{2.41}$$

where $\bar{y}_h$ is the vector of cell-center values of the target state $\bar{y}$, analogously to the right hand side discretization in (2.35). The optimality condition in (OS) is discretized as follows. We employ midpoint quadrature for the discretization of the $L^2$-scalar product, which leads to

$$\sigma M_h u_h - M_h p_h = 0. \tag{2.42}$$

Putting it all together, we obtain the discretized optimality system

$$\tilde{K}_h x_h = b_h, \tag{2.43}$$

where $\tilde{K}_h$ is the saddle point matrix

$$\tilde{K}_h = \begin{pmatrix} M_h & 0 & L_h \\ 0 & \sigma M_h & -M_h \\ L_h & -M_h & 0 \end{pmatrix}, \tag{2.44}$$

and

$$x_h = (y_h, u_h, p_h)^T, \quad b_h = (M_h \bar{y}_h, 0, M_h f_h)^T \tag{2.45}$$

are the vector of unknowns and the right hand side vector, respectively. Regarding the discussion in Section 2.3.1, note that the matrix $\tilde{K}_h$ is only symmetric if the discretization of the state equation leads to a symmetric $L_h$.

We remark that we do not distinguish between a discrete operator and its matrix representation. From now on we assume that the ordering of the optimization variables is given by (2.45), and each of the unknowns $y_h, u_h$ and $p_h$ is ordered lexicographically.

**Discretize-Then-Optimize** In the DTO approach the state equation is discretized as before with the MFE method of Section 2.3.2. It remains to discretize $\mathcal{J}$. To this end, the target state is discretized in the same way as above, yielding $\bar{y}_h$, and the integrals appearing in the norms in $\mathcal{J}$ are discretized consistent with the state equation, yielding the the discretized objective functional

$$\begin{aligned}\mathcal{J}_h(y_h, u_h) &= \frac{1}{2}\|y_h - \bar{y}_h\|^2 + \frac{\sigma}{2} && (2.46)\\ &= \frac{1}{2}y_h^T M_h y_h - y_h^T M_h \bar{y}_h + \bar{y}_h^T M_h \bar{y}_h + \frac{\sigma}{2}u_h^T M_h u_h. && (2.47)\end{aligned}$$

The finite dimensional optimization problem now reads

$$\begin{aligned}\text{minimize} \quad & \mathcal{J}_h(y_h, u_h) \\ \text{subject to} \quad & \mathcal{C}_h(y_h, u_h) = 0,\end{aligned} \tag{2.48}$$

where the discretized constraints $\mathcal{C}_h$ are given by (2.35). The optimality conditions for (2.48) can be easily derived using the discrete Lagrangian

$$\begin{aligned}\mathcal{L}_h(y_h, u_h, p_h) &= \mathcal{J}_h(y_h, u_h) + p_h^T \mathcal{C}_h(y_h, u_h) && (2.49)\\ &= \frac{1}{2}y_h^T M_h y_h - y_h^T M_h \bar{y}_h + \bar{y}_h^T M_h \bar{y}_h + \frac{\sigma}{2}u_h^T M_h u_h \\ &\quad + p_h^T(L_h y_h - M_h u_h - M_h f_h),\end{aligned}$$

where $p_h$ are the Lagrange multipliers, see e.g. [127, Chap. 12]. The first-order conditions for (2.48) are given by

$$\begin{aligned}0 = \nabla_{y_h}\mathcal{L}_h(y_h, u_h, p_h) &= \nabla_{y_h}\mathcal{J}_h(y_h, u_h) + p_h^T \nabla_{y_h}\mathcal{C}_h(y_h, u_h) && (2.50a)\\ &= M_h y_h - M_h \bar{y}_h + L_h^T p_h && (2.50b)\\ 0 = \nabla_{u_h}\mathcal{L}_h(y_h, u_h, p_h) &= \nabla_{u_h}\mathcal{J}_h(y_h, u_h) + p_h^T \nabla_{u_h}\mathcal{C}_h(y_h, u_h) && (2.50c)\\ &= \sigma M_h u_h - M_h p_h && (2.50d)\\ 0 = \nabla_{p_h}\mathcal{L}_h(y_h, u_h, p_h) &= \mathcal{C}_h(y_h, u_h) && (2.50e)\\ &= L_h y_h - M_h f_h - M_h u_h. && (2.50f)\end{aligned}$$

Here, $\nabla_{y_h}$ denotes the gradient with respect to the unknowns $y_h$ only, and likewise for the remaining unknowns $u_h$ and $p_h$. Writing (2.50) in matrix form yields again a saddle point system

$$K_h x_h = b_h, \tag{2.51}$$

with the unknown vector and the right hand side given by (2.45). Here, the saddle point matrix is given by

$$K_h = \begin{pmatrix} M_h & 0 & L_h^T \\ 0 & \sigma M_h & -M_h \\ L_h & -M_h & 0 \end{pmatrix}. \tag{2.52}$$

Note that in contrast to $\tilde{K}_h$ in (2.43), the matrix $K_h$ derived by the DTO approach is always symmetric. From now on we will not distinguish anymore between $K_h$ and $\tilde{K}_h$ but instead we assume that either the discretized system was derived employing the DTO approach or the discretization following the OTD approach yields a symmetric saddle point matrix.

Finally we remark that in the context of quadratic programming, a matrix of the type (2.52) is called a Karush-Kuhn-Tucker (KKT) matrix, in reference to the Karush-Kuhn-Tucker conditions of first-order optimality, cf. [127]. The name "saddle point system" originates from the property that a solution $(y_h^*, u_h^*, p_h^*)$ of (2.51) is a saddle point for the Lagrangian (2.49), i.e.

$$\mathcal{L}_h(y_h^*, u_h^*, p_h) \leq \mathcal{L}_h(y_h^*, u_h^*, p_h^*) \leq \mathcal{L}_h(y_h, u_h, p_h^*), \quad y_h, u_h, p_h \in \mathbb{R}^N. \tag{2.53}$$

Frequently, it will be useful to consider the generic $2 \times 2$ saddle point matrix

$$\begin{pmatrix} H & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix} = \begin{pmatrix} b_x \\ b_p \end{pmatrix}, \tag{2.54}$$

which is obtained if we collect the optimization variables $y \in \mathbb{R}^N$ and $u \in \mathbb{R}^N$ in the single vector of unknowns $x \in \mathbb{R}^{2N}$, and the square submatrix $H \in \mathbb{R}^{2N \times 2N}$ and the rectangular submatrix $C \in \mathbb{R}^{N \times 2N}$ are defined by

$$H = \begin{bmatrix} M_h & \\ & \sigma M_h \end{bmatrix}, \quad C = \begin{bmatrix} L_h & -M_h \end{bmatrix}. \tag{2.55}$$

The letter $H$ indicates that the upper left block in (2.54) originates from the Hessian of the discrete Lagrangian (2.49), the letter $C$ stands for the discrete constraints. As before, we denote the saddle point matrix by $K$. The form (2.54) naturally appears when stating the optimality conditions as a saddle point problem in Hilbert spaces, cf. Appendix A.1. Many algorithms from numerical optimization apply to the generic formulation (2.54) and no particular structure of the unknown vector $x$ is assumed. In the present case it is crucial to exploit the knowledge of the additional structure imposed on $x$ due to the PDE-based partitioning.

### The Linear-Quadratic Model Problem

Let us consider a special case of (2.51) which will frequently serve as example. To this end, let the optimization problem given by Example 2.1 where the constraints are discretized on a uniform grid. Then $L_h$ corresponds to a standard five-point discretization of $-\Delta$ and $M_h = h^2 I$ (the MFE discretization of Section 2.3.2 on a uniform grid and $D = I$ reduces to the standard 5-point stencil). Although this is a very basic example for an optimization problem, it exhibits several typical properties

and already foreshadows some difficulties which can arise in more complex applications. For future reference we define the discrete optimality system

$$\begin{aligned} h^2 y_h - \Delta_h p_h &= h^2 \bar{y}_h \\ \sigma h^2 u_h - h^2 p_h &= 0 \\ -\Delta_h y_h - h^2 u_h &= h^2 f_h \end{aligned} \tag{LQP$_h$}$$

where $-\Delta_h$ denotes the five-point stencil.

### Stability and Convergence

A matrix $K_h$ given by (2.52) is nonsingular if $C$ in (2.55) has full row rank and $H$ in (2.55) is regular on $\ker C$. However, to obtain a stable numerical method we need that $K_h^{-1}$ is uniformly bounded as $h \to 0$. If the OTD approach is applicable (i.e. in the symmetric and unconstrained case $\mathcal{U}_{\text{ad}} = U$), by exploiting the connection between the minimization problem (OP) and the saddle point problem defined by the first-order conditions, stability results are obtained within the approximation theory of saddle point problems [43], see [31] and Appendix A.1. Results on the convergence order then follow from (A.10).

With respect to the DTO approach, and including the case $\mathcal{U}_{\text{ad}} \subset U$, first estimates for the errors $\|y^* - y_h^*\|, \|u^* - u_h^*\|$ in the case of distributed control have been given in [64]. In particular, for a piecewise constant approximation $u_h$ the optimal order estimate

$$\|u^* - u_h^*\|_{L^2(\Omega)} = \mathcal{O}(h) \tag{2.56}$$

has been proved. Similar results have been published recently in [50]. Corresponding results in a finite difference setting have been proved in [36], employing the notion of discrete $H^m$-regularity of $L_h$ [86]. In [94], a so-called variational concept is proposed where $u$ is not explicitly discretized but is obtained by a projection step $u_h^* = \Pi_{\mathcal{U}_{\text{ad}}} p_h^*$, thus essentially inheriting the approximation order of $p_h$ (which is, e.g., $\mathcal{O}(h^2)$ if $H^2$-regularity holds for the adjoint equation and piecewise linear finite elements are employed). In [121], a post-processing step is included which increases the order for $u_h^*$. Adaptive finite element discretizations have been considered in [21].

In any case, due to our choice of discretization for state and adjoint, we obtain overall $\mathcal{O}(h)$-convergence in $L^2$. Furthermore, for $y_h$ and $p_h$, $h^2$-order superconvergence with respect to cell-center values holds. In [121], a supercloseness result yielding second-order superconvergence at the center of elements is shown to hold for piecewise constant $u_h$. The superconvergence in a discrete $L^2$-norm is also observed in our numerical experiments, cf. Section 4.5.

### 2.3.4 Some Properties of Saddle Point Matrices

Recall that each block in the $3 \times 3$ saddle point matrix $K_h$ is a matrix of dimension $N$, where $N$ is the cardinality of the triangulation $\mathcal{T}_h$, see Section 2.3.2.[5] This is due to several facts: first, we consider the case of distributed control which implies that the number of $u_h$ is of the same order as that of $y_H$, and second, we use the same finite dimensional space for the approximation of all three unknowns. As was discussed in Section 2.3.1, at least for the OTD approach a different discretization e.g. for the adjoints could lead to a different dimensionality of the discrete equation.

An important characteristic of $K$ is given by its inertia

$$\text{inertia}(K) = (2N, N, 0), \tag{2.57}$$

i.e. $K$ has $2N$ positive and $N$ negative eigenvalues. Since the number of negative eigenvalues is of the same order of magnitude as those with positive signs, we say that $K$ is *strongly indefinite*. This has an adverse effect on the convergence properties of many solution methods. For instance, Krylov solvers are known to converge rather slow for problems with indefinite coefficient matrices and effective preconditioning is mandatory, cf. Section 3.1.2. We remark that, employing least-squares finite element techniques, one can derive a positive definite and symmetric coefficient matrix for the saddle point problem [31]. However, this can imply a squaring of certain operators and essentially a corresponding increase of the associated condition numbers. Further difficulties might arise due to the need to compute discrete negative index Sobolev norms, and, employing first-order least-squares formulations for the constraints implies a significant increase in the number of unknowns. Therefore it is unclear whether a numerical solution method can effectively benefit from the theoretical advantage of an SPD coefficient matrix. To our knowledge, no practical implementation of a solution method for PDE constrained optimization based on least-squares techniques has been published.

The following Theorem gives further insight in the spectral properties of $K$ [137]

**Theorem 2.13.** *Assume the Hessian block $H$ in (2.54) is positive definite symmetric and denote its eigenvalues with $0 < \mu_1 \leq \mu_2 \leq \ldots \leq \mu_{2N}$. Furthermore, since $C$ has full row rank $N$ it has $N$ singular values which we denote by $0 < \sigma_1 \leq \sigma_2 \leq \ldots \leq \sigma_N$. Denoting the spectrum of $K$ by $\Lambda(K)$ we have*

$$\Lambda(K) \subset I^- \cup I^+ \tag{2.58}$$

*with*

$$I^- = \Big(\frac{1}{2}(\mu_1 - \sqrt{\mu_1^2 + 4\sigma_N^2}), \frac{1}{2}(\mu_{2N} - \sqrt{\mu_{2N}^2 + 4\sigma_1^2})\Big) \tag{2.59}$$

[5] For a uniform grid, we have $N = h^{-2}$, with $h$ the mesh size.

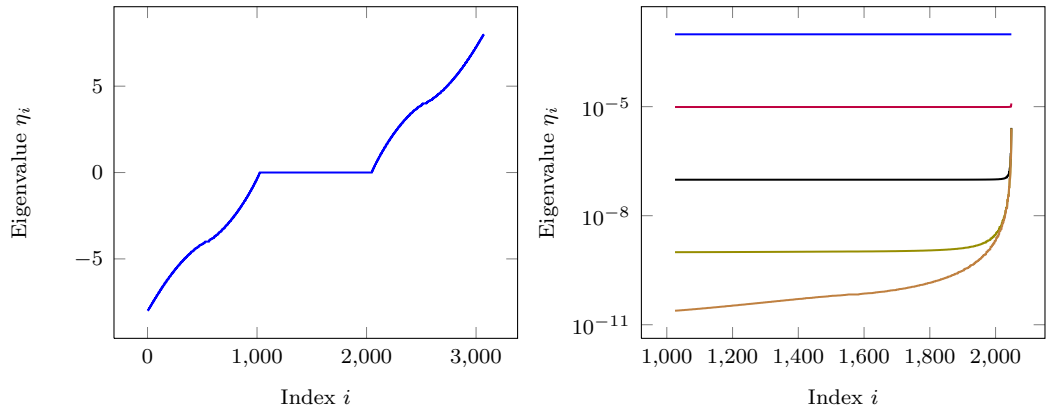



Figure 2.3: Left: Eigenvalues $\eta_i, i = 1, \ldots, 3072$ of $K$ for the linear-quadratic model problem with $h = 1/32$ and regularization parameter $\sigma = 1.0$. Right: Eigenvalues $\eta_i \in I^+, i = 1025, \ldots, 2048$ for — $\sigma = 1.0$, — $\sigma = 1.0_{-2}$, — $\sigma = 1.0_{-4}$, — $\sigma = 1.0_{-6}$ and — $\sigma = 1.0_{-8}$.

*and*

$$I^+ = \left(\mu_1, \frac{1}{2}(\mu_{2N} + \sqrt{\mu_{2N}^2 + 4\sigma_N^2})\right). \tag{2.60}$$

For $(\mathrm{LQP}_h)$ we have $\mu_1 = \ldots = \mu_N = \sigma h^2$, $\mu_{N+1} = \ldots = \mu_{2N} = h^2$ and $\sigma_i = \sqrt{\lambda_i^2 + h^4}$, where $\lambda_i$ is the $i$-th eigenvalue of the negative discrete Laplacian. Figure 2.3 (left) shows the eigenvalue distribution of $K$ for $(\mathrm{LQP}_h)$ and the mesh size $h = 1/32$, i.e. the dimension of $K$ is $n = 3h^{-2} = 3072$. The regularization parameter is chosen as $\sigma = 1.0$. In Figure 2.3 (right) we plot the lower half of the eigenvalues in $I^+$ for different values of $\sigma$. We see that for larger values of $\sigma$ the lower half of the eigenvalues in $I^+$ is strongly clustered around the minimal absolute eigenvalue $\mu_1 = \sigma h^2$, however for fairly small values the eigenvalues decay rapidly to zero and are not clustered anymore. This will have an additional adverse effect on iterative solvers, unless suitable preconditioners are developed. A further difficulty is given by potentially large values of the condition number of $K$, which is given by

$$\kappa(K) = \frac{\max|\eta|}{\min|\eta|}, \ \eta \in \Lambda(K). \tag{2.61}$$

From (2.59) and (2.60) we gather that asymptotically for $h \to 0$

$$\kappa(K) \sim \frac{\lambda_{\max}}{\sigma h^2}, \tag{2.62}$$

where $\lambda_{\max} \approx 8$ is the largest eigenvalue of the discrete negative Laplacian $L_h$. We see that for any fixed value of $\sigma$ the condition number $\kappa(K)$ asymptotically grows like

Table 2.1: Condition numbers $\kappa(K)$ for different values of the regularization parameter $\sigma$ and different mesh sizes $h^{-J}$.

| | $J$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\sigma$ | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $1.0_{0}$ | $4.930_{2}$ | $2.029_{3}$ | $8.173_{3}$ | $3.275_{4}$ | $1.311_{5}$ | $5.243_{5}$ | $2.097_{6}$ | $8.389_{6}$ |
| $1.0_{-2}$ | $4.930_{4}$ | $2.029_{5}$ | $8.173_{5}$ | $3.275_{6}$ | $1.311_{7}$ | $5.243_{7}$ | $2.097_{8}$ | $8.389_{8}$ |
| $1.0_{-4}$ | $4.930_{6}$ | $2.029_{7}$ | $8.173_{7}$ | $3.275_{8}$ | $1.311_{9}$ | $5.243_{9}$ | $2.097_{10}$ | $8.389_{10}$ |
| $1.0_{-6}$ | $4.930_{8}$ | $2.029_{9}$ | $8.173_{9}$ | $3.275_{10}$ | $1.311_{11}$ | $5.243_{11}$ | $2.097_{12}$ | $8.389_{12}$ |
| $1.0_{-8}$ | $4.930_{10}$ | $2.029_{11}$ | $8.173_{11}$ | $3.275_{12}$ | $1.311_{13}$ | $5.243_{13}$ | $2.097_{14}$ | $8.389_{14}$ |
| $1.0_{-10}$ | $4.930_{12}$ | $2.029_{13}$ | $8.173_{13}$ | $3.275_{14}$ | $1.311_{15}$ | $5.243_{15}$ | $2.097_{16}$ | $8.389_{16}$ |

$\mathcal{O}(h^{-2})$, i.e. the asymptotic behavior of the discretized second-order PDE operator is dominant. On the other hand, for any fixed mesh size $h$, we see that $\kappa(K) \sim \mathcal{O}(1/\sigma)$. For $\sigma \to 0$ the strong growth of $\kappa(K)$ can lead to severe difficulties of iterative solvers and, for example, the maximum attainable accuracy of Krylov methods can be adversely affected, cf. Section 3.2. In Table 2.1 we report the condition numbers of $K$ according to (2.59) and (2.60) for different values of $\sigma$ and mesh sizes $h = 2^{-J}$, $J = 3, 4, \ldots, 10$. The large condition numbers arising for combinations of small $h$ and $\sigma$ make it clear that a reduction of $\kappa(K_h)$ is compulsory should any iterative method achieve reliable results and satisfying convergence rates. The results of this section show that we have to deal with two possible sources of ill-conditioning when devising an efficient solution method for discretized PDE constrained optimization problems: the mesh size $h$ as well as the regularization parameter $\sigma$. We remark that in most publications, concerning optimal control, the values of $\sigma$ are moderate and usually fixed a priori. The problem-dependent determination of $\sigma$ is a complex subtopic in inverse problems [153].

## Summary

The contents of this chapter serve as the foundation for the further development of efficient algorithms for the numerical solution of discretized PDE constrained optimization problems. We discussed the relevant results concerning existence and uniqueness of solutions in the continuous setting. Optimal solutions have been characterized by necessary and sufficient first-order conditions which constitute a coupled system of partial differential equations termed the optimality system. Discretization of the PDE constrained optimization problems has been carried out by discretizing the corresponding optimality system, which is the optimize-then-discretize approach, and by discretizing the optimization problem, which corresponds to the discretize-then-optimize approach. For the particular instance of the constraints given by the diffu-

sion equation, we carried out the discretization on general quadrilateral grids with the lowest-order Raviart-Thomas mixed method which, using quadrature, yields a cell-centered finite difference stencil. We discussed the stability of the discrete optimality system and concluded the chapter with the discussion of some properties of the arising saddle point matrices.

# 3 A One-Level Method for the Numerical Solution of Saddle Point Systems Arising in PDE-Constrained Optimization

A major objective of this thesis is to develop a solution method that allows to solve discretized PDE constrained optimization problems with optimal complexity, i.e. with computational cost proportional to the number of discrete unknowns. As a preliminary step, in the present chapter we devise a one-level method building on constraint preconditioning. In the first section, we provide an overview of solution methods for symmetric saddle point systems (2.54) and we will describe a specific constraint preconditioner in detail.

## 3.1 Numerical Methods for the Solution of Saddle Point Systems

Saddle point systems do appear not only in the context of quadratic programming, but they arise from an abundance of different applications, such as computational fluid dynamics, discretizations of mixed variational formulations, interior point methods and many more. Arguably the most prominent example is given by the Stokes equation which describes the laminar flow of a strongly viscous fluid. There, the matrix $H$ is a discretized (vector-)Laplacian and the constraints $C$ result from the mass conservation or incompressibility condition. Thus, $C$ is the discretized divergence and $C^T$ is a discrete (negative) gradient. Similar systems arise at each substep for a linearization process of the Navier-Stokes equations. For instance, the Picard linearization of the Navier-Stokes equation leads to the Oseen equations. There, the matrix $H$ represents a convection-diffusion operator, the constraints are again given by the incompressibility condition. An example for a saddle point system originating from a mixed variational formulation has been encountered in Section 2.3.2. There, the discretization technique using a particular mixed finite element method led to the saddle point system (2.34). However, as was described briefly in Section 2.3.2, an elimination approach in conjunction with carefully chosen quadrature reduced the saddle point system to a positive definite system in the pressure unknowns alone.

Due to the ubiquity of saddle point systems in computational science, a large variety of solution methods has emerged over the years and naturally we can only briefly sketch the most important approaches. For a survey on numerical methods for saddle point

systems we refer to [22], methods specific to the field of quadratic programming and optimization can be found in [69, 127]. In the following discussion we limit ourselves to those methods that are most widely used in quadratic programming and have the potential to serve as building blocks for other algorithms such as multigrid methods.

According to [22], numerical methods for the solution of saddle point systems can be classified into two broad categories: *segregated* and *coupled* or "all at once" methods. In segregated methods, the system (2.54) is decoupled into smaller subsystems for the unknowns $x$ and $p$. These reduced systems are then solved separately, usually coupled together in an outer iterative scheme. In contrast, coupled methods solve simultaneously for all components of the unknown vector $x$. To this end, either direct factorizations of the coefficient matrix $K$ or iterative algorithms like Krylov methods are employed. We defer the discussion of multigrid methods until Section 4.2.

### 3.1.1 Segregated Methods

The popularity of segregated methods is based on the fact that the decoupling strategy allows to build on well-studied solution methods available for equations arising from the decoupling or elimination. Two frequently employed instances of segregated solution methods are the range space and the null space method.

**Range Space Methods** The range space method is essentially the reduction of the system (2.54) to a system for the Schur complement. For this method it is required that $H$ is invertible. The reduction can easily be derived considering the block factorization

$$K = \begin{pmatrix} H & C^T \\ C & 0 \end{pmatrix} = \begin{pmatrix} I & \\ CH^{-1} & I \end{pmatrix} \begin{pmatrix} H & C^T \\ & S \end{pmatrix}, \tag{3.1}$$

where $S = -CH^{-1}C^T$ is the Schur complement. From (3.1) follows that $K$ is regular if and only if the Schur complement is. Multiplying the system (2.54) with the inverse of the left factor in (3.1), we obtain

$$\begin{pmatrix} H & C^T \\ & S \end{pmatrix} \begin{pmatrix} x \\ p \end{pmatrix} = \begin{pmatrix} f_x \\ f_p - CH^{-1}f_x \end{pmatrix}. \tag{3.2}$$

The solution of the original system (2.54) is then computed by blockwise backsubstitution in (3.2). In general, the largest part of the computational effort has to be expended for the solution of the system involving $S$, given by the second block row in (3.2). A major advantage of reduction method such as the range space approach is that the subsystems to be solved are of smaller dimension than the original problem, in our case of dimension $N$ instead of $3N$.[1] However, a potential drawback is that

[1] This advantage is more pronounced when the dimension of the control space is considerably smaller than that of the state space.

the Schur complement $S$ might be a dense matrix, even if $H$ and $C$ are sparse. Dense Schur complements occur for example in the aforementioned application to the Stokes and Oseen equations. An example for a sparse Schur complement in the context of groundwater flow is given in [117], and not least we again mention the derivation of the discrete system for the state equation, see Section 2.3.2.

In order to avoid the explicit construction of $S$, the Schur complement system could be solved iteratively. Provided that systems with $H$ can be solved efficiently (e.g. by multigrid methods), the matrix-vector product with $S$ can be computed at reasonable cost. Difficulties with this approach are deriving reliable stopping criteria for the nested inner-outer scheme and preconditioning of $S$, since individual entries of $S$ can not be accessed explicitly.

**The Uzawa Iteration** Closely related to the range space approach is the Uzawa method [13], which can also be interpreted as an iterative method for the solution of the Schur complement system. The Uzawa iteration is one of the earliest methods developed for the iterative solution of saddle point systems and in particular inexact variants find wide-spread use in computational fluid dynamics. The classical Uzawa iteration for the system (2.54) is given by

$$x^{k+1} = H^{-1}(f_x - C^T p^k) \tag{3.3a}$$

$$p^{k+1} = p^k + \alpha(Cx^{k+1} - f_p), \tag{3.3b}$$

with a relaxation parameter $\alpha > 0$. At each step, the solution of a system with coefficient matrix $H$ is required. The connection to the range space method becomes apparent if we eliminate $x^{k+1}$ from the update step (3.3b) for $p^{k+1}$. This yields

$$p^{k+1} = p^k + \alpha(CH^{-1}f_x - f_p - CH^{-1}C^T p^k), \tag{3.4}$$

which is precisely a step of the stationary Richardson iteration for the Schur complement system in (3.2). Without suitable preconditioning convergence of the Uzawa iteration is rather slow, a notable exception being the steady-state Stokes equation. In that case, the Schur complement is spectrally equivalent to the identity matrix and the Uzawa iteration converges at a rate which is independent of the discretization parameter $h$. A further difficulty is to determine the optimal relaxation parameter $\alpha$, which at least in principle requires estimates for the extremal eigenvalues of the Schur complement matrix.

Important modifications of Uzawa's method for practical problems are given by inexact variants [39, 61]. Here, the system involving the coefficient matrix $H$ is not solved exactly but only approximately, for instance using an iterative method or multigrid [152].

**Null Space Methods** A frequently employed approach in optimization is the null space method. From an abstract viewpoint there is a close connection to the range

space method, since in fact the null space method can be derived by applying the range space method to a suitable permutation of the matrix $K$. However, in contrast to the range space approach, the null space method does not require $H$ to be regular. Instead, it is sufficient for $H$ to be invertible on the null space of the constraints, i.e. the condition $\ker(H) \cap \ker(C) = \{0\}$ has to hold. In contrast to the full regularity of $H$, this condition is often satisfied for constrained minimization problems, cf. the well-known inf-sup condition. For a practical algorithm using the null space method, a representation of a basis for the null space of $C$ is needed. For now let us assume that such a basis is given by the columns of $Z \in \mathbb{R}^{2N \times N}$, i.e. $CZ = 0$. To determine such a $Z$ is a crucial step in the null space method and will be discussed after the description of the algorithm has been completed. Let $Y \in \mathbb{R}^{2N \times N}$ be any regular matrix such that the $2N \times 2N$ matrix $[Y\ Z]$ is regular. In particular, this implies that $Y$ is a basis for the range space of $C^T$. Using $Y$ and $Z$, we partition a vector $x$ as

$$x = Yx_Y + Zx_Z, \tag{3.5}$$

with the range space component $x_Y \in \mathbb{R}^N$ and the null space component $x_Z \in \mathbb{R}^N$. Substituting (3.5) into the second block row of (2.54) yields

$$CYx_Y = f_p. \tag{3.6}$$

Note that $CY$ is a regular $N \times N$ matrix and thus the range space component is well-defined as the solution of (3.6). By substituting the decomposition (3.5) into the first block row of (2.54) we obtain

$$HYx_Y + HZx_Z + C^T p = f_x, \tag{3.7}$$

and multiplying this expression from the left with $Z^T$ yields

$$Z^T HZx_z = Z^T f_x - Z^T HYx_Y, \tag{3.8}$$

where again we used $CZ = 0$. After solving (3.8) for $x_Z$, $x$ is obtained from (3.5). The adjoint unknown $p$ then is computed from

$$(CY)^T p = Y^T f_x - Y^T Hx, \tag{3.9}$$

which is derived from the first row of (2.54) by multiplication with $Y^T$.

The $N \times N$ matrix $Z^T HZ$, which we denote by $H_Z$, is called the *reduced Hessian*. The major computational effort of the null space method consists in computing a suitable null space basis $Z$ and in solving (3.8). Due to assumptions, $H_Z$ is positive definite and furthermore symmetric, (3.8) can be solved iteratively with the conjugate gradient (CG) method. Indeed, this is a popular choice since it avoids the need to explicitly form $H_Z$ and uses only matrix-vector products with the factors $Z, H$ and $Z^T$. However, here the same difficulty as in the range space method, namely

preconditioning a non-assembled matrix, arises. The popularity of the null space method in optimization is related to the fact that in nonlinear problems, frequently approximations of $H_Z$ are maintained with cheap low-rank updates from the Broyden family, such as BFGS.

Computing $Z$ may easily be the most demanding step in the application of the null space method. Further, the condition number of $H_Z$ and hence the efficiency of the null space method depend strongly on the particular choice of $Z$. If the problem size permits the use of direct methods, in general a null space basis is computed by sparse elimination techniques applied to the constraint matrix $C$, preferably in such a way that the columns of $Z$ are orthonormal. In the context of PDE constrained optimization, a suitable null space basis can be expressed in terms of solutions with the state operator. For $C$ as in (2.55), we define

$$Z = \begin{bmatrix} L_h^{-1} M_h \\ I \end{bmatrix}. \tag{3.10}$$

It remains to define $Y$. From an algorithmic viewpoint, the choice

$$Y = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{3.11}$$

best suits our needs. Other possible choices for $Y$ are discussed in [127]. The choice of a range space basis $Y$ has a much smaller impact on the algorithm, compared to $Z$. In fact, some variations of the null space method are motivated by this observation and neglect the *cross-term* $Z^T H Y$ on the right hand side of (3.8). With (3.10) and (3.11), the reduced Hessian for our model problem is given by

$$H_Z = M_h^T L_h^{-T} M_h L_h^{-1} M_h + \sigma M_h. \tag{3.12}$$

The null space method and numerous variants are of particular interest in nonlinear problems, where they appear as reduced Hessian, reduced Newton or reduced sequential quadratic programming (SQP) methods [29, 66, 67, 127]. These methods may be derived by applying Newton's method to the reduced functional $\hat{\mathcal{J}}_h(u_h)$. Finally let us remark that the null space method is closely connected to the family of *constraint preconditioning* techniques which will be discussed in Section 3.1.3.

### 3.1.2 Coupled Methods

In contrast to the introduced segregated methods are the coupled or "all at once" solution approaches with the objective to solve simultaneously for all components of the unknown vector of the KKT system (2.54). This can be done either with direct methods or in an iterative fashion. The interest in coupled approaches arises due to the high computational cost induced by the iterative solution of the Schur complement

systems in the reduced methods. In particular note that the number of iterations required to invert, say, $H_Z$, can not be reduced by a higher efficiency of the inner solver related to $Z$. The all-at-once approaches aim at reducing the cost by solving for $C, C^T$ only once during the course of the iterations.

**Direct Factorization Methods** In direct methods, first a triangular factorization of the KKT matrix $K$ is computed and the solution of the KKT system is then achieved by forward elimination and backward substitution. The Cholesky factorization can not be used, since $K$ is indefinite. Instead, variants of the LU decomposition, accounting for a given sparsity pattern to reduce fill-in, are commonly employed. The factorization process consists of several phases: the first phase is an ordering phase that exploits the sparsity pattern in order to reduce fill-in, the second phase is an analyzing or symbolic phase in which a pivot sequence and necessary data structures are determined and the last phase is a numerical phase in which the actual factorization is computed.

Over the years, many fine-tuned algorithms based on the methods proposed in [59] have been developed. Implementations which find wide-spread use in the optimization community are the MA27 and MA47 algorithms of the HSL library [99]. Algorithm MA47 is especially well adapted to the solution of saddle point systems since it takes into account that no pivots can be chosen from the $(2, 2)$ zero block in $K$.

Direct methods enjoy a considerable popularity in the optimization community, which in part is owed to the robustness properties and to the ability to serve as a black-box method, since in many cases the constraint Jacobian might be dense or no special structure can be assumed, different to the PDE constrained setting. On the other hand, regardless of the high level of sophistication direct methods have reached over time, the problem size which can be treated by these approaches is limited both by memory and computational workload constraints, in particular in the PDE constrained context. Since we focus here on PDE related problems, we refer to [59] and the numerous references given in [22].

**Krylov Methods** Iterative methods provide an alternative to direct factorizations that is less demanding on computational resources. Only matrix-vector products with the system matrix $K$ are needed during the course of the iterations. Thus, only $K$ and a few work vectors have to be stored. We remark though that this low demand on computational effort has to be put into perspective as soon as preconditioning issues enter the solution method.[2] Iterative techniques can be divided into the groups of stationary and nonstationary methods. In stationary methods, iteration matrices and other used information are independent on the current iteration index. In contrast, nonstationary methods depend on the available information at a given iteration. Stationary methods are not very effective by themselves and are rarely used as stand-alone

[2]Preconditioners usually require access to individual matrix elements and can utilize techniques that require a significant amount of computational cost.

solvers. They will be briefly touched upon in the context of smoothing iterations for multigrid method, cf. Section 4.1. In the following we will restrict ourselves to the presentation of Krylov subspace iterations for indefinite problems. Arguably the best-known representative of Krylov iterations is the conjugate gradient method, however it can not be applied to solve the KKT system (2.54) since it is applicable only to positive definite symmetric systems (but see the remark in Section 3.1.3 for an exception to this rule).[3] Since Krylov methods also play an important role in the multigrid context – they are frequently used as smoothing iteration as well as accelerator on the fine grid – we will provide some further details.

Given an initial guess $x^0$ and corresponding residual $r^0 = b - Kx^0$, a Krylov method generates a sequence of iterates $x^k$ in the space $x^0 + \mathcal{K}^k(K, r^0)$ with the Krylov subspace

$$\mathcal{K}^k(K, r^0) = \text{span}(r^0, Kr^0, \ldots, K^{k-1}r^0). \tag{3.13}$$

Uniqueness of the iterate $x^k$ is enforced by requiring that the corresponding residual $r^k$ is orthogonal (with respect to the Euclidean inner product) to a $k$-dimensional space. Different choices for this spaces and ways of enforcing or approximating the orthogonality gives rise to a variety of Krylov methods. For indefinite problems, widely used algorithms include the minimal residual variants MINRES [130] and GMRES [139]. Here, orthogonality is achieved by minimizing the Euclidean norm of $r^k$ over the affine subspace

$$r^0 + K\mathcal{K}^k(K, r^0). \tag{3.14}$$

The MINRES method is applicable to symmetric systems and requires a positive definite preconditioner, the GMRES algorithm is a further generalization applicable also to non-symmetric systems and can use an indefinite preconditioner.

The Krylov methods outlined above belong to the class of *optimal* methods, to be understood in the sense that full orthogonality of residuals is guaranteed. Full orthogonality and short-term recurrence for vector updates such as in cg type iterations can in general not be achieved at the same time. The optimal GMRES method for instance requires $k$ vector operations in iteration $k$. Consequently, the Krylov subspace grows in each iteration and may become unacceptably large. Therefore, several methods have been developed that compromise on the full orthogonality in favor of affordable cost per iteration step. To this class of non-optimal Krylov methods belong the short-term recurrence based Quasi Minimal Residual (QMR) and BiCG-type variants as well as restarted and truncated versions of optimal methods. For a further discussion on different variants of Krylov methods we refer to [19] and references therein.

It is well-known that the convergence of cg-like methods is determined by the minimal polynomial on the set of eigenvalues of $K$, i.e. for the $k$-th residual one has

$$\|r^k\| \leq \min_{\pi \in \mathcal{P}_k, \pi(0)=1} \max_{\lambda \in \sigma(K)} \|\pi(\lambda)\| \|r^0\|, \tag{3.15}$$

[3] Furthermore, in general it should be avoided to apply the cg method to the normal equations with system matrix $K^T K$, since this effectively squares the condition number.

where $\mathcal{P}_k$ denotes the space of polynomials with degree $k$ and $\sigma(K)$ denotes the spectrum of $K$. Fast convergence is achieved if the eigenvalues are sufficiently clustered away from the origin. In the case of the cg method, the analogous estimate for the error with respect to the energy norm is further estimated using Chebyshev polynomials of the first kind to obtain

$$\|e^k\|_K \leq 2\left(\frac{\sqrt{\kappa(K)}-1}{\sqrt{\kappa(K)}+1}\right)^k \|e^0\|_K, \quad \kappa(K) = \frac{\lambda_{\max}(K)}{\lambda_{\min}(K)}. \tag{3.16}$$

In the indefinite case, explicit solutions to the min-max problem on the right hand side of (3.15) are not easily derived. However, in [22] an example for a special case is given which illustrates the challenge an indefinite problem may pose to a Krylov solver. To this end, let us assume that the eigenvalues of $K$ are given by $\lambda_{\min} \leq \ldots \leq \lambda_i < 0 < \lambda_{i+1} \leq \ldots \leq \lambda_{\max}$, and furthermore that $|\lambda_{\min}| = \lambda_{\max} = 1$ and $|\lambda_i| = \lambda_{i+1}$. Then, following the argument in [22], one obtains

$$\min_{\pi\in\mathcal{P}_k,\pi(0)=1} \max_{\lambda\in\sigma(K)} \|\pi(\lambda)\| \leq 2\left(\frac{\lambda_{i+1}^{-1}-1}{\lambda_{i+1}^{-1}-1}\right)^{[k/2]}. \tag{3.17}$$

Now it is interesting to note that the right hand side of (3.17) corresponds to the estimate (3.16) in step $[k/2]$ for a positive definite matrix $K$ with $\sigma(K) \subset (\lambda_{i+1}^2, 1)$, i.e. a condition number of $\lambda_{i+1}^{-2}$. Noting that the indefinite matrix by assumption had a condition number of $\lambda_{i+1}^{-1}$, we see that the derived bound implies twice as many steps for an indefinite matrix with condition number $\kappa$ as would be required for a positive definite symmetric matrix with a condition number of only $\sqrt{\kappa}$. This shows the large increase in computational effort needed to solve indefinite systems and furthermore emphasizes the need for preconditioning.

### 3.1.3 Preconditioning

The aim of (left[4]) preconditioning is to find a matrix (or a linear process) $B$, the so-called preconditioner, such that the transformed system

$$B^{-1}Kx = B^{-1}b \tag{3.18}$$

has more favorable spectral properties than the original system, e.g. a lower condition number or a stronger clustering of eigenvalues. The convergence estimate for Krylov methods (3.15) equally applies to the preconditioned system (3.18) and thus the convergence depends on $\sigma(B^{-1}K)$ instead of $\sigma(K)$. Finding a suitable preconditioner $B$ involves the in general conflicting objectives

[4]We confine our presentation to left preconditioning and refer to e.g. [19] for the alternative approaches of right or symmetric preconditioning.

- $B^{-1}$ is a close approximation to $K^{-1}$, and
- the application of $B^{-1}$ to a vector can be computed efficiently.

Most preconditioners for saddle point systems build on techniques related to the solution methods discussed in Sections 3.1.1 and 3.1.2. For example, direct methods can give rise to preconditioners in the form of inexact factorizations. In addition, different block factorizations of the KKT matrix, for instance (3.1), are utilized to derive e.g. certain block triangular matrices to be used as preconditioners. The segregated methods also give rise to block preconditioners. The performance of such block preconditioners chiefly depends on the availability of efficient solution methods for the individual blocks. While efficient methods for the constraint-related blocks are most likely available from a large body of PDE solvers, this is different for the Schur complement and the reduced Hessian. Since both are virtually never are available explicitly in matrix form, suitable approximations have to be constructed. This can lead to elaborate schemes of inner-outer iteration techniques. In the following, we focus on constraint preconditioning and refer to [22] for pointers to other preconditioning approaches.

**Constraint Preconditioning** Consider the system (2.54) and a preconditioner

$$B_{CP} = \begin{bmatrix} \hat{H} & C^T \\ C & 0 \end{bmatrix}, \tag{3.19}$$

where $\hat{H}$ is a symmetric matrix in $\mathbb{R}^{2N\times 2N}$. The matrix $B_{CP}$ can be interpreted as the saddle point system resulting from a minimization problem with a different quadratic energy but the same constraints, which explains the expression *constraint preconditioner* [102, 116]. Since $B_{CP}$ again is an indefinite matrix, we note that (3.19) can not be used with all iterative methods, for instance MINRES requires a positive definite preconditioner. Also, at first sight it is not evident why solving systems with $B_{CP}$ should be more efficient than solving (2.54). However, there a few desirable properties which make it worthwhile to have a closer look at constraint preconditioning. Using an orthogonal factorization of $C^T$ involving a null space basis $Z$ and a subsequent similarity transformation in [102] it is shown that the preconditioned matrix $B_{CP}^{-1}K$ is similar to

$$\tilde{B}_{CP}^{-1}\tilde{K} = \begin{bmatrix} I & & \\ \star & (Z^T\hat{H}Z)^{-1}H_Z & \\ \star & \star & I \end{bmatrix}. \tag{3.20}$$

Here, $I$ is the $N \times N$ identity, the symbol $\star$ denotes nonzero $N \times N$ blocks whose specific structure is of no importance for the analysis and the matrix $Z \in \mathbb{R}^{2N\times N}$ is a basis for the null space of the constraints. Furthermore, recall from Section 3.1.1 that $H_Z \in \mathbb{R}^{N\times N}$ is the reduced Hessian. From (3.20) we conclude

**Theorem 3.1** (Eigenvalues of $B_{CP}^{-1}K$)**.** *Let $K$ be a saddle point matrix as given in* (2.54)*, let $B_{CP}$ be a constraint preconditioner as specified in* (3.19) *and let $Z \in \mathbb{R}^{2N\times N}$ be a basis for the null space of the constraints $C$. Then the preconditioned matrix $B_{CP}^{-1}K$ has*

*i) an eigenvalue 1 with multiplicity $2N$, and*

*ii) $N$ eigenvalues $\lambda$ defined by the generalized eigenvalue problem*

$$H_Z w = \lambda Z^T \hat{H} Z w. \tag{3.21}$$

A further consequence for the iteration behavior of Krylov methods is given by

**Theorem 3.2** (Krylov subspace dimension for $B_{CP}^{-1}K$)**.** *In addition to the above assumptions, let $\hat{H}$ be such that $Z^T\hat{H}Z$ is positive definite symmetric and assume that $(Z^T\hat{H}Z)^{-1}H_Z$ has $k$ distinct eigenvalues $\lambda_i$, $1 \le i \le k$, $1 \le k \le N$. Then the dimension of the Krylov subspace $\mathcal{K}(B_{CP}^{-1}K, b)$ is at most $k+2$.*

Proofs for both Theorems can be found in [102]. The important consequence of these assertions is the fact that an optimal Krylov method such as GMRES applied to the preconditioned system will terminate after at most $k+2$ steps. Fast convergence can be expected if the approximation $Z^T\hat{H}Z$ to the reduced Hessian $H_Z$ leads to a good clustering of the eigenvalues of $(Z^T\hat{H}Z)^{-1}H_Z$. In the extremal case that $Z^T\hat{H}Z = H_Z$, convergence occurs within three iterations. Motivated by the desirable properties, several different implementations have surfaced recently, see e.g. [15, 136]. A concrete example and numerical results will be given in Section 3.2 and 3.3.

*Remark* 3.3. The close relation between constraint preconditioning and the null space method has been exploited in [52, 74, 136]. In [74] it was shown that the solution of (2.54) with the preconditioned CG method, where the preconditioner is of the form (3.19), generates iterates within the null space of the constraints, provided the initial guess is feasible. Thus, the resulting iterative process is closely related to the solution of (3.8) within the null space method by conjugate gradient. This provides an explanation for the surprising fact that the cg method can be applied directly to the indefinite system (2.54). We emphasize that this is a special property of a constraint preconditioner. Furthermore, all subsystem solutions have to be computed with high accuracy, quickly rendering this approach costly.

At the end of this section, let us remark that in many practical situations, the line between segregated and all-at-once approaches tends to be blurred: inexact variants of the segregated methods might be employed as preconditioners in an all-at-once approach [66, 67]. *Partial elimination* has been applied in [34] to the linear-quadratic model problem. The resulting system then resembles a system for the biharmonic equation, and algebraic multigrid was employed. A symmetric indefinite preconditioner was proposed recently in [140] for the model problem ($\mathrm{LQP}_h$). The method

appears to be robust and converges mesh independent, however, as it uses special problem-dependent inner products it is not obvious how to generalize the approach to different problems.

## 3.2 A Block-Triangular Constraint Preconditioner Based on a Reduced Hessian Approximation

In this section we present an implementation of a specific constraint preconditioner (3.19) along with numerical results. The general principle discussed in this section will serve as a building block for a smoothing iteration within a multigrid method for KKT systems, cf. Section 4.3.

We use (3.10) to define $Z$. We set

$$\hat{H} = \begin{bmatrix} 0 & 0 \\ 0 & \hat{H}_Z \end{bmatrix}, \tag{3.22}$$

which yields

$$B_{CP} = \begin{bmatrix} & & L_h^T \\ & \hat{H}_Z & -M_h \\ L_h & -M_h & \end{bmatrix}. \tag{3.23}$$

Since $Z^T \hat{H} Z = \hat{H}_Z$ and due to (3.21), we conclude that $\hat{H}_Z$ should be some approximation to the reduced Hessian $H_Z$. Due to Theorems 3.2 and 3.1, the convergence of a Krylov method preconditioned with $B_{CP}$ defined by (3.23) depends on $\hat{H}_Z$. Useful expressions for $\hat{H}_Z$ will be given below.

A single application of $B_{CP}^{-1}$ to a vector $x_h = (x_h^y, x_h^u, x_h^p)$ is obtained by solving $B_{CP}\tilde{x}_h = x_h$. Due to the lower block-triangular structure, $B_{CP}$ can be inverted in three steps by blockwise forward substitution, see Algorithm 1. For steps 1 and 3 of

$\tilde{x}_h = B_{CP}^{-1} x_h$

| | | |
|---|---|---|
| 1: | $\tilde{x}_h^p \leftarrow L_h^T \tilde{x}_h^p = x_h^y$ | |
| 2: | $\tilde{x}_h^u \leftarrow \hat{H}_Z \tilde{x}_h^u = x_h^u + M_h \tilde{x}_h^p$ | (RHE) |
| 3: | $\tilde{x}_h^y \leftarrow L_h \tilde{x}_h^y = x_h^p + M_h \tilde{x}_h^u$ | |

Algorithm 1: One application of the constraint preconditioner $B_{CP}^{-1}$ defined by (3.23) to a vector $x_h = (x_h^y, x_h^u, x_h^p)$.

Algorithm 1 we assume that efficient solution methods for the subsystems with $L_h$ and $L_h^T$ are available. Recall that $L_h$ and $L_h^T$ represent discretized differential operators and

in principle any method appropriate for the solution of the state and adjoint equation could be employed here.

The crucial issue in Algorithm 1 is the second step, labeled with (RHE), which comprises the solution of a system with the coefficient matrix given by the reduced Hessian approximation $\hat{H}_Z$. Due to (3.12), obviously it is not possible to explicitly assemble the $N \times N$ matrix $H_Z$ and define $\hat{H}_Z$ based on, e.g. an incomplete factorization. We opt for the iterative approach and to this end define the matrix-vector product (MatVec) with $H_Z$ according to (3.12). The necessary steps which implement

$$\check{x}_h^u = H_Z x_h^u = (M_h L_h^{-T} M_h L_h^{-1} M_h + \sigma M_h) x_h^u$$

1: $v_h \leftarrow L_h v_h = M_h x_h^u$
2: $w_h \leftarrow L_h^T w_h = M_h v_h$
3: $\check{x}_h^u \leftarrow M_h w_h + \sigma M_h x_h^u$

Algorithm 2: Definition of MatVec $H_Z x_h^u$ given by (3.12). MatVec is employed within an iterative solution method in step 2 of Algorithm 1.

the MatVec are stated in Algorithm 2. Steps 1 and 2 in Algorithm 2 again require the solution of the forward and adjoint problems with coefficient matrices $L_h$ and $L_h^T$, respectively. Here it is natural to employ the solver which is already used in steps 1 and 3 of Algorithm 1.

Since $H_Z$ is positive definite, we employ the CG method. The approximation accuracy of $\hat{H}_Z$ to $H_Z$ is then determined by the accuracy of the cg method, i.e. the stopping criterion which is chosen to control the iterations. In the extremal case that the cg method is driven to full convergence, we expect to approximate $H_Z$ proportional to the numerical round-off error. From Theorem 3.2 we then conclude that a Krylov method such as GMRES preconditioned with $B_{CP}$ (in exact arithmetic) converges in three steps.

## 3.3 Numerical Results

In this section we present numerical results which have been obtained with our implementation of the preconditioner (3.23) using algorithms 1 and 2 applied within a GMRES method. The implementation uses the PETSc (Portable, Extensible Toolkit for Scientific Computation) framework [16–18], which in particular provides the necessary linear algebra data structures such as vectors and matrices, as well as a library of Krylov methods. The inner BLAS (Basic Linear Algebra Subroutines) kernel is provided by the GOTO BLAS library [72] which is an architecture-optimized implementation of BLAS.

We consider the linear-quadratic model problem ($\mathrm{LQP}_h$) as test problem. Due to linearity, it is sufficient to consider a zero right hand side and a suitably general initial guess $x_h^0$. To be precise, we provide the initial guess $x_h^0$ as a vector with random components $x_i \in [-1, 1]$. The exact solution $x_h^*$ of ($\mathrm{LQP}_h$) is identically zero and, denoting the iteration counter of the outer GMRES method with $m$, the error is given by the current iterate $x_h^m$. In order to compare errors for different discretization parameters $h$, we introduce the discrete $L^2$-norm and define

$$e_{u_h}^m = \|u_h^m - u_h^*\|_{L^2,h} = h\Big(\sum_{T_i \in \mathcal{T}_h} |u_{h,i}^m - u_h^*(z_i)|^2\Big)^{1/2}, \tag{3.24}$$

as the discrete $L^2$-error with respect to the control component $u_h$. In (3.24), $z_i$ denotes the center of the grid cell $T_i$. In the same manner we define the error with respect to the state and adjoint components $y_h$ and $p_h$ which correspondingly is denoted by $e_{y_h}^m$ and $e_{y_h}^m$, respectively. We define the total discrete error in the $m$-th iteration by

$$e_h^m = ((e_{y_h}^m)^2 + (e_{u_h}^m)^2 + (e_{p_h}^m)^2)^{1/2}. \tag{3.25}$$

For the following experiments, the solution of linear systems with coefficient matrix $L_h$ or $L_h^T$ is obtained using a direct factorization method. Here we use the PETSc built-in Cholesky decomposition since $L_h$ is symmetric positive definite. Let us point out that for large-scale problems, one of the sparse factorization methods discussed in Section 3.1.2 or an iterative method should be employed. Since our main objective is to develop a multigrid method, we do not pursue these issues. In our experiments, the factorization is performed once in the initialization phase, and the solutions are computed using the corresponding backsubstitutions with the triangular factors. These are required in steps 1 and 3 of Algorithm 1. The inversion of $\hat{H}_Z$, i.e. the solution of the reduced Hessian equation (RHE) required in step 2 of Algorithm 1, is computed with the CG method using (3.12) and MatVec given by Algorithm 2. The approximation quality of $\hat{H}_Z$ to $H_Z$ is governed by two stages: the accuracy of the MatVec for $H_Z$ and the accuracy of the CG solution. For the following experiments we assume that the accuracy of MatVec is of the order of round-off error, in particular, the same triangular backsubstitutions as in steps 1 and 3 of Algorithm 1 are used for solves with $L_h, L_h^T$ in Algorithm 2. Let $k$ denote the iteration count of the inner CG method. The first experiment is conducted such that $\hat{H}_Z \approx H_Z$ as accurate as possible. To this end, we stop the CG iterations if the *inner residual*

$$r_{cg}^k = b_h^m - \hat{H}_Z x_h^{u^m,k} \tag{3.26}$$

with respect to the right hand side $b_{cg}^{u,m} = x_h^{u^m} + M_h x_h^{\tilde{p}^m}$ in (RHE) satisfies

$$\|r_{cg}^k\|_2 \leq \varepsilon_{\mathrm{mach}}. \tag{3.27}$$

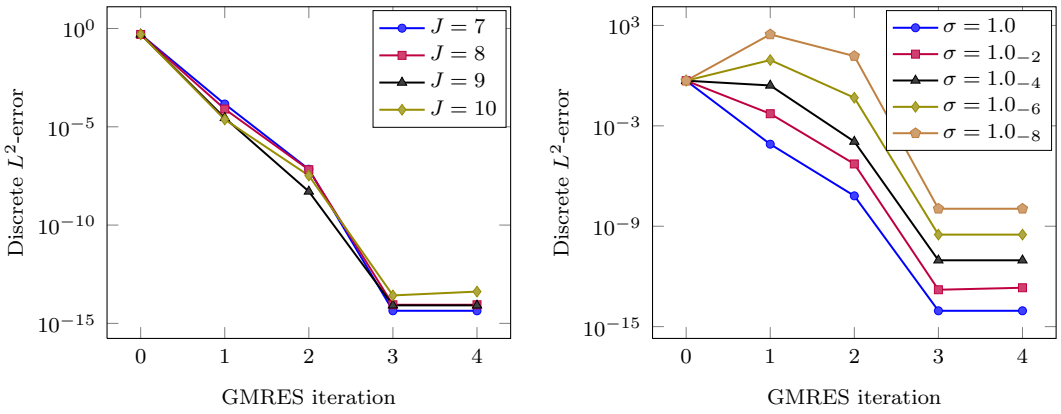



Figure 3.1: Convergence of $B_{CP}$-GMRES with $\hat{H}_Z \approx H_Z$ and $L_h, L_h^T$ inverted by direct factorization. Varying mesh size $h = 2^{-J}$, $J = 7, 8, 9, 10$, fixed regularization parameter $\sigma = 1.0$ (left) and fixed mesh size $h = 2^{-8}$, varying regularization parameter $\sigma$ (right).

Here, $\varepsilon_{\text{mach}} \sim 1.1_{-16}$ is the relative precision (the machine epsilon) for double precision floating point arithmetic. The outer GMRES iteration is stopped based on the *outer residual*

$$r^m = b_h - K_h x_h^m, \tag{3.28}$$

according to

$$\|r^m\|_2 \leq \max(\varepsilon_{\text{mach}}, \varepsilon_{\text{mach}} \|b_h\|_2). \tag{3.29}$$

Let us remark that the tight tolerance in (3.29) is chosen exclusively for testing purposes. A sensible stopping criterion in PDE-related applications should take into account the order of the discretization error, $\mathcal{O}(h^2)$ in the present case.

Figure 3.1 shows the reduction of $e_h$ vs. the number of outer GMRES iterations. In agreement with Theorem 3.2 we see that convergence is achieved within 3 iterations. The number of GMRES iterations is independent of the mesh size $h$ as well as of the regularization parameter $\sigma$ (but see further comments below), which is apparent from Figure 3.1 left and right, respectively. Figure 3.2 shows the error reduction with respect to the individual error components $e_h^y, e_h^u$ and $e_h^p$. Here, the fixed mesh size $h = 2^{-8}$ is chosen, the regularization parameter is $\sigma = 1.0_{-8}$. Hence, the plot on the left in Figure 3.2 shows the errors $e_h^y, e_h^u$ and $e_h^p$ corresponding to the case —●— on the right side of Figure 3.1. Recall that the initial guess $x_h^0$ was chosen as a random vector and thus is infeasible with respect to the constraints $C$. In contrast, for the results shown on the right of Figure 3.2 a feasible $x_h^0$ was chosen, i.e. $L_h x_h^{y,0} = M_h x_h^{u,0}$ holds. As we see, convergence occurs in one iteration less for the case of a feasible $x_h^0$ as compared to an infeasible $x_h^0$. The convergence behavior shown in Figure 3.2 reflects the fact that the solution of the $H_Z$-equation yields a descent direction with respect to the reduced functional $\hat{\mathcal{J}}$ under the condition that $x_h^0$ is feasible. This

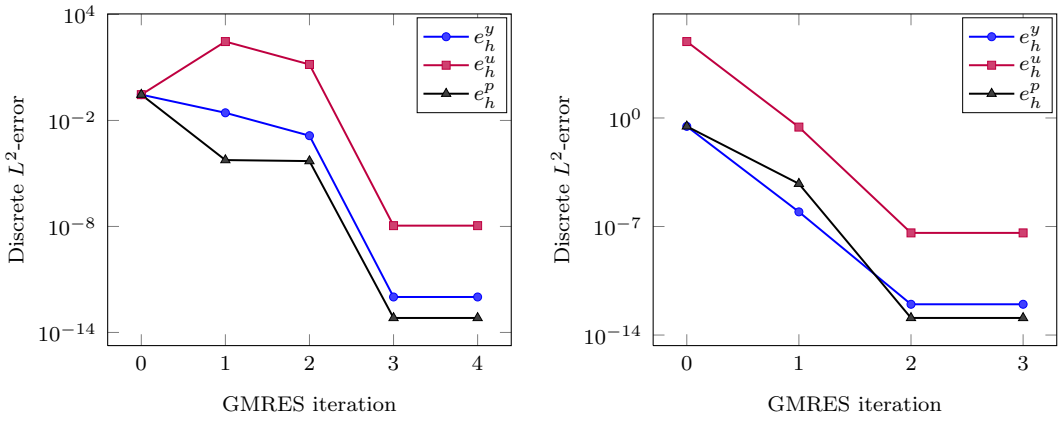



Figure 3.2: Error reduction with respect to the individual components $e_h^y, e_h^u, e_h^p$. Fixed mesh size $h = 2^{-8}$, initial guess infeasible with respect to $C$ (left) and feasible (right).

issue is fundamental for the convergence of reduced space methods, feasible methods in general require a feasible initial guess and a preliminary step is required to obtain $x_h^0$. This is not the case if a reduced space method is only used as preconditioner. However, in that case, the solution of (RHE) is not necessarily a descent direction and thus an increase of $e_h^u$ can occur. Since due to the projection onto the null space of $C$ all later iterates are feasible, here this happens only in the first iteration. Both the magnitude of the increase in $e_h^u$ as well as the size of the initial error $e_h^u$ in the case of a feasible $x_h^0$ depend on the value of $\sigma$. This is due to the optimality condition (2.42) which enforces a different scaling of magnitude $\mathcal{O}(\sigma)$ between $u_h$ and $p_h$. This scaling is clearly not respected by an arbitrary initial guess. Note that for practical problems a useful initial guess for the control $u_h$ might be available. The construction of a feasible $x_h^0$ then requires the solution of the constraint PDE prior to the solution of the optimization problem. Note further that for all experiments in this section we plot the actual error and not the residual norm. Although GMRES exhibits a monotonic decrease of the residual norm, for ill-conditioned systems the corresponding error can nevertheless increase, as happens here for $\sigma \ll 1$. Error-minimizing Krylov methods do exist but are seldom used in practice due to relatively high computational costs.

R.Weiss, SISC,src

Although the number of outer GMRES iterations does not depend on the mesh size $h$ or the regularization parameter $\sigma$, the complete approach is far from having these properties. For each application of $B_{CP}$, two systems of size $N \times N$ need to be solved, one each with the coefficient matrix $L_h$ and $L_h^T$. Furthermore, the same amount of work is required for each inner iteration of the CG method. Thus, the total cost $C_{B_{CP}\text{-GMRES}}$ for the $B_{CP}$-GMRES method can roughly be given as

$$C_{B_{CP}\text{-GMRES}} = \sum_{m=1}^{\bar{m}} (2 + \bar{k}^m) C_N, \tag{3.30}$$

Table 3.1: Iteration numbers $\bar{k}^m$ for the inner CG method at outer GMRES iteration $m$. The mesh size is fixed as $h = 2^{-8}$, the regularization parameter $\sigma$ varies.

| | $\sigma$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $m$ | $1.0$ | $1_{-1}$ | $1_{-2}$ | $1_{-3}$ | $1_{-4}$ | $1_{-5}$ | $1_{-6}$ | $1_{-7}$ | $1_{-8}$ |
| 1 | 6 | 7 | 9 | 13 | 22 | 43 | 100 | 267 | 772 |
| 2 | 6 | 7 | 9 | 13 | 22 | 44 | 101 | 269 | 765 |
| 3 | 6 | 8 | 11 | 15 | 26 | 52 | 119 | 309 | 873 |
| $\sum$ | 18 | 22 | 29 | 41 | 70 | 139 | 320 | 845 | 2410 |

where $C_N$ denotes the cost for solves with $L_h, L_h^T$, $\bar{m}$ denotes the number of outer GMRES iterations required to satisfy (3.29) and $\bar{k}^m$ denotes the number of iterations the CG iteration requires to satisfy (3.27). In order to obtain a robust and mesh independent solution method with optimal complexity $\mathcal{O}(n)$ based on $B_{CP-GMRES}$ the following requirements need to be satisfied:

- the number of outer GMRES iterations $\bar{m}$ is mesh-independent and robust w.r.t. $\sigma$,
- the number of inner CG iterations $\bar{k}^m$ is mesh-independent and robust w.r.t. $\sigma$ for all $m$ and,
- the cost $C_N$ is $\mathcal{O}(N)$.

The first requirement is satisfied according to Theorem 3.2 if the eigenvalues of (3.21) are independent of $h$, e.g. by $\hat{H}_Z \approx H_Z$ to numerical round-off. The last requirement can be satisfied by using an optimal solver such as full multigrid for the constraint subproblems. Satisfying the second requirement poses the main challenge. Note that the iteration count $\bar{k}^m$ is not affected by the specific solution method for the constraints. Indeed, this is one major drawback for reduced space methods, as was discussed in Section 3.1.1. For the model problem ($\mathrm{LQP}_h$), it turns out that $\bar{k}^m$ almost exclusively depends on the value of $\sigma$. Table 3.1 shows the value $\bar{k}^m$ for $m = 1, 2, 3$ and different values of $\sigma$. The mesh size is chosen as $h = 2^{-8}$. The large iteration numbers demonstrate that $B_{CP}$-GMRES as it stands is not a very competitive method. The iteration numbers indicate that the condition numbers of $K_h$ with respect to $\sigma$ as given in Table 2.1 are mirrored by $H_Z$ (for a detailed discussion of the spectral properties of $H_Z$ we refer to Section 4.6). Note that even for moderate values the constant of proportionality with respect to $C_N$ can become fairly large. For $\sigma = 1.0_{-4}$ the cost $C_{B_{CP}\text{-GMRES}}$ for the solution of the optimal control problem is 146 times the cost $C_N$ which is required for the solution of the underlying PDE.

The presented results make it clear that preconditioning $H_Z$ would be desirable in order to reduce the large number of inner iterations. Unfortunately, preconditioning

an unassembled operator is not easily done. Few publications exist which specifically address this issue. A notable exception is [125], where preconditioning strategies for $H_Z$ in unassembled form are suggested. In particular, a preconditioner $B_H$ with

$$B_{H_Z}^{-1} = W^T \tilde{H}^{-1} W \tag{3.31}$$

is proposed, where $\tilde{H} \approx H$ and $W^T Z = I$, i.e. $W^T$ is a left inverse for the null space basis $Z$. In the case of the LQMP it is possible to chose $\tilde{H} = H$ which then obviously is the best choice. For elimination of variables via a null space basis, the left inverse $W = [0\ I]$ is advocated in [125]. However, in this case preconditioning just amounts to a diagonal scaling $B_{H_Z}^{-1} = \frac{1}{\sigma} M_h^{-1}$ with little or no[5] effect on the CG iteration. More elaborate preconditioners are based on power series expansions to approximate $H_Z^{-1}$, however, the involved computations quickly become expensive. In any case, most publications focus on alleviating ill-conditioning which results from a particular choice of $Z$. In the present situation, where $Z$ is built from a solver of a forward problem, it is justified to assume that techniques to bound $\kappa(Z)$ independent of $h$ are readily available, such as multigrid preconditioners. Here, the main source of ill-conditioning originates from $H$ for $\sigma \to 0$.

Instead of preconditioning, we follow a different approach by using an approximation $\hat{H}_Z$ which is cheaper to evaluate. We study two cases: first, we obtain $\hat{H}_Z$ by relaxing (3.27), now instead we stop the iterations as soon as

$$\|r_{cg}^k\|_2 \leq \max(\varepsilon_{\text{mach}}, \varepsilon \|b_{cg}^{u,m}\|), \tag{3.32}$$

with $\varepsilon = 0.5$. The resulting preconditioner is denoted by $B_{CP_1}$. In the second case, we simply replace $H_Z$ with the preconditioner $B_{H_Z}$ from (3.31) and denote the resulting variant by $B_{CP_2}$. Note that for the discretization of Section 2.3.2 $M_h$ is diagonal and therefore the evaluation of $B_{H_Z}^{-1}$ is cheap. In the general case that $M_h$ is not diagonal, e.g. for finite element discretizations, two strategies are possible: the first is to approximate $M_h$ by its lumped version, the second is to employ a few steps of the CG method. Since $M_h$ should be positive definite and well-conditioned, the CG method converges rapidly. Figures 3.4 and 3.3 show the convergence histories for the outer GMRES iteration preconditioned with $B_{CP_1}$ and $B_{CP_2}$, respectively. In both Figures, on the left the fixed regularization parameter $\sigma = 1.0_{-6}$ was chosen and the mesh size $h = 2^{-J}$ varies, with $J = 7, 8, 9, 10$. On the right side in both Figures, results for a fixed mesh size $h = 2^{-8}$ and varying regularization parameter $\sigma$ are shown. First, we see that robustness with respect to $\sigma$ is lost in both cases, as could be expected when going from $H_Z$ to $\hat{H}_Z$. However, this loss is considerably more sever for the case of $\hat{H}_Z = B_{H_Z}$, i.e. $B_{CP_2}$. In particular, note the different scaling of the x-axis in both Figures, which gives the number of outer GMRES iterations. For moderate values of $\sigma$, e.g. down to $\sigma = 1.0_{-4}$, both methods perform nearly equal. This is

[5] For the model problem ($\text{LQP}_h$) we have $M_h = h^2$ and thus $B_{H_Z}$ is constant.

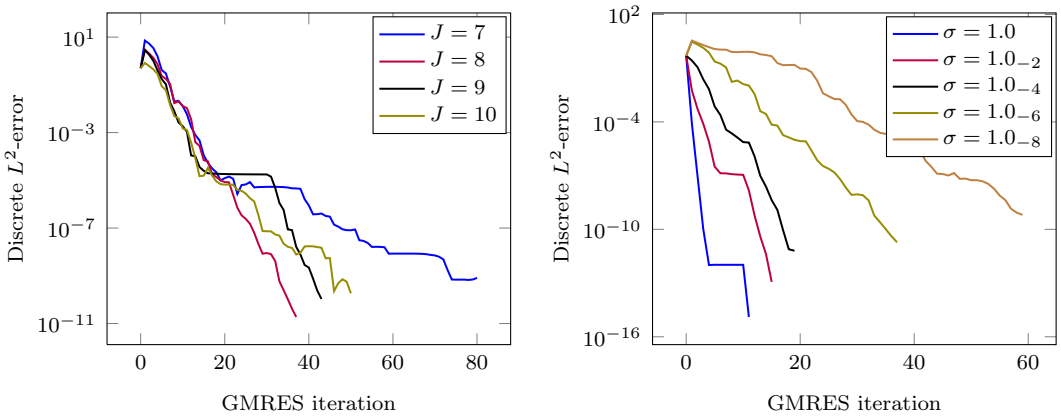



Figure 3.3: Error reduction of $B_{CP_1}$-preconditioned GMRES, i.e. $\hat{H}_Z = H_Z$ with the relaxed stopping criterion (3.32). Different mesh sizes $h = 2^{-J}$ and fixed regularization parameter $\sigma = 1.0_{-6}$ (left), fixed mesh size $h = 2^{-8}$ and varying regularization parameter (right).

explained by recalling expression (3.12), from which we gather that $H_Z$ can be seen as a perturbation of a (scaled) identity for large enough $\sigma$.[6] For $\sigma \to 0$ the behavior changes in favor of the second approach, i.e. the perturbation terms gain influence and should not be omitted anymore. Recalling that, as before, solutions of systems with $L_h, L_h^T$ still are obtained by a direct method, it comes as no surprise that mesh independence is retained to a large extent for both approaches, more so for the first approach, since $B_{H_Z}$ is conditioned independently of $h$ and to a lesser extent for the second approach, where $\hat{H}_Z$ introduces a mild dependence on $h$.

For the second test case, the outer GMRES method has to be a flexible variant [138] since the inner CG iterations do not constitute a stationary linear process. Furthermore, since the number of outer iterations has increased significantly, in order to reduce computational costs a restarted version of GMRES is employed. Here, we use a restart value of 10, i.e. the dimension of the Krylov subspace is limited to 10. Note that the restarted version of GMRES can not retain the monotonic decrease even in the residuals. In particular, eigenvalues close to the origin pose potential trouble for restarted versions. Interestingly enough, numerical experiments showed that larger restart values could lead to worse convergence behavior and very quickly the residual norm would stagnate until the next restart occurred. A similar behavior has also been observed in the context of discretized convection-diffusion problems [62].

With respect to the preconditioner $B_{CP_1}$, the total number of the inner conjugate gradient iterations, $\sum_{m=1}^{\bar{m}} \bar{k}^m$, is a quantity of interest in order to estimate the computational cost. This quantity is given in Table 3.2 for fixed $h = 2^{-8}$, i.e. corresponding to the results shown on the right of Figure 3.3. As expected, the total iteration count

[6] A more detailed analysis of $H_Z$ will be given in Sections 4.3 and 4.5.5

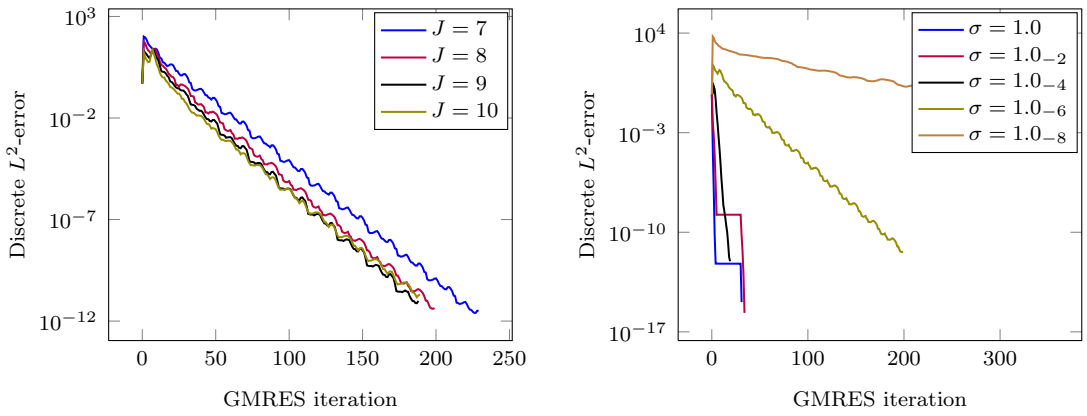



Figure 3.4: Error reduction of $B_{CP_2}$-preconditioned GMRES(10), i.e. $\hat{H}_Z = B_{H_Z}$. Different mesh sizes $h = 2^{-J}$ and fixed regularization parameter $\sigma = 1.0_{-6}$ (left), fixed mesh size $h = 2^{-8}$ and varying regularization parameter (right).

Table 3.2: Total number of inner iterations $\bar{k}^m$ for the relaxed CG method within $B_{CP_1}$. The mesh size is fixed as $h = 2^{-8}$, the regularization parameter $\sigma$ varies.

| $\sigma$ | $1$ | $1_{-1}$ | $1_{-2}$ | $1_{-3}$ | $1_{-4}$ | $1_{-5}$ | $1_{-6}$ | $1_{-7}$ | $1_{-8}$ |
|---|---|---|---|---|---|---|---|---|---|
| $\sum_{m=1}^{\bar{m}} \bar{k}^m$ | 11 | 13 | 15 | 24 | 49 | 129 | 254 | 482 | 1370 |

is lower than in the first experiment, however again for $\sigma \to 0$ unacceptably large iteration numbers result, in particular in view of the fact that forward problems are still solved exactly at each outer GMRES and within each inner CG iteration. Thus, both approaches are still far from being a method which exhibits optimal complexity.

A sizable part of publications concerned with the analysis of nested Krylov methods is based on the assumption of exact arithmetic. Although it is well-known that numerical round-off error due to finite precision arithmetic can slow down convergence and also limit the final attainable accuracy (which is proportional to $\varepsilon_{\text{mach}}\kappa(H_Z)$), cf. [76, 120] for the positive definite and [144] for the indefinite case, for many applications these facts are not a serious limitation. Reasonable stopping criteria related to the discretization accuracy stop the iterations before the attainable accuracy is reached, and many ways of accelerating convergence are available. However, in the context of nested Krylov iterations, these effects appear in a new light as outer iterations might be sensitive to these issues [10, 101]. Often it is advised to take special measures in order to limit the influence of numerical round off, see e.g. [74], where iterative refinement and a particular residual update strategy are proposed within a null-space projected gradient method.

Our preliminary implementation does not incorporate any special measures regarding numerical round-off and thus some effects can be clearly observed in the results

presented in this section, in particular in Figure 3.1, right, but also in all other cases where $\sigma \to 0$. Keep in mind that the stopping criterion (3.29) for the outer GMRES iteration prescribed the same tolerance for all experiments in this section. A particular problematic issue is to employ a reliable stopping criterion for the inner iteration when $\kappa(H_Z)$ is large. Proposed bounds on the forward error $e_{cg}^k$ which are more reliable than (3.32), cf. [11], inevitably require some estimate of $\|H_Z\|^{-1}$, since $e_{cg}^k = H_Z^{-1} r_{cg}$. An estimate of $\lambda_{\min} = 1/\|H_Z^{-1}\|_2$ can be obtained at the cost of a few matrix-vector products utilizing the Lanczos algorithm, however keep in mind that in the present case a matrix-vector product is expensive as it involves repeated solutions of the forward problem.

## Summary

In this chapter we discussed the most widely used numerical algorithms suitable for the solution of saddle point systems arising in PDE constrained optimization. For a particular instance of a constraint preconditioner we presented an implementation and numerical results.

Although the outer GMRES iteration exhibited the favorable properties of robust and mesh-independent convergence, which could be expected based on Theorems 3.1 and 3.2, we associated the following potential drawbacks with this approach: repeated exact solutions of the constraint equations are required at each outer and within each inner iteration, inner iterations for $H_Z$ are not robust with respect to $\sigma$, efficient single-level preconditioning of a non-assembled $H_Z$ is still an open topic, and adjusting and implementing reliable stopping criteria for nested inner-outer Krylov iterations is a non-trivial task.

# 4 A Multigrid Method for the Solution of Linear-Quadratic Optimal Control Problems

The main part of this chapter is devoted to the development and presentation of a coupled multigrid method for the solution of saddle point systems (2.51) arising from the discretization of PDE constrained optimization problems. In subsequent chapters, the devised method will be adapted to handle inequality constraints on the control, and it will be employed for the solution of the systems, which are generated when applying Newton-type methods to optimization problems with nonlinear constraints.

The multigrid method is one of the most efficient solution methods for linear systems arising from discretized second-order elliptic boundary value problems. The algebraic error usually satisfies an estimate of the form

$$\|w_h^{m+1} - w_h\| \leq \varrho \|w_h^m - w_h\| \tag{4.1}$$

with a constant $\varrho \in (0,1)$ which is bounded uniformly in $h$. Here, $\varrho$ is called the asymptotic convergence rate if it is related to the spectral radius of the multigrid iteration matrix $G_{MG}$, or the contraction number if $\varrho$ results from an estimate of $\|G_{MG}\|$.

For practical applications, it is crucial that $\varrho \ll 1$ holds (often $\varrho < 0.25$ which denotes the so-called "textbook efficiency" of multigrid methods). However, even if this can not be achieved, multigrid methods often prove to be very good preconditioners, resulting in rapid convergence if accelerated by a Krylov method. Reducing the relative algebraic error to a tolerance $\varepsilon$ can be achieved in $m \geq \log \varepsilon / \log \varrho$ iterations. In PDE-related applications, it is reasonable to relate the tolerance $\varepsilon$ to the discretization accuracy, i.e. $\varepsilon = \mathcal{O}(h^\kappa)$ with $\kappa$ being the consistency order of the discretization. Thus, $m = \mathcal{O}(|\log h|)$ iterations and, assuming a cost of $\mathcal{O}(h^{-d})$ per iteration, a total complexity of $\mathcal{O}(h^{-d}|\log h|)$ is obtained when computing a solution with an algebraic error of the same order as the discretization error. Even more efficient is the full multigrid method which, by a combination of multigrid cycles with nested iteration, yields a solver with the optimal complexity $\mathcal{O}(h^{-d})$.

First applications of the two- and multigrid idea to Poisson's equation have been carried out as early as 1961 by Fedorenko and Bachvalov. It was the pioneering work of Brandt [40] and independently Hackbusch [85] that later sparked the success which multigrid methods enjoy today. For details on the multigrid method we refer to the classical monographs [41, 85, 147], among others.

In contrast to other fast solution methods for Poisson's equation (see [47] for a discussion of *cyclic reduction*, the *fast Fourier transform* and the *Buneman algorithm*),

the multigrid method is much more versatile. In fact, the multigrid idea should be seen more as a framework which allows to devise fast solvers for a wide range of problems by adapting the main components to the problem at hand. This flexibility is the key property which motivates us to employ the multigrid paradigm for the construction of an efficient solution algorithm for discrete optimality systems.

Before we introduce our proposed method we will give an overview of state-of-the-art of solution methods for optimization and optimal control problems which employ multigrid ideas in some way. First though let us convey the necessary details of the general multigrid idea using the well-understood case of the Poisson equation as an illustrative example. Furthermore we establish the required notation as well as certain concepts such as the grid hierarchy and the generic algorithm on which we will build our subsequent developments.

## 4.1 Multigrid for Scalar Elliptic Equations

Following the exposition in Section 2.3.2 with the diffusion tensor $D = I$ we obtain a linear system

$$L_h w_h = b_h, \quad w_h \in W_h. \tag{4.2}$$

To simplify the notation, we omit the $\hat{}$-symbol, which in Section 2.3.2 was used to denote quantities defined with respect to the rectangular computational grid. Here, we denote the computational domain by $\Omega$ and the space of grid functions on the rectangular grid $\mathcal{T}_h$ by $W_h$. Furthermore, we do not distinguish between a function $w_h \in W_h$ and its representation as a vector.

The basic observation which led to the development of multigrid methods is that classical stationary iterative processes such as the Jacobi or Gauss-Seidel iteration exhibit a *smoothing property* when employed to solve (4.2). The high frequency components of the error are quickly reduced, however as soon as the error becomes smooth in a sense to be made precise below, further reduction occurs only at a slower rate of $1-\mathcal{O}(h^2)$ in case of Jacobi- and Gauss-Seidel iterations, which means that the solution process effectively stagnates. Let $\tilde{w}_h^m$ denote the current iterate after a few steps of a smoothing iteration applied to (4.2) and let $e_h = w_h^* - \tilde{w}_h^m$ denote the error to the exact solution $w_h^*$ of (4.2). Due to linearity, $e_h$ satisfies the residual equation

$$L_h e_h = r_h, \tag{4.3}$$

where $r_h = b_h - L_h \tilde{w}_h^m$ is the current residual. Since the error $e_h$ is smooth, it can be accurately approximated by a function $e_H$ on a coarser mesh with a mesh size $H > h$. The crucial multigrid idea is to solve the residual equation on this coarser mesh, which is possible with substantially lower computational cost,[1] and use the result as

[1] For standard coarsening, i.e. halving the grid size in each direction, in two dimensions the number of unknowns on the next coarser grid is reduced by a factor of four.

a correction to the fine grid iterate, i.e. compute $w_h^{m+1} = \tilde{w}_h^m + e_H$. This process is called the *coarse grid correction* and we have essentially described a two-grid method. Recursive application of the coarse grid correction with intermediate smoothing steps on each grid level yields the multigrid method. In order to define a multigrid method, we need to specify

- a hierarchy of grids together with suitable interpolation operators between the different grid levels,
- a smoothing iteration to be applied on each grid level,
- a definition for the residual equation, and in particular the differential operator, on each level,
- and a coarse grid solver which is to be employed when recursion ends on the coarsest mesh.

## Smoothing Iterations

Classical smoothing iterations for multigrid methods are given by stationary iterative methods such as the Jacobi- or the Gauss-Seidel iteration. A unified framework is provided by the formulation as a preconditioned Richardson iteration. Let $B_{L_h}$ denote a preconditioner for $L_h$. Then the $B_{L_h}$-preconditioned Richardson iteration applied to (4.2) reads

$$\begin{aligned} w_h^{k+1} &= w_h^k + B_{L_h}^{-1}(b_h - L_h w_h^k) \\ &= (I - B_{L_h}^{-1} L_h) w_h^k + B_{L_h}^{-1} b_h \\ &= G w_h^k + B_{L_h}^{-1} b_h. \end{aligned} \tag{4.4}$$

The operator $G$ is called the iteration matrix. Subtracting the exact solution $w_h^*$ on both sides of (4.4) we obtain

$$e_h^{k+1} = G e_h^k, \tag{4.5}$$

i.e. $G$ is also the error propagation matrix. A solution of (4.2) is a fixed point of the iteration (4.4). The classical Richardson iteration is obtained for $B_{L_h} = \alpha I$ with a relaxation parameter $\alpha > 0$. The Jacobi- and the Gauss-Seidel iteration can be obtained by defining $B_{L_h}$ based on a splitting of $L_h$. Let $L_h = D - E - F$ where $D$ denotes the diagonal part of $K$ and $E, F$ denote the parts below and above the diagonal, respectively. Then the Jacobi iteration is given by (4.4) with $B_{L_h} = D$ and the Gauss-Seidel iteration is obtained for $B_{L_h} = D - E$.

In the following it will be convenient to denote a general smoothing iteration of the form (4.4) by

$$\tilde{w}_h = \mathcal{S}_h^\nu(w_h, b_h), \tag{4.6}$$

where $\nu$ is the number of smoothing steps applied to $w_h$ and $b_h$ denotes the right hand side. The effectiveness of an iteration (4.6) as a smoother is measured by the *smoothing factor* $\mu(\mathcal{S}_h)$. Several different definitions of $\mu(\mathcal{S}_h)$ exist, depending on the context. For now, we refer to Section 4.3, where we will apply local Fourier analysis (LFA) to assess the smoothing property of a smoothing iteration for optimal control problems. In general, $\mu(\mathcal{S}_h)$ measures how fast the high frequency components of the error are reduced. For the classical theory in [85], the smoothing rate is measured by estimates of the form

$$\|L_h S_h^\nu\| \leq \varrho(\nu) h^{-\alpha}, \quad \varrho(\nu) \to 0 \text{ for } \nu \to \infty, \tag{4.7}$$

where $S_h$ is the iteration matrix of the smoother (4.6).

Here we remark that the classical Jacobi iteration does not yield acceptable smoothing rates unless underrelaxation is employed, i.e. $B_{L_h} = \frac{1}{\omega} D$ for some relaxation parameter $0 < \omega < 1$. In particular the smoothing is $h$-dependent and of the same order as the convergence rate, i.e. $1 - \mathcal{O}(h^2)$, if $\omega = 1$. The optimal $h$-independent smoothing rate $\mu(\omega - \text{JAC}) = 3/5$ is obtained for $\omega = 4/5$. On the contrary, although it is well-known that overrelaxation with the optimal $\omega$ improves the Gauss-Seidel convergence rate from $1 - \mathcal{O}(h^2)$ to $1 - \mathcal{O}(h)$, the optimal smoothing rate is obtained for a relaxation parameter close to unity. Therefore, almost always the Gauss-Seidel method without relaxation is employed. In any case, both the convergence and the smoothing rate depend on the ordering of the grid points. For node-based discretizations, red-black Gauss-Seidel (GS-RB) has a better smoothing rate, however for cell-centered discretizations this is not the case anymore [123]. The lexicographical Gauss-Seidel (GS-LEX), which will be used as smoother unless noted otherwise, exhibits the smoothing rate $\mu(\text{GS-LEX}) = 0.5$. Further details on the smoothing properties of classical iterative methods can be found in any multigrid textbook and will not be given here.

## Grid Hierarchy and Interpolation Operators

So far we have considered the linear system (4.2) obtained by discretizing the model problem on a fixed grid with mesh size $h$. Let us now introduce a hierarchy of grids. We denote the different grid levels by the parameter $j$, $j = 0, \ldots, J$, where $j = 0$ stands for the coarsest mesh. The mesh for each level is denoted by $\mathcal{T}_j$ and the associated mesh size is given by $h_j = 2^{-j} h_c$ with $h_0 = h_c$ denoting the coarsest mesh size. For notational simplicity, in the following we will refer to level-dependent quantities by using the level index $j$ as a subscript instead of the mesh size $h_j$.

The standard coarsening process for cell-centered discretizations in two space dimensions is depicted in Figure 4.1. A coarse grid cell $T_i \in \mathcal{T}_{j-1}$ is obtained by merging the four corresponding fine grid cells which we denote with $T_i^s, s = 1, 2, 3, 4$. Clearly the spaces $W_j$, $j = 0, \ldots, J$ defined on each triangulation $\mathcal{T}_j$ are nested,

$$W_0 \subset W_1 \subset \ldots \subset W_J = W_h. \tag{4.8}$$

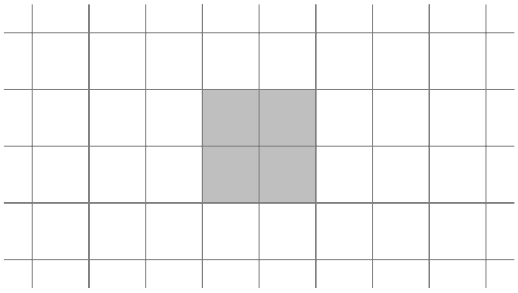


Figure 4.1: Standard coarsening for cell-centered discretizations. A coarse grid cell $T_i \in \mathcal{T}_{j-1}$ is the union of four fine grid cells in $\mathcal{T}_j$.

For each $W_j$ we define the corresponding discrete $L^2$-inner product by

$$(w, v)_{W_j} = h_j^2 \sum_{T_i \in \mathcal{T}_j} w_i v_i, \ w, v \in W_j. \tag{4.9}$$

The coarse grid correction process requires to transfer residuals from a finer level to the next coarser one, and to transfer error functions from a coarser level to the next finer one. This will be achieved by interpolation operators. For second-order differential equations, a suitable interpolation or *prolongation* is given by bilinear interpolation. For cell-centered discretizations, the bilinear prolongation $P^j_{j-1,\mathrm{BL}} : W_{j-1} \to W_j$ is, in the usual stencil notation [147, 154], given by

$$P^j_{j-1,\mathrm{BL}} = \frac{1}{16} \left] \begin{array}{cccc} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{array} \right[, \tag{4.10}$$

where the notation $] \cdot [$ indicates that (4.10) is to be understood in the distributive sense [147]. A natural way to define a *restriction* operator is by way of the adjoint. Let $R^{j-1}_{j,\mathrm{BL}} : W_j \to W_{j-1}$ be defined by $R^{j-1}_{j,\mathrm{BL}} = (P^j_{j-1,\mathrm{BL}})^*$, i.e.

$$(P^j_{j-1,\mathrm{BL}} w_{j-1}, v_j)_{W_j} = (w_{j-1}, R^{j-1}_{j,\mathrm{BL}} v_j)_{W_{j-1}}, \quad v_j \in W_j, \ w_{j-1} \in W_{j-1}. \tag{4.11}$$

Then, in matrix notation we have $R^{j-1}_{j,\mathrm{BL}} = \frac{1}{4}(P^j_{j-1,\mathrm{BL}})^T$. From a practical point of view, also other restriction operators might prove to be beneficial. One particular choice frequently employed for cell-centered discretizations is the *four point average* restriction, defined by the stencil

$$R^{j-1}_{j,\mathrm{FPA}} = \frac{1}{4} \left[ \begin{array}{ccc} 1 & & 1 \\ & \cdot & \\ 1 & & 1 \end{array} \right]. \tag{4.12}$$

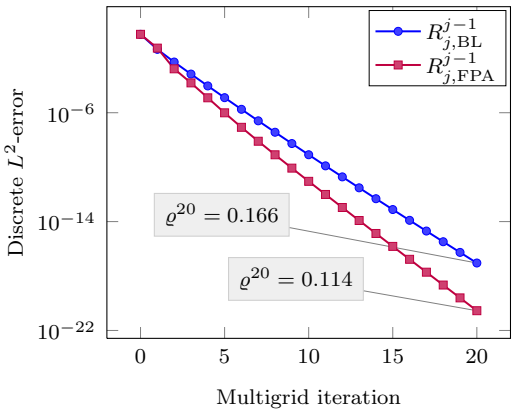



Figure 4.2: Error reduction for a $V_{1,1}$-cycle with GS-LEX smoothing applied to Poisson's equation for $J = 10$ and $h_c = \frac{1}{2}$, comparing four point average and bilinear restriction.

By way of Local Fourier Analysis it has been shown [123] that the pair $P^j_{j-1,\mathrm{BL}}, R^{j-1}_{j,\mathrm{FPA}}$ exhibits smaller reduction factors than the canonical pair $P^j_{j-1,\mathrm{BL}}, R^{j-1}_{j,\mathrm{BL}}$. This is confirmed by numerical experiments, see Figure 4.2. We show the error reduction for a standard $V_{1,1}$-cycle with lexicographical Gauss-Seidel (GS-LEX) smoothing applied to the model problem and $J = 10, h_c = \frac{1}{2}$. Clearly, employing $R^{j-1}_{j,\mathrm{FPA}}$ as restriction operator results in better reduction rates. Furthermore, it should be noted that due to the smaller stencil, the computational cost for one application of $R^{j-1}_{j,\mathrm{FPA}}$ is considerably lower than that for $R^{j-1}_{j,\mathrm{BL}}$ and no modifications at domain boundaries are needed. This implies a very welcome benefit in parallel computations due to additional savings on communication costs (tacitly assuming that domain boundaries are properly aligned within the coarsening process).

With respect to a second-order differential operator, both combinations of $P^j_{j-1,\mathrm{BL}}$ with either $R^{j-1}_{j,\mathrm{FPA}}$ or $R^{j-1}_{j,\mathrm{BL}}$ satisfy the well-known heuristic [85, 154]

$$m_P + m_R > 2m. \tag{4.13}$$

Here, $2m$ is the order of the differential operator, $m_P$ denotes the polynomial order of the interpolation $P^j_{j-1}$, i.e. polynomials of degree $m_P - 1$ are interpolated exactly, and $m_R$ denotes the order of the restriction $R^{j-1}_j$ which is defined as the order of the corresponding adjoint interpolation. The need for condition (4.13) has not only theoretical reasons (in particular it is needed to prove consistency of the coarse grid approximation [85, §6.3.2]) but also has a practical impact which was demonstrated in numerical experiments [90]. The theory in [38] guarantees $h$-independent contraction factors only for the $W$- and the variable $V$-cycle whereas the convergence factor of the $V$-cycle is not uniformly bounded. Presumably this is related to the fact that the natural injection in the cell-centered finite differences case employed in the framework

of [38] violates (4.13). On a final note let us mention that the sparsity pattern of Galerkin coarse grid matrices can only be maintained by specific combinations of restriction and prolongation operators and most canonical pairs satisfying (4.13) lead to an increased stencil size of coarse grid matrices. Further insight in the proper choice of interpolation operators is given for instance in [41, 90].

## Coarse Grid Correction and Coarse Grid Operator

The coarse grid correction step on a given level $j$ involves solving the residual equation (4.3) for the error on the next coarser grid $j-1$, i.e. to solve

$$L_{j-1}e_{j-1} = r_{j-1} = R_j^{j-1} r_j. \tag{4.14}$$

To this end, we need to define the operator $L_{j-1}$ on a coarser grid level $j-1$. There are two different and widely used approaches. The first is to define the coarse grid operator in terms of the fine grid operator via the *Galerkin product*

$$L_{j-1} = R_j^{j-1} L_j P_{j-1}^j. \tag{4.15}$$

This is the standard way in algebraic multigrid methods and most useful if classical geometric coarsening approaches fail. However, both precomputing the coarse grid matrices or applying them in terms of matrix-vector products entails a relatively high computational cost. The approach predominant in geometric multigrid is to use direct discretization, where the operators $L_j,\ j = 0, \ldots, J-1$ are obtained by employing the same discretization method as on the finest level $J$. This second approach is employed in the following.

After solving (4.14), its solution is used to correct the current fine grid iterate $\tilde{w}_j$ via the correction step

$$\tilde{\tilde{w}}_j = \tilde{w}_j + P_{j-1}^j e_{j-1}. \tag{4.16}$$

In compact form, the whole process of coarse grid correction can be written as

$$\begin{aligned} \tilde{\tilde{w}}_j &= \tilde{w}_j + P_{j-1}^j L_{j-1}^{-1} R_j^{j-1} r_j \\ &= (I - P_{j-1}^j L_{j-1}^{-1} R_j^{j-1} L_j)\tilde{w}_j + P_{j-1}^j L_{j-1}^{-1} R_j^{j-1} b_j. \end{aligned} \tag{4.17}$$

Note that, although formally the coarse grid correction (4.17) looks like an iteration of the form (4.4), it can never be a convergent iteration, since the restriction has a non-trivial kernel. Thus, components of the error lying in the kernel of the restriction will not be annihilated by the iteration (4.17). In our situation, however, the null space of the restriction is approximately spanned by the high frequency components of the error which can be effectively removed by the smoothing iteration. In fact, this so-called *complementarity* of the smoothing iteration and the coarse grid correction is a property essential to the fast convergence of multigrid methods.

It remains to define how to solve the coarse grid equation. To this end, the idea of smoothing and coarse grid correction is applied recursively to (4.14) until the coarsest level $j = 0$ is reached. There, equation $L_0 e_0 = r_0$ is solved either by a direct method or with a convergent iterative method. Both approaches are common and depend on the specific problem at hand. In both cases, the crucial idea holds that exact solution of the coarse grid problem is easily affordable since $h_0$ will be relatively large.

## The Multigrid Algorithm

The algorithm described until now consists of *presmoothing* and subsequent coarse grid corrections. In practice, it is beneficial to introduce additional *postsmoothing* steps after the coarse grid correction. We denote the number of pre- and postmooth-

$x_j^{m+1} \leftarrow MG^\gamma(j, b_j, x_j^m)$

1: **if** $j = 0$ **then**
2: $\quad x_0 \leftarrow K_0 x_0 = b_0$ /∗ exact solve on coarsest grid ∗/
3: **else**
4: $\quad \tilde{x}_j = (\mathcal{S}_j)^{\nu_1}(x_j^m, b_j)$ /∗ $\nu_1$ presmoothing steps ∗/
5: $\quad b_{j-1} = \mathcal{R}_j^{j-1}(b_j - K_j \tilde{x}_j)$ /∗ restrict residual ∗/
6: $\quad v_{j-1} = MG^\gamma(j-1, b_{j-1}, 0)$ /∗ grid recursion ∗/
7: $\quad \tilde{\tilde{x}}_j = \tilde{x}_j + \mathcal{P}_{j-1}^j v_{j-1}$ /∗ correction step ∗/
8: $\quad x_j^{m+1} = (\mathcal{S}_j)^{\nu_2}(\tilde{\tilde{x}}_j, b_j)$ /∗ $\nu_2$ postsmoothing steps ∗/

Algorithm 3: The multigrid cycle $MG^\gamma(j, b_j, x_j^m)$ for the solution of $L_j x_j = b_j$.

ing iterations by $\nu_1$ and $\nu_2$, respectively and collect the individual steps in Algorithm 3. Here we have introduced small notational changes such that Algorithm 3 applies to a system $K_h x_h = b_h$ instead of (4.2), i.e. we have set $K_h = L_h$ and $x_h = w_h$. Furthermore, we have set $\mathcal{R}_j^{j-1} = R_{j,\mathrm{FPA}}^{j-1}$ for the restriction operator and $\mathcal{P}_{j-1}^j = P_{j-1,\mathrm{BL}}^j$ for the prolongation. These changes will allow us to apply Algorithm 3 directly to a saddle point system (2.54). All that remains is to specify the interpolation operators $\mathcal{R}_j^{j-1}$, $\mathcal{P}_{j-1}^j$, the smoothing iteration $\mathcal{S}_j^\nu$, the coarse grid solver and the representation of $K_h$ on different grid levels.

The additional parameter $\gamma$ is the cycle index which gives rise to different types of multigrid cycles. The algorithm described so far is obtained by setting $\gamma = 1$ and amounts to the standard $V$-cycle, whereas for $\gamma = 2$ we obtain the $W$-cycle. The $W$-cycle is much more robust but also computationally more expensive than the $V$-cycle. Thus it is in particular well suited for difficult problems. Also convergence theory for the $W$-cycle often is more mature than for $V$-cycles. Often for practical problems the unorthodox $F$-cycle offers a good compromise between cost and robustness. It is

defined recursively by an $F$-cycle followed by a $V$-cycle on the same grid level, [41]. The number of smoothing steps $\nu_1, \nu_2$ are usually chosen as small constants. A level-dependent number of smoothing steps which increases with decreasing $j$ leads to the *variable* $V$-cycle.

The multigrid iteration defined by Algorithm 3 can be written in the compact form of (4.4). For the two-grid method, the iteration matrix is

$$G_{MG} = S_h^{\nu_2}(I - \mathcal{P}_{j-1}^{j} L_H^{-1} \mathcal{R}_j^{j-1} L_h) S_h^{\nu_1}. \tag{4.18}$$

This is the basis for the classical two-grid convergence theory [85], where a norm estimate for (4.18), which does not depend on $h_J$, results from the smoothing property (4.7) and the approximation property

$$\|L_h^{-1} - \mathcal{P}_{j-1}^{j} L_H^{-1} \mathcal{R}_j^{j-1}\| \leq C h^{\alpha}. \tag{4.19}$$

## Full Multigrid

As was briefly mentioned before, with the multigrid algorithm 3 one can compute a solution of the discrete problem with an accuracy up to the order of discretization error with a cost of $\mathcal{O}(h^{-d}|\log h|)$. The combination of nested iteration with multigrid cycles, also called the *full multigrid* (FMG) method, reduces the computational complexity for the same task to the optimal order $\mathcal{O}(h^{-d})$. The idea of nested iteration is to compute initial values for fine grid computations on coarse meshes. To this end, an additional prolongation operator $\tilde{\mathcal{P}}_{j-1}^{j}$ is introduced. Then, the exactly computed solution of the coarse grid equation is prolongated to the next finer level and used as initial value for a small number of multigrid cycles on that level. This is applied recursively until the finest level is reached. The whole process is given in Algorithm 4. The number of multigrid cycles $m_j$ which are employed on each level usually is taken

FMG

1: $\tilde{x}_0 \leftarrow K_0 x_0 = b_0$ /* exact solve on coarsest grid */
2: **for** $j = 1$ to $J$ **do**
3: $\quad x_j^0 \leftarrow \tilde{\mathcal{P}}_{j-1}^{j} \tilde{x}_{j-1}$ /* interpolate initial guess */
4: $\quad$ **for** $m = 0$ to $m_j - 1$ **do**
5: $\quad\quad x_j^{m+1} \leftarrow MG^{\gamma}(j, b_j, x_j^m)$ /* perform $m_j$ multigrid cycles */
6: $\quad\quad \tilde{x}_j = x_j^{m_j}$

Algorithm 4: The full multigrid algorithm given as the combination of nested iteration and multigrid cycles $MG^{\gamma}(j, b_j, x_j^m)$.

as 1 if the basic multigrid cycle is fast converging, e.g. with a rate $\varrho < \frac{1}{4}$. From an

implementation viewpoint one would like to chose the FMG prolongation $\tilde{\mathcal{P}}_{j-1}^{j} = \mathcal{P}_{j-1}^{j}$. For well-behaved problems such as second-order elliptic problems this is indeed a suitable choice. For difficult problems, in particular if FMG is employed within a nonlinear iteration such as Newton's method, it might be required to use a higher order interpolation for $\tilde{\mathcal{P}}_{j-1}^{j}$. Further details on the FMG method can be found in the references given at the beginning of this section.

## 4.2 Multigrid Methods in Optimal Control and Optimization

A natural way to exploit the efficiency of multigrid methods when solving optimal control and optimization problems is to employ multigrid solvers for the solution of the various subproblems arising in the segregated solution approaches discussed in Section 3.1.1. The first realization of this idea to our knowledge was done in [83]. Here, the reduced form (2.6) of the optimal control model problem with equality constraints given by the Poisson equation was considered. The reduced form leads to

$$u_h + C_h u_h = f_h, \tag{4.20}$$

with a compact operator $C_h$. Note that the left hand side corresponds to the reduced Hessian $H_Z$, the regularization parameter $\sigma$ is absorbed by $C_h$. Since (4.20) can be considered a Fredholm equation of the second kind, the multigrid method of [84] was applied to (4.20). This algorithm uses only matrix-vector products to construct the discretization $C_h$, however, each evaluation of $C_h u_h$ involves the solution of two elliptic problems. These have been obtained with a standard multigrid method for elliptic equations. The smoothing step for the outer multigrid iteration is given by one step of Picard's iteration

$$\tilde{u}_h = C_h u_h + f_h. \tag{4.21}$$

The crucial observation here is that the compact part of the reduced Hessian features a smoothing property which is inherited by the discretization $C_h$ (see also the discussion in Section 4.6), therefore the operator $C_h$ itself can be used as a smoother. The Picard iteration converges for $\rho(C_h) < 1$, which essentially yields a condition on the ratio of the regularization parameter $\sigma$ and the coarsest mesh size $h_0$, cf. Section 4.3. Violation of that condition leads to amplification of smooth modes and thus ultimately the multigrid method will diverge.

A related approach is presented in [58], where a reduced SQP algorithm is applied to a shape optimization problem. As was detailed in Section 3.1.1, this involves projections on the null space of the constraints using a null space basis $Z$. In [58] the fundamental basis (3.10) is employed and the application of $L_h^{-1}$ is performed with a standard multigrid method.

As was outlined in Section 3.1.1, a common property of reduced space methods based on a null space projection is the feasibility of all intermediate iterates with respect to the constraints, and it is precisely the feasibility which induces an often unnecessary amount of computational work. Thus, a large potential gain in efficiency is to be expected from coupled methods which solve the constraints only once simultaneous with the solution of the optimization problem.

The construction of a coupled multigrid method is a significant challenge and a major task here is to devise an efficient smoothing strategy for the KKT system. In the last two decades several coupled multigrid approaches for saddle point systems have appeared in the literature, with the bulk of the methods applying to mixed problems or Stokes and Navier-Stokes equations. We mention the Braess-Sarazin smoother [37] as well as collective smoothing strategies such as Box- or Vanka smoothers [41, 151].

Coupled multigrid methods specifically adapted to solve optimization problems are relatively scarce in the literature and have been developed only recently. First ideas have been pitched in [9] under the name of *one-shot methods*. There, a simple gradient-descent scheme is employed as a smoothing iteration. In [58], in addition to the above-mentioned RSQP approach, a transforming smoother[2] yielding an inexact null space iteration was employed for the solution of a topology optimization problem. The same method was applied to a parameter identification problem in groundwater flow [141]. The essential notion resulting in an improved efficiency was to use approximate solutions to the subproblems only instead of the full solutions in [83]. Our presented approach is most closely related to these approaches and differs mainly in the way the reduced Hessian is included. However, no detailed numerical experiments have been performed, our smoothing analysis is novel and the robustness question has not been discussed.

The most comprehensive study of coupled multigrid methods for optimal control to date has bee conducted by Borzi, [33–35]. Following the idea of box-smoothing, the smoothing iteration was chosen as a collective Gauss-Seidel method (CGSM) which amounts to the successive solution of local $3 \times 3$ saddle point problems. The collective smoothing approach was also employed in [14]. For standard elliptic constraints, the method is shown to possess good efficiency and robustness with respect to the regularization parameter. However, the extension to different constraint PDEs is not always obvious. For example, anisotropic operators as constraints might require stronger coupled smoothing procedures such as line smoothers or ILU-smoothing. Both cases are not straightforwardly incorporated into the CGSM smoothing. For standard operators, line smoothing yields tridiagonal systems which can be solved efficiently, e.g. by the Thomas algorithm or even parallel variants. However, the combination of collective and line-smoothing would yield systems of considerable bandwidth which have to be solved exactly at each smoothing iteration. It is unclear how to extend the idea of ILU-smoothing to the CGSM approach. This method is the only coupled multigrid

[2]Transforming smoothers have been developed in [155] for the multigrid solution of flow problems.

method to our knowledge that has been extended to handle additional inequality constraints. Additional inequality constraints with respect to the control are the topic of Chapter 5.

The same author has also developed the first and to date only algebraic multigrid method for optimal control problems [34]. We remark however, that AMG was applied to the partially reduced system with eliminated control, which essentially amounts to a coupled system of two elliptic equations. In this sense, the AMG approach falls in between the segregated and full-space methods. The more general case of AMG solvers for the full optimality system has not been addressed so far.

A different methodology combining optimization and multigrid labeled MG/Opt has been developed in [109, 124]. The key idea is to apply the multigrid concept as an outer method. To this end, a hierarchy

$$\min \mathcal{J}_j(y_j, u_j) \quad \text{s.t. } \mathcal{C}_j(y_j, u_j) = 0, \quad j = 0, \ldots, J \tag{4.22}$$

of minimization problems is constructed. On each level $j$, a standard optimization method such as truncated Newton is employed to approximately solve (4.22). These solutions are embedded in a standard multigrid cycle, where, similar to the Full Approximation Storage (FAS), the solutions are transferred in contrast to residuals and corrections in linear defect correction methods. This concept has been picked up in [75], where a special instance of MG/Opt with a trust-region Newton method as optimizer on each level is presented. Although the optimization method employed for each $j$ is commonly denoted as the smoother within MG/OPT, it should be made clear that in general there are no high- or low frequency spaces associated with the solutions. Rather the MG/Opt method should be understood in the sense of adopting and extending the nested iteration concept of computing improved initial guesses. Furthermore, in [150] it is made clear that, at least for grid-based problems, with respect to computational efficiency the MG/Opt-related approaches are not on par with the aforementioned CSGM or other defect-correction approaches. The strength of the MG/Opt method lies in the flexible framework which is complemented by a satisfactory convergence theory and thus might prove beneficial for more general optimization problems where classical grid-based approaches might not be applicable.

Last not least let us mention that in [45, 46] a multiscale approach based on wavelet discretizations and employing a nested conjugate gradient scheme was applied to elliptic optimal control problems.

We return now to the issue of constructing a multigrid method for the solution of saddle point systems (2.51). According to Algorithm 3 we need to define the transfer operators, the coarse grid solver, the representation of (2.51) on the grid levels $j$ and the smoothing iteration. We will begin with the last item.

## 4.3 A Smoothing Iteration for Discrete Optimality Systems

Analogously to the scalar case we define the smoothing iteration on grid level $j$ in terms of a preconditioned Richardson iteration

$$w_j^{k+1} = w_j^k + B_{K_j}^{-1}(b_j - K_j w_j^k). \tag{4.23}$$

The preconditioner $B_{K_j}$ will be based on a block-triangular constraint preconditioner $B_{CP}$ (3.23). Since the major part of the computational work required to evaluate $B_{CP}^{-1}$, besides the evaluation or inversion of $\hat{H}_Z$, originates from the full solution of the state and adjoint subproblems, we replace these by approximate solutions. To this end, we set

$$B_{K_j} = \begin{bmatrix} & & \hat{L}_j^T \\ & \hat{H}_{Z_j} & -M_j \\ \hat{L}_j & -M_j & \end{bmatrix}, \tag{4.24}$$

where the blocks $\hat{L}_j$ and $\hat{L}_j^T$ are suitable approximations to the differential operators $L_j$ and $L_j^T$, respectively. We stress the fact that these approximations will not be constructed explicitly but rather are given implicitly by the definition of the actions of $\hat{L}_j^{-1}$ and $(\hat{L}_j^T)^{-1}$ on a given vector. The precise form of these approximations will be given below. Note that the (2,3) and (3,2) blocks $-M_j$ in (4.24) are inherited without modification from (3.23), since no inversion of $M_j$ is required when evaluating $B_{K_j}^{-1}$. Note that due to these approximations, $B_{K_j}$ is not a constraint preconditioner anymore. One step of the smoothing iteration defined by (4.23) with preconditioner $B_{K_j}$ (4.24) is given by Algorithm 5. Again, different possible approximations $\hat{H}_Z$ can

$\tilde{w}_j = \mathcal{S}_j^1(w_j, b_j)$

| | | |
|---|---|---|
| 1: | $v_j = b_j - K_j w_j$ | /∗ compute local residual[a] ∗/ |
| 2: | $\tilde{w}_j^p \leftarrow \hat{L}_j^T w_j^p = v_j^y$ | /∗ apply $B_{K_j}^{-1}$ ∗/ |
| 3: | $\tilde{w}_j^u \leftarrow \hat{H}_Z w_j^u = v_j^u + M_j \tilde{w}_j^p$ (RHA) | |
| 4: | $\tilde{w}_j^y \leftarrow \hat{L}_h w_j^y = v_j^p + M_j \tilde{w}_j^u$ | |
| 5: | $\tilde{w}_j = \tilde{w}_j + w_j$ | /∗ add correction ∗/ |

[a]Not to be confused with the restricted residual, i.e. the right hand side $b_j$ of the error equation.

Algorithm 5: One smoothing step defined by iteration (4.23) with preconditioner $B_{K_j}$ given by (4.24).

be made. The most simple choice is

$$\hat{H}_Z = \sigma M_j. \tag{4.25}$$

As remarked in Section 3.2, no approximation to $M_j$ is needed since $M_j$ is diagonal for the employed discretization scheme, cf. Section 2.3.2. But also the general case of non-diagonal $M_j$ is not problematic, since $M_j$ in general is s.p.d. and well-conditioned.

A second choice we make is to define $\hat{H}_Z$ analogously to Section 3.2 by a matrix-vector product and approximately solve (RHA) by a few steps of a suitable iterative method. The expression (3.12) suggests to define

$$\hat{H}_Z = M_j^T \hat{L}_j^{-T} M_j \hat{L}_j^{-1} M_j + \sigma M_j, \tag{4.26}$$

where the same approximations $\hat{L}_j, \hat{L}_j^T$ as in (4.24) are used. The evaluation of (4.26) within an iterative method proceeds analogously to Algorithm 2 and is stated in Algorithm 6. Also here, no approximation of $M_j$ is needed. In the extremal case

$\tilde{x}_j^u = \hat{H}_Z x_j^u = (M_j \hat{L}_j^{-T} M_j \hat{L}_j^{-1} M_j + \sigma M_j) x_j^u$

1: $v_j \leftarrow \hat{L}_j v_j = M_j x_j^u$
2: $w_j \leftarrow \hat{L}_j^T w_j = M_j v_j$
3: $\tilde{x}_j^u \leftarrow M_j w_j + \sigma M_j x_j^u$

Algorithm 6: Definition of approximate MatVec $\hat{H}_Z x_j^u$ given by (4.26). MatVec is employed within an iterative method for the approximate solution of (RHA), cf. line 3 in Algorithm 5.

$\hat{L}_j = L_j$, the operator $\hat{H}_Z$ defined by (4.26) is positive definite, cf. Section 3.2. Therefore, we employ a few steps of the conjugate gradient method for the approximate solution of (RHA). In our numerical experiments often just one iteration turned out to be sufficient.

It remains to define the approximations $\hat{L}_j$ and $\hat{L}_j^T$. To this end, we employ a few steps (often just one) of an iterative method of the form (4.4) whenever the inverses $\hat{L}_j^{-1}$ and $\hat{L}_j^{-T}$ need to be evaluated, either in Algorithm 5 or 6. The preconditioners $B_{L_j}$ and $B_{L_j^T}$ needed in (4.4) usually result from a splitting of the matrices $\hat{L}_j$ and $\hat{L}_j^T$, respectively. The specific form is problem dependent. For $L_j$ being the discrete Laplacian we employ GS-LEX, cf. Section 4.1. Other, more complicated operators might require sophisticated schemes such as incomplete LU-factorizations (ILU) or alternating line Gauss-Seidel methods (ALGS). In general, any smoothing iteration that results in a robust multigrid solver for $L_j^T w_j^p = v_j^y$ or $L_j w_j^y = v_j^p + M_j \tilde{w}_j^u$, respectively, is a good candidate for (4.4) and its corresponding preconditioner $B_j$. This statement

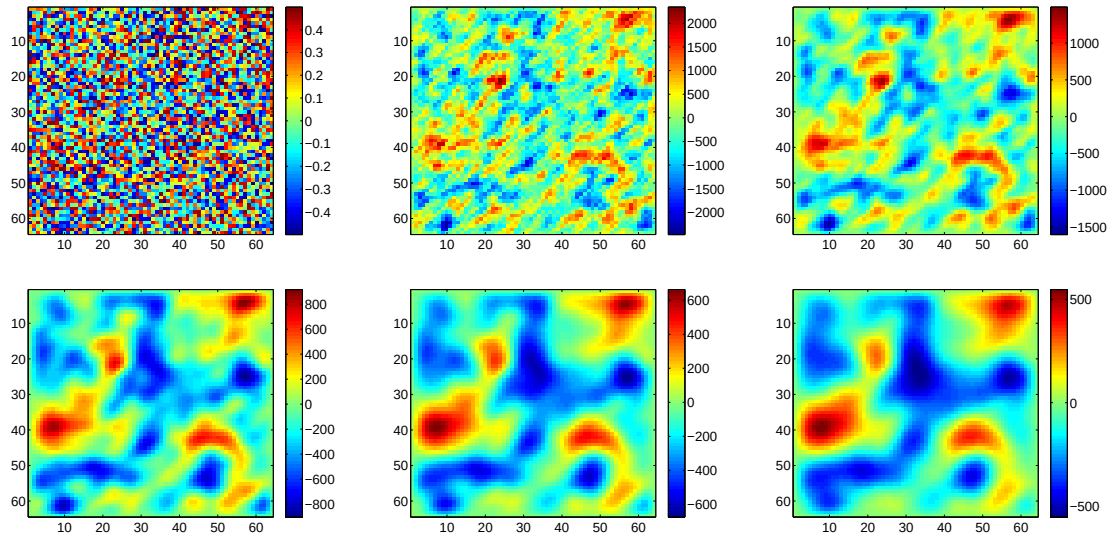


Figure 4.3: Smoothing effect of the iteration (4.27) with $\alpha = \beta = 1$ illustrated by color plots of $e_h^u$ after $\nu = 0, 1, 2, 4, 6, 8$ smoothing steps. The constraint smoother is GS-LEX, the mesh width is $h = 2^{-6}$ and the regularization parameter is $\sigma = 1_{-4}$.

will be further corroborated by a smoothing analysis, see below. In the following we will often use the term *constraint smoother* to denote the smoothing iteration employed for the subsystems with coefficient matrices $L_j, L_j^T$.

Putting it all together, we denote the smoothing iteration on grid level $j$ by

$$\tilde{w}_j = \mathcal{S}_{j,\alpha,\beta}^{\nu}(w_j, b_j), \tag{4.27}$$

meaning that $\tilde{w}_j$ is obtained from applying $\nu$ smoothing steps of the form (4.23) to $w_j$. Here, $\alpha$ denotes the number of iterations for solves with $\hat{L}_j$ and $\hat{L}_j^T$ and $\beta$ denotes the number of conjugate gradient steps applied to approximately solve (RHA). The smoothing iteration which results if we define $\hat{H}_Z$ by (4.25) is denoted with $\mathcal{S}_{j,\alpha}^{\nu}$.

Figure 4.3 gives an indication of the smoothing effect of the iteration (4.27) with $\alpha = \beta = 1$ and GS-LEX employed as constraint smoother. The Figure shows color plots of the error with respect to the control component, $e_h^u = u_h - u_h^*$ after $\nu$ smoothing iterations with $\nu = 0, 1, 2, 4, 6$ and 8. The mesh width is $h = 2^{-6}$ and the regularization parameter is $\sigma = 1_{-4}$.

The different levels of inexactness and approximations as well as the nested iterations severely complicate a rigorous analysis of the convergence and smoothing properties of the just defined iterative method. In general, iteration (4.27) is a slowly or not at all converging method. However, it will turn out to be an efficient smoother when employed within the multigrid algorithms 3 or 4. Nevertheless we will to some extent investigate the behavior of the iteration (4.27) when applied to the linear-quadratic model problem ($\mathrm{LQP}_h$). To this end, let us briefly consider the extremal case that

$\hat{L}_j = L_j$ and $\hat{L}_j^T = L_j^T$. A similarity transformation, cf. (3.20), implies that the iteration matrix $G = I - B_{K_j}^{-1} K_j$ is similar to

$$\tilde{G} = \begin{bmatrix} 0 & & \\ \star & I - \hat{H}_Z^{-1} H_Z & \\ \star & \star & 0 \end{bmatrix}. \tag{4.28}$$

Recall from Section 3.2 that, if additionally $\hat{H}_Z = H_Z$, i.e. $B_{K_j} = B_{CP}$ (cf. definition (3.23)), preconditioned GMRES converges in three steps. First we remark that the same assertion holds for the $B_{K_j}$-preconditioned Richardson iteration, since in that case it is easy to see that $\tilde{G}$ is nilpotent of order three, i.e. $\tilde{G}^3 = 0$, which yields the assertion.

A first step at reducing the required computational effort was made by employing the cheap approximation (4.25), cf. Section 3.2, where the preconditioner resulting from (3.23) was denoted by $B_{CP_2}$. Setting $B_{K_j} = B_{CP_2}$ in (4.23) is closely related to Hackbusch's method [83] which was briefly discussed in Section 4.2. In fact, both approaches yield the same smoothing step with respect to the control component $w_j^u$. The smoothing step of (4.23) with respect to $w_j^u$ reads

$$\tilde{w}_j^u = (I - \hat{H}_Z^{-1} H_Z) w_j^u + q, \tag{4.29}$$

where $q$ is a vector whose precise expression is irrelevant for the following arguments, it suffices to say that $q$ depends on the state component $w_j^y$ (due to the nonzero block in (4.28) denoted by the symbol $\star$) and on $B_{K_j}^{-1} b_j$. Using (4.25) in (4.29), we see that (4.29) is precisely of the form (4.21) with $C_h$ given by (using the subscript $j$ instead of $h$)

$$C_j = -\frac{1}{\sigma} M_j^{-1} M_j^T L_j^{-T} M_j L_j^{-1} M_j. \tag{4.30}$$

The results in Section 3.2 clearly showed that the approximation $\hat{H}_Z$ introduced a dependence on the regularization parameter $\sigma$ with deteriorating convergence for $\sigma \to 0$. In fact, for small values of $\sigma$, the method converges only due to the outer GMRES iteration. The employed preconditioner, i.e. the Richardson iteration (4.23), is not a convergent iteration by itself if $\sigma$ is too small. In the case of ($\mathrm{LQP}_h$) we can infer a condition on $\sigma$ which ensures convergence from the requirement that $\varrho(C_j) < 1$ holds, where $\varrho(C_j)$ is the spectral radius of $C_j$. To this end, let $\lambda$ denote an eigenvalue of $L_j$. From (4.30) then follows the condition

$$\sigma > \frac{1}{\lambda^2} h_j^4, \tag{4.31}$$

which is most restrictive for $\lambda = \lambda_{\min}$ the smallest eigenvalue. For $L_j$ being the discrete Laplacian we have the estimate (see, e.g. [86])

$$\|L_j^{-1}\|_2 = \frac{1}{\lambda_{\min}} \leq \frac{1}{8 \sin^2(\pi h_j / 2)} = \frac{1}{2\pi^2 h_j^2 + \mathcal{O}(h_j^4)}, \tag{4.32}$$

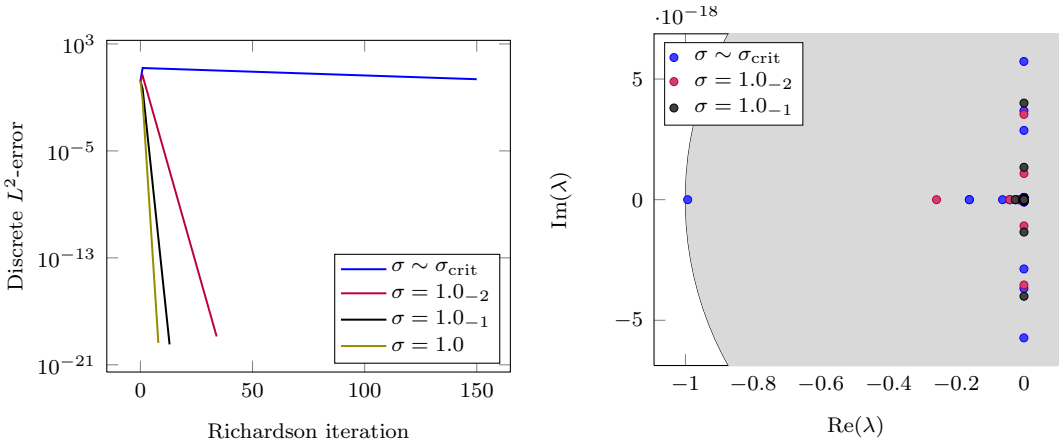



Figure 4.4: Left: Convergence of $B_{K_j}$-Richardson, $\hat{H}_Z = B_{H_Z}$, $L_h, L_h^T$ inverted by direct factorization, fixed mesh size $h = 2^{-8}$ and regularization parameters $\sigma$ between 1.0 and $2.6_{-3} \sim \sigma_{\text{crit}}$. Right: Eigenvalues of the iteration matrix $G$ for $\sigma = 2.6_{-3}, 1.0_{-2}, 1.0_{-1}$, $h = 2^{-4}$.

and together with (4.31) we obtain

$$\sigma > \frac{1}{4\pi^4} = \sigma_{\text{crit}}. \tag{4.33}$$

Apart from minor technical details concerning the discretization of $L_j$, this corresponds to a similar condition given in [84] and investigated in numerical experiments. Furthermore, the same condition has been derived in the context of reduced space methods, where Picard linearization is applied and the resulting $H_Z$ is inverted with CG iterations. We stress the fact that (4.33) exclusively depends on the extremal eigenvalues of $L_h$ or rather $\|L_j^{-1}\|_2$, which in the case of an elliptic operator does not depend on the mesh size $h$. Thus, for several single-mesh algorithms which use (4.25) as a simple approximation $\hat{H}_Z$, convergence can not be established if $\sigma$ violates (4.33). For multigrid methods, the situation is slightly different, as will be explained below.

Figure 4.4 on the left shows the convergence history of the Richardson iteration preconditioned with $B_{CP_1}$ for the mesh size $h = 2^{-8}$ and values of the regularization parameter $\sigma$ ranging between 1.0 and $2.6_{-3}$, which is close to the value of $\sigma_{\text{crit}}$. In the same Figure on the right we depict the eigenvalue distribution of the iteration matrix $G$ for $\sigma = 2.6_{-3}, 1.0_{-2}$ and $1.0_{-1}$ and $h = 2^{-4}$. For large to moderate $\sigma$ the convergence is fast, e.g. occurring in less than ten iterations for $\sigma = 1.0$, thus matching the results for preconditioned GMRES presented in Section 3.3. All eigenvalues of $G$ are real (up to numerical round-off error) and the majority of eigenvalues is clustered tightly around zero, which is attributed to the direct factorizations of $L_j$ and $L_j^T$. However, for smaller values of $\sigma$, the reduced approximation quality of $\hat{H}_Z$ generates eigenvalues approaching $-1$, which eventually become much smaller if $\sigma < \sigma_{\text{crit}}$, leading to divergence.

In the context of a multigrid method, focusing on the convergence properties of (4.23) might be too pessimistic, in fact, in the numerical results in Section 4.5 we will see that the parameter limit for the multigrid convergence is not as strict as (4.33) suggests. This is owing to the fact that the smoothing property of (4.23) is crucial for the success of a multigrid method, not its convergence properties.[3] Nevertheless, the previous considerations let us expect that the approximation quality of (4.27) will depend on the quality of the constraint smoother and that the relation of the regularization parameter $\sigma$ to the mesh size $h$ will further impact the convergence and smoothing properties.

### 4.3.1 Local Fourier Smoothing Analysis

The aim of the local Fourier analysis (LFA), or local mode analysis as it is termed by Brandt, is to give *quantitative estimates* of the smoothing and convergence factors for a practical multigrid method. In this respect it differs from classical convergence proofs which mostly provide qualitative results. The LFA is seen as an essential tool for the design of an efficient multigrid method for general problems. Here we focus on the smoothing analysis based on LFA, i.e. the aim of this section is to determine (an estimate of) the smoothing factor $\mu(\mathcal{S}_{j,\alpha,\beta})$ for smoothing iterations of the type (4.27). For a detailed introduction into the topic of LFA we refer to [147].

The LFA considers the effect of all appearing operators in the multigrid method when applied to the frequency functions

$$\phi(\theta, x) = e^{i\theta \cdot x/h} = e^{i\theta_1 x_1/h} e^{i\theta_2 x_2/h} \tag{4.34}$$

on the infinite grid[4]

$$G_h = \{x = hk = (hk_1, hk_2) | k \in \mathbb{Z}^2\}. \tag{4.35}$$

Thus the LFA neglects the boundary treatment within the multigrid algorithm and in this sense predicts rates which can be obtained provided a proper boundary treatment is conducted. Due to the periodicity of (4.34) it is sufficient to consider the frequencies

$$\theta = (\theta_1, \theta_2) \in [-\pi, \pi)^2. \tag{4.36}$$

With $G_h$ we associate the infinite coarse grid $G_H$, which is defined analogously to (4.35) for the coarse mesh size $H = 2h$. Due to aliasing, on $G_H$ only those frequencies with $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2$ can be distinguished, which leads to the definition that

$$\phi(\theta, x) \text{ is a low frequency component if } \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2 \tag{4.37}$$

$$\phi(\theta, x) \text{ is a high frequency component if } \theta \in \left[-\pi, \pi\right)^2 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2. \tag{4.38}$$

[3] Naturally, the smoothing iteration has to be a convergent process if it is to be used as coarse grid solver, which is often the case in practice.

[4] For notational simplicity we confine ourselves to the case of a uniform mesh size $h$ in both coordinate directions.

For $\theta \in \left[-\pi,\pi\right)^2$ all functions $\phi(\theta,x)$ are the formal eigenfunctions of any discrete operator represented by a difference stencil, i.e. the relation

$$L_h\phi(\theta,x) = \tilde{L}_h(\theta)\phi(\theta,x) \tag{4.39}$$

holds with

$$\tilde{L}_h(\theta) = \sum_{\mathbf{j}} s_{\mathbf{j}} e^{i\theta\cdot\mathbf{j}}, \tag{4.40}$$

where $\mathbf{j} \in \mathbb{Z}^2$ and $s_{\mathbf{j}}$ are the stencil elements of $L_h$. Due to (4.39) we call $\tilde{L}_h(\theta)$ the formal eigenvalue or *symbol* of the operator $L_h$. For the standard 5-point stencil of the discrete Laplacian, for example, one obtains

$$\tilde{L}_h(\theta) = \frac{1}{h^2}\Big(4 - \big(e^{i\theta_1} + e^{i\theta_2} + e^{-i\theta_1} + e^{-i\theta_2}\big)\Big) = \frac{2}{h^2}(2 - \cos\theta_1 - \cos\theta_2). \tag{4.41}$$

In order to apply the LFA based smoothing analysis we have to assume that the smoothing iteration can be given by a local splitting, i.e. a splitting based on the stencil notation,

$$L_h^+\tilde{w}_h + L_h^- w_h = f_h, \tag{4.42}$$

where, as before, $\tilde{w}_h$ denotes the smoothed approximation of $w_h$. This is a natural assumption for many classical methods such as $\omega$-JAC and GS-LEX, however it is not satisfied for coloring-based Gauss-Seidel methods. For the necessary modifications to treat e.g. red-black Gauss-Seidel within the LFA framework we refer to [147]. Note that the use of the conjugate gradient iteration for (RHA) in Algorithm 5 prevents to define (4.27) by a splitting of a system corresponding to (4.42), however we can apply the LFA to the iteration $\mathcal{S}_{j,1}$, since in this case a splitting is obtained easily.

From (4.42) we obtain the symbol of the corresponding smoothing operator $S_{L_h}$ as

$$\tilde{S}_{L_h}(\theta) = -\frac{\tilde{L}_h^-(\theta)}{\tilde{L}_h^+(\theta)}, \tag{4.43}$$

where $\tilde{L}_h^+(\theta)$ and $\tilde{L}_h^-(\theta)$ are the symbols of $L_h^+$ and $L_h^-$, respectively. The smoothing rate $\mu_{\mathrm{LFA}}(S_{L_h})$ is then defined as the largest possible amplification factor with respect to the high frequency components,

$$\mu_{\mathrm{LFA}}(S_{L_h}) = \sup\left\{ |\tilde{S}_{L_h}(\theta)| : \theta \in \Big[-\pi,\pi\Big)^2 \setminus \Big[-\frac{\pi}{2},\frac{\pi}{2}\Big)^2 \right\}. \tag{4.44}$$

In order to perform the LFA smoothing analysis for (4.27) we need the extension of the definitions (4.39) and (4.44) to systems of equations. To this end, we introduce the frequency functions

$$\Phi(\theta,x) = (1,\ldots,1)^T\phi(\theta,x). \tag{4.45}$$

Then we obtain

$$K_h\Phi(\theta,x) = \tilde{K}_h(\theta)\Phi(\theta,x) \tag{4.46}$$

with the symbol

$$\tilde{K}_h(\theta) = \begin{pmatrix} \tilde{K}_h^{1,1}(\theta) & \dots & \tilde{K}_h^{1,Q}(\theta) \\ \vdots & & \vdots \\ \tilde{K}_h^{Q,1}(\theta) & \dots & \tilde{K}_h^{Q,Q}(\theta) \end{pmatrix}, \tag{4.47}$$

where $\tilde{K}_h^{l,q}(\theta), 1 \le l,q \le Q$ are the symbols of the scalar discrete operators $K_h^{l,q}$ of the $Q \times Q$-system given by $K_h$. Corresponding to the splitting (4.42) in the scalar case we assume that (4.27) can be given by a splitting

$$K_h^+ \tilde{w}_h + K_h^- w_h = f_h, \tag{4.48}$$

with the associated symbols $\tilde{K}_h^+(\theta), \tilde{K}_h^-(\theta)$. The smoothing factor for a system of equations now is defined by

$$\mu_{\mathrm{LFA}}(S_h) = \sup\left\{ \rho\Big( \big(\tilde{K}_h^+(\theta)\big)^{-1} \tilde{K}_h^-(\theta) \Big) : \theta \in \Big[-\pi,\pi\Big)^2 \setminus \Big[-\frac{\pi}{2},\frac{\pi}{2}\Big)^2 \right\}, \tag{4.49}$$

with $\rho$ denoting the spectral radius. As mentioned before, $S_{j,1}$ is amenable to a splitting of the form (4.48), which is given by

$$K_h^+ = \begin{pmatrix} & & L_h^{T,+} \\ & \hat{H}_Z & -M_h \\ L_h^+ & -M_h & \end{pmatrix}, \quad K_h^- = \begin{pmatrix} M_h & & L_h^{T,-} \\ & & \\ L_h^- & & \end{pmatrix}, \tag{4.50}$$

with $L_h^+, L_h^-$ and $L_h^{T,+}, L_h^{T,-}$ resulting from a splitting (4.42) of the state and adjoint operators $L_h$ and $L_h^T$, respectively. For notational simplicity in the following we assume that $L_h = L_h^T$. The symbols corresponding to (4.50) then are given by

$$\tilde{K}_h^+(\theta) = \begin{pmatrix} & & \tilde{L}_h^+(\theta) \\ & \tilde{\hat{H}}_Z(\theta) & -\tilde{M}_h(\theta) \\ \tilde{L}_h^+(\theta) & -\tilde{M}_h(\theta) & \end{pmatrix}, \quad \tilde{K}_h^-(\theta) = \begin{pmatrix} \tilde{M}_h(\theta) & & \tilde{L}_h^-(\theta) \\ & & \\ \tilde{L}_h^-(\theta) & & \end{pmatrix}. \tag{4.51}$$

With

$$\Big(\tilde{K}_h^+(\theta)\Big)^{-1} = \begin{pmatrix} \tilde{A}_h(\theta)\tilde{H}_h(\theta)(\tilde{L}_h^+(\theta))^{-1} & \tilde{A}_h(\theta) & (\tilde{L}_h^+(\theta))^{-1} \\ \tilde{H}_h(\theta)(\tilde{L}_h^+(\theta))^{-1} & (\tilde{\hat{H}}_Z(\theta))^{-1} & 0 \\ (\tilde{L}_h^+(\theta))^{-1} & 0 & 0 \end{pmatrix} \tag{4.52}$$

we obtain

$$\tilde{S}_h(\theta) = \begin{pmatrix} \tilde{A}_h(\theta)\tilde{H}_h(\theta)\tilde{A}_h(\theta) + \tilde{S}_{L_h}(\theta) & 0 & \tilde{A}_h(\theta)\tilde{H}_h(\theta)\tilde{S}_{L_h}(\theta) \\ \tilde{H}_h(\theta)\tilde{A}_h(\theta) & 0 & \tilde{H}_h(\theta)\tilde{S}_{L_h}(\theta) \\ \tilde{A}_h(\theta) & 0 & \tilde{S}_{L_h}(\theta) \end{pmatrix} \tag{4.53}$$

as the symbol of the smoothing operator corresponding to the iteration $\mathcal{S}_{j,1}$. Both in (4.52) and (4.53) we have introduced the abbreviations

$$\tilde{A}_h(\theta) = (\tilde{L}_h^+(\theta))^{-1}\tilde{M}_h(\theta) \text{ and } \tilde{H}_h(\theta) = (\tilde{\hat{H}}_Z(\theta))^{-1}\tilde{M}_h(\theta). \tag{4.54}$$

Furthermore, note that $\tilde{S}_{L_h}(\theta)$ is the symbol of the smoothing iteration employed for the constraints $L_h$, i.e it is based on a splitting of $L_h$ and given by (4.43). According to (4.49), the smoothing factor $\mu_{\text{FLA}(S_h)}$ is given by the largest absolute eigenvalue of (4.53), which we determine by computing the roots of the polynomial

$$\begin{aligned}&\det\big(\tilde{S}_h(\theta) - \lambda\big)\\ &\quad= \lambda\Big(\big(\tilde{A}_h(\theta)^2\tilde{H}_h(\theta) + \tilde{S}_{L_h}(\theta) - \lambda\big)\big(\lambda - \tilde{S}_{L_h}(\theta)\big) + \tilde{A}_h(\theta)^2\tilde{H}_h(\theta)\tilde{S}_{L_h}(\theta)\Big)\end{aligned} \tag{4.55}$$

The two roots save $\lambda = 0$ are given by

$$\lambda_{1,2} = \tilde{S}_{L_h}(\theta) + \frac{\tilde{A}_h(\theta)^2\tilde{H}_h(\theta)}{2} \pm \sqrt{\frac{\tilde{A}_h(\theta)^2\tilde{H}_h(\theta)}{2}\Big(2\tilde{S}_{L_h}(\theta) + \frac{\tilde{A}_h(\theta)^2\tilde{H}_h(\theta)}{2}\Big)} \tag{4.56}$$

This expression already conveys that $\mu_{\text{LFA}}(S_h)$ can be considered as a perturbation of the constraint smoothing factor, depending on $\tilde{S}_{L_h}(\theta)$, with terms depending on the smoothing with respect to the control component, governed by $\tilde{H}_h(\theta)$. More insight can be obtained if we consider the model problem ($\text{LQP}_h$), i.e. $L_h$ is the discrete Laplacian. Furthermore, we eliminate the explicit dependence of $\rho\big(\tilde{S}_h(\theta)\big)$ on $\theta$ by considering $\omega$-JAC as constraint smoother. Under these assumptions, and choosing the optimal $\omega = 4/5$, we have $\tilde{S}_{L_h}(\theta) = 3/5$. Furthermore, $\tilde{H}_h(\theta) = 1/\sigma$ and since for $\omega$-JAC one obtains $\big(\tilde{L}_h^+(\theta)\big)^{-1} = \omega/4$ we have $\tilde{A}_h(\theta) = \frac{1}{5}h^2$. Substituting these quantities in (4.56) we obtain

$$\mu_{\text{LFA}}(S_j) = \frac{3}{5} + \frac{h_j^4}{50\sigma} + \sqrt{\frac{h_j^4}{50\sigma}\big(\frac{6}{5} + \frac{h_j^4}{50\sigma}\big)} \tag{4.57}$$

as the smoothing factor for the iteration $\mathcal{S}_{j,1}$ with $\omega$-JAC, $\omega = 4/5$, as constraint smoother when applied to the linear-quadratic model problem ($\text{LQP}_h$). Note that $\mu_{\text{LFA}}(S_j)$, in contrast to $\mu_{\text{LFA}}$ in the scalar case, explicitly depends on the mesh size $h_j$. This is usually the case when deriving smoothing factors for systems, due to differential operators of different order appearing in $K_j$. In Figure 4.5 we plot the smoothing factor $\mu_{\text{LFA}}(S_j)$ in (4.57) as a function of the mesh size $h_j$ and the regularization parameter $\sigma$ for the range $(\sigma, h_j) \in \big[1.0, 1.0_{-6}\big] \times \big[2^{-2}, 2^{-12}\big]$. The large area which is colored dark blue indicates that for these combinations of $\sigma$ and $h_j$ the smoothing factor $\mu_{\text{LFA}}(S_j)$ is close to that of $\mu_{\text{LFA}}(S_{L_h})$ for the constraint smoother. On the other hand, dark red coloring indicates that $\mu_{\text{LFA}}(S_j) > 1$, which is obtained for combinations of small

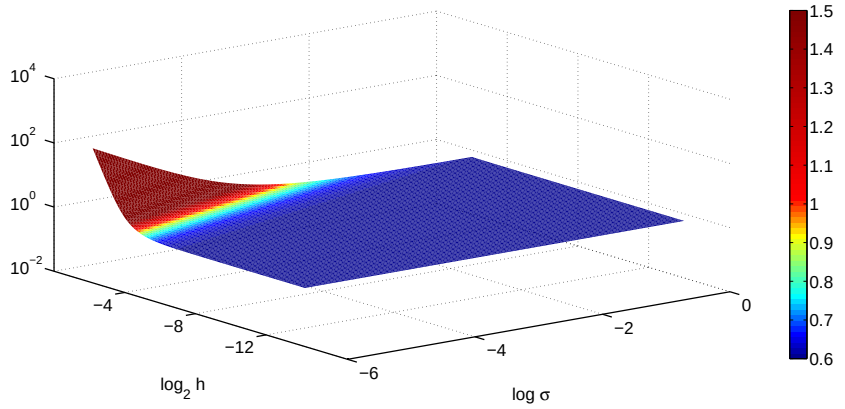



Figure 4.5: The smoothing factor $\mu_{\text{LFA}}(S_j)$ (4.57) for the smoothing iteration $\mathcal{S}_{j,1}$ with $\omega$-JAC as constraint smoother, plotted as a function of the regularization parameter $\sigma \in \left[1.0, 1.0_{-6}\right]$ and the mesh size $h_j \in \left[2^{-2}, 2^{-12}\right]$.

$\sigma$ and large $h_j$. A smoothing factor $\mu_{\text{LFA}}(S_j) > 1$ is an indication for degradation of convergence or even divergence of the multigrid process. For (4.57) holds that $\mu_{\text{LFA}}(S_j) = 1$ for $\sigma = h_j^4/4$ and thus we obtain the condition

$$\mu_{\text{LFA}}(S_j) < 1 \text{ if } \sigma > \frac{h_j^4}{4}. \tag{4.58}$$

In Section 4.5 we will further elaborate on (4.58) with numerical experiments and an ensuing discussion of the consequences of (4.58) for the robustness of our multigrid solver. A modification of the control-smoothing component which avoids (4.58) will be developed in Section 4.5.5.

## 4.4 A Multigrid Method for Discrete Optimality Systems

### Grid Hierarchy and Interpolation Operators

Naturally, the grid hierarchy for the optimality system is the one which is implied by the hierarchy of the underlying discretized PDEs and was already given in Section 4.1. For a system of equations, the rule of thumb (4.13) translates to

$$m_{R_\alpha} + m_{P_\beta} > m_{\alpha\beta}, \quad \alpha, \beta = 1, 2, 3 \tag{4.59}$$

where $m_{\alpha\beta}$ is the order of the differential operator corresponding to the $\beta$-th unknown in equation $\alpha$, $R_\alpha$ is the restriction operator for the residual of equation $\alpha$ and $P_\beta$ is the interpolation employed for the $\beta$-th unknown. Consequently we employ the same

interpolation operators as in the scalar case, namely the prolongation $P^j_{j-1,\mathrm{BL}}$ (4.10) and the restriction $R^{j-1}_{j,\mathrm{FPA}}$ (4.12), and apply them in a blockwise fashion. To this end, we define the block-diagonal matrices

$$\mathcal{R}^{j-1}_j = \begin{pmatrix} R^{j-1}_{j,\mathrm{FPA}} & & \\ & R^{j-1}_{j,\mathrm{FPA}} & \\ & & R^{j-1}_{j,\mathrm{FPA}} \end{pmatrix}, \quad \mathcal{P}^j_{j-1} = \begin{pmatrix} P^j_{j-1,\mathrm{BL}} & & \\ & P^j_{j-1,\mathrm{BL}} & \\ & & P^j_{j-1,\mathrm{BL}} \end{pmatrix}. \tag{4.60}$$

Keep in mind that for the chosen discretization scheme and grid hierarchy, all unknowns are located at the cell centers and therefore (4.60) best suits our needs, although the general approach offers more flexibility in that different interpolation operators could be employed in each block-row. This might be of special interest when the OTD approach and different discretization schemes for state and adjoint unknowns are employed, possibly resulting in different grid locations for the different unknowns.

### The Coarse Grid Solver

It remains to define the coarse grid solver component, i.e. the solution method for the error equation on the coarsest level $j = 0$,

$$K_0 x_0 = b_0. \tag{4.61}$$

Here we have three options, we can

- employ the smoothing iteration (4.6) and iterate until convergence,
- employ three steps of the $P_{CP}$-GMRES method,
- assemble $K_0$ and employ a direct factorization.

Numerical experiments showed no difference between these approaches. Since the first option can only be employed if the iteration is a convergent process, cf. the discussion in Section 4.3, we avoid this choice. Also with respect to computational efficiency, authoritative measurements are difficult due to the smallness of $N$ for $j = 0$. Unless noted otherwise, in the sequel we employ the $P_{CP}$-GMRES method for implementational reasons since the same data structures can be used on all levels including the coarsest one.

## 4.5 Numerical Results

In this section we present numerical results obtained with the devised multigrid method. The first aim is to assess the convergence properties and efficiency of the multigrid method, to this end we consider the model problem ($\mathrm{LQP}_h$). Then we consider constraint equations which present greater difficulties. In particular, we test

our method on problems where the constraint equation is given by the diffusion and convection-diffusion equation and in settings for which pointwise smoothing is known to fail for the scalar problem. The section is closed with a discussion and experiments regarding the robustness of the present method with respect to the regularization parameter $\sigma$.

Our measure for the convergence speed of the multigrid iteration is based on the reduction factor

$$\varrho^m = \frac{\|v_J^m\|}{\|v_J^{m-1}\|}, \tag{4.62}$$

where the superscript $m$ denotes the multigrid iteration index and $v_J^m$ is some suitable fine grid quantity available at each iteration. In general, measurements can only be based on the fine grid residual $r_J^m = b_J - K_J x_J^m$ and in this case $\|\cdot\|$ denotes the Euclidean norm. However, in particular for ill-conditioned problems, convergence speed estimates with respect to $r_J$ can significantly deviate from those with respect to the actual error. Therefore, in situations permitting we construct an analytically known exact solution and base the measurement on the error norms defined in Section 3.3. In particular, recall definition (3.24) of $e_{u_h}^m$, denoting the discrete $L^2$-error with respect to the control component, and definition (3.25) for the total discrete $L^2$-error $e_h^m$. Naturally, both error definitions are to be understood here with respect to the fine grid, i.e. $h = h_J$. An estimate for the asymptotic convergence factor of the multigrid iteration can then be given either by $\varrho^m$ or as the geometric mean

$$\varrho_{\text{avg}}^m = \sqrt[m-m_0]{\prod_{i=m_0}^{m} \varrho^i} = \sqrt[m-m_0]{\frac{\|v_h^m\|}{\|v_h^{m_0}\|}}, \tag{4.63}$$

with a small positive number $m_0$. In both cases, the number $m$ of iterations has to be suitably large. Since the first few iterations usually do not reflect the asymptotic convergence behavior very well, it is common to begin the measurements only after $m_0$ iterations have been performed. In this respect, recall the discussion in Section 3.3 with respect to feasible vs. infeasible initial guesses $x_J^0$. Therefore, in all our experiments, we set $m_0 = 1$ in order to avoid the influence of a particular initial guess $x_J^0$ on the initial residual $r_J^0$ or error $e_J^0$. The default smoothing iteration is $\mathcal{S}_{j,1,1}^{\nu}$ and lexicographical pointwise Gauss-Seidel as constraint smoother. Other parameters for $\alpha, \beta$ and different choices for constraint smoothing will be discussed in Sections 4.5.2 and 4.5.3. A comparative study of $\mathcal{S}_{j,1,1}^{\nu}$ and $\mathcal{S}_{j,1}^{\nu}$ will be given in Section 4.5.5.

### 4.5.1 A Model Problem

We begin with the discussion of the linear-quadratic model problem ($\text{LQP}_h$) as defined in Section 2.3.3. The regularization parameter is chosen as $\sigma = 1.0_{-2}$. The target state $\bar{y}_J$ as well as the right-hand side $f_J$ are set to zero, such that the exact solution

Table 4.1: Reduction factors $\varrho^{20}$ and $\varrho^{20}_{\text{avg}}$ with respect to the discrete $L^2$-error $e_h$ for the linear-quadratic model problem ($\text{LQP}_h$) and different types of multigrid cycles.

| | | $J$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | 5 | 6 | 7 | 8 | 9 | 10 |
| $V_{1,1}$ | $\varrho^{20}$ | $1.13_{-1}$ | $1.14_{-1}$ | $1.14_{-1}$ | $1.14_{-1}$ | $1.15_{-1}$ | $1.15_{-1}$ |
| | $\varrho^{20}_{\text{avg}}$ | $1.05_{-1}$ | $1.06_{-1}$ | $1.08_{-1}$ | $1.08_{-1}$ | $1.09_{-1}$ | $1.09_{-1}$ |
| $V_{2,1}$ | $\varrho^{20}$ | $7.07_{-2}$ | $7.21_{-2}$ | $7.30_{-2}$ | $7.31_{-2}$ | $7.31_{-2}$ | $7.31_{-2}$ |
| | $\varrho^{20}_{\text{avg}}$ | $6.57_{-2}$ | $6.79_{-2}$ | $6.84_{-2}$ | $6.89_{-2}$ | $6.93_{-1}$ | $6.93_{-1}$ |
| $V_{2,2}$ | $\varrho^{20}$ | $5.15_{-2}$ | $5.26_{-2}$ | $5.34_{-2}$ | $5.35_{-2}$ | $5.35_{-2}$ | $5.35_{-2}$ |
| | $\varrho^{20}_{\text{avg}}$ | $4.70_{-2}$ | $4.92_{-2}$ | $4.99_{-2}$ | $5.04_{-2}$ | $5.07_{-2}$ | $5.07_{-2}$ |
| $F_{1,1}$ | $\varrho^{20}$ | $8.18_{-2}$ | $8.20_{-2}$ | $8.20_{-2}$ | $8.20_{-2}$ | $8.21_{-2}$ | $8.21_{-2}$ |
| | $\varrho^{20}_{\text{avg}}$ | $7.81_{-2}$ | $7.82_{-2}$ | $7.88_{-2}$ | $7.86_{-2}$ | $7.87_{-2}$ | $7.87_{-2}$ |
| $W_{1,1}$ | $\varrho^{20}$ | $8.18_{-2}$ | $8.20_{-2}$ | $8.20_{-2}$ | $8.20_{-2}$ | $8.21_{-2}$ | $8.21_{-2}$ |
| | $\varrho^{20}_{\text{avg}}$ | $7.80_{-2}$ | $7.82_{-2}$ | $7.88_{-2}$ | $7.86_{-2}$ | $7.87_{-2}$ | $7.87_{-2}$ |

of ($\text{LQP}_h$) is $u_J^* = 0$. The initial guess $x_J^0$ is set to a normalized random vector. Due to linearity, this is a sufficiently general test setting under the justified assumption that the random vector $x_J^0$ is general enough, i.e. contains components with respect to all eigenvalues of $K_J$. The primary reason as to why we prefer this setting over other common test settings is that a large number of multigrid iterations can be performed before the machine accuracy affects the results. Thus we can choose a suitably large $m$ in order to estimate the asymptotic convergence rate by $\varrho^m$ or $\varrho^m_{\text{avg}}$. As smoother we employ the iteration $\mathcal{S}^\nu_{j,\alpha,\beta}$ defined by (4.42) with $\alpha = \beta = 1$ and we use the lexicographical point Gauss-Seidel iteration as constraint smoother.

In Table 4.1 we present the reduction factors $\varrho^{20}$ and $\varrho^{20}_{\text{avg}}$ with respect to $e_h$ defined in (3.25) for different types of multigrid cycles. The size of the coarsest mesh is $h_c = 1/4$, the mesh size of the finest grid is then $h_J = 2^{-(J+2)}$. From Table 4.1 we clearly observe that the reduction rates are independent of the resolution on the finest mesh. Furthermore, the reduction rates are of the same order as the reduction rates which one obtains when solving the scalar Poisson model problem for cell-centered discretizations with a multigrid method which employs the same smoothing iteration that is employed as the constraint smoother within $\mathcal{S}^\nu_{j,\alpha,\beta}$. In particular, for the pair $P^j_{j-1,\text{BL}}$, $R^{j-1}_{j,\text{FPA}}$ a convergence factor $\varrho^{20} = 0.114$ was obtained for the $V_{1,1}$-cycle with GS-LEX smoothing applied to the scalar model problem, cf. Figure 4.2. Here, the $V_{1,1}$-cycle with $\mathcal{S}^\nu_{j,1,1}$ and GS-LEX as constraint smoother yields the reduction factor $\varrho^{20} = 0.115$. Thus we find that the smoothing factor of the constraint smoother dominates the iteration. For the chosen value of $\sigma = 1_{-2}$ this is consistent with (4.57). Although (4.57) was derived specifically for $\mathcal{S}^\nu_{j,1}$, it was already pointed out that we

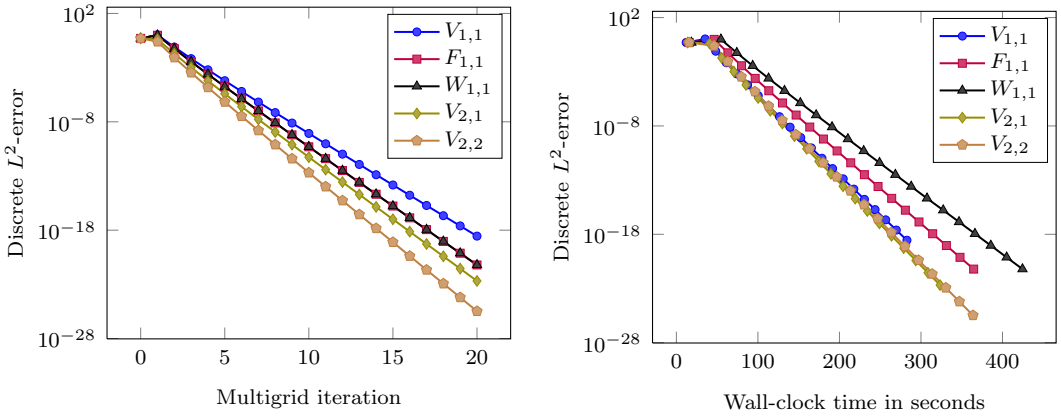



Figure 4.6: Reduction of discrete $L^2$-error $e_J^m$ vs. number of multigrid iterations $m$ (left) and wall-clock time measured in seconds (right). The fine grid mesh size is $h_J = 2^{-10}$ and the regularization parameter is $\sigma = 1_{-2}$.

expect the difference between the smoothing iterations $\mathcal{S}_{j,\alpha}^{\nu}$ and $\mathcal{S}_{j,\alpha,\beta}^{\nu}$ to manifest itself more clearly for smaller values of $\sigma$. This will be further discussed in Section 4.5.5.

Clearly we can expect stronger smoothing to result in better reduction rates. That this is indeed the case can also be gathered from Table 4.1. For the $V$-cycle we immediately see that a doubled amount of smoothing approximately leads to a proportionally reduced convergence factor. Since the cost for the $V_{2,2}$-cycle is double the cost for the $V_{1,1}$-cycle, we expect both approaches to have approximately the same efficiency. We also see that the $F_{1,1}$-cycle is more efficient than the $W_{1,1}$-cycle since it achieves the same convergence speed at a lower cost per iteration. This is also in accordance with observations for the scalar model problem.

Figure 4.6 shows the iteration histories of the different tested multigrid cycles for the fixed level $J = 8$. First we notice that the reduction factors $\varrho^m$ are essentially constant for each iteration. Furthermore, from the right part of Figure 4.6 we deduce that a smaller reduction factor per iteration does not always pay off in terms of computational efficiency, which is measured here by the wall-clock time. The reason is the higher cost per iteration due to more smoothing iterations or visits to the coarser grids in $F$- and $W$-cycles. Here, the most efficient cycle is the $V_{2,2}$-cycle, however the performance for all tested $V$-cycles is roughly the same, i.e. our preliminary considerations regarding the efficiency are affirmed here. For this model problem, the gain in convergence speed can not justify the higher cost for an $F$- or $W$-cycle. These findings are also in agreement with results for the scalar Poisson model problem.

In Figure 4.7 we show the convergence histories for the $V_{1,1}$- and the $W_{1,1}$-cycle with respect to the individual error components $e_J^{y,m}, e_J^{u,m}$ and $e_J^{p,m}$. The error reduction exhibits the same rate for all three components. For this problem $e_J^{y,m}$ and $e_J^{p,m}$ are visually not distinguishable. The convergence histories shown to the left of Figure 4.7

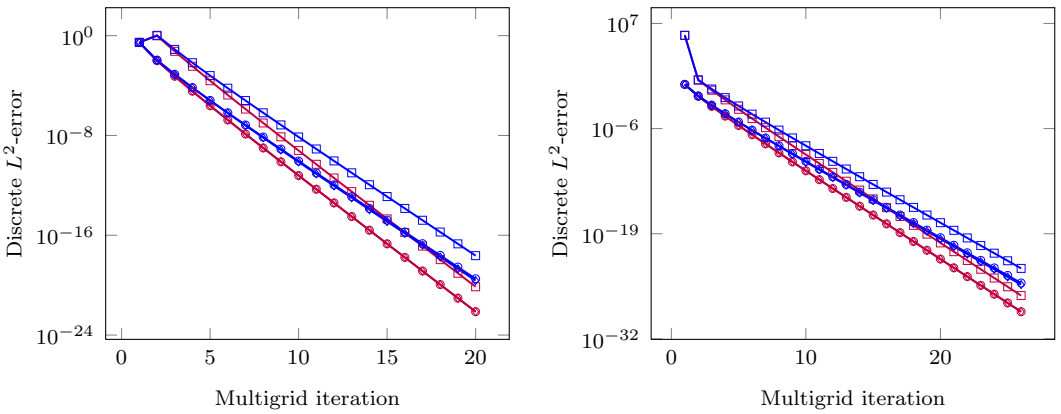



Figure 4.7: Discrete $L^2$-errors vs. number of multigrid iterations $m$ for feasible (left) vs. infeasible (right) initial guess $x_J^0$. The symbols denote the error norms $\circ$ $e_J^{y,m}$, $\square$ $e_J^{u,m}$ and $\diamond$ $e_J^{p,m}$. The colors indicate the — $V_{1,1}$-cycle and the — $W_{1,1}$-cycle. The fine grid mesh size is $h_J = 2^{-10}$ and the regularization parameter is $\sigma = 1_{-2}$.

are those corresponding to the previously shown results in Figure 4.6. Note the increase of $e_J^{u,1}$ which affects also $e_J^1$ as seen in Figure 4.6. Neither the residual norm (not depicted) nor the error components $e_J^y$ and $e_J^p$ exhibit this behavior but instead decrease monotonically. As was discussed in Section 3.3 the probable cause is the infeasibility and the wrong scaling of $x_J^0$. This is further corroborated by the results shown on the right. There, $x_J^0$ is constructed such that $L_J y_J^0 = M_J u_J^0$ holds. The feasibility of $x_J^0$ results in monotone convergence for all error components including $e_J^{u,m}$.

### Full Multigrid as Optimal Solver

For practical problems it does not make sense to reduce the algebraic error to the level of machine accuracy as we have done in the above experiments. Instead the multigrid iterations should be stopped when the algebraic error and the discretization error are balanced, i.e. when the algebraic error reaches the order $\mathcal{O}(h_J^\kappa)$, where $\kappa$ depends on the consistency order of the discretization. The computational cost to achieve this goal is of order $\mathcal{O}(h_J^{-d} \log h_J)$ for pure multigrid iterations, cf. the discussion at the beginning of this chapter. The full multigrid algorithm yields a solution with an error of order $\mathcal{O}(h_J^\kappa)$ at the optimal cost of $\mathcal{O}(h_J^{-d})$. We now solve the model problem using the full multigrid approach in order to show that we achieve the solution of the KKT system in $\mathcal{O}(h_J^{-d})$ cost. In order to measure the discretization error, we can not use the homogeneous problem as before. Instead, we set the right hand side $b_J = (M_J \bar{y}_J, 0, 0)$, where $\bar{y}_J$ is the discretization of a target state $\bar{y}$ such that an exact solution $y^*, u^*, p^*$ is known. In particular, we set $y^* = \sin(3\pi x_1)\sin(4\pi x_2)$, $u^* = -\Delta y^*$, $p^* = \sigma u^*$ and $\bar{y} = y^* - \Delta p^*$. On each level in FMG one multigrid iteration with a $V_{1,1}$-cycle is employed. This is sufficient here since the error reduction rate of the $V_{1,1}$-cycle is

Table 4.2: Discrete $L^2$-error $e_h$ and wall-clock time in seconds for one FMG cycle. On each level, one multigrid iteration with a $V_{1,1}$-cycle is used.

| $J$ | $h_J$ | $n$ | Time[s] | ratio | $e_h$ | ratio |
|---|---|---|---|---|---|---|
| 5 | 1/128 | 49152 | 0.1680 | — | $6.06389_{-5}$ | — |
| 6 | 1/256 | 196608 | 0.7578 | 4.511 | $1.59350_{-5}$ | 0.263 |
| 7 | 1/512 | 786432 | 3.6797 | 4.856 | $4.06443_{-6}$ | 0.255 |
| 8 | 1/1024 | 3145728 | 16.4102 | 4.459 | $1.02422_{-6}$ | 0.252 |
| 9 | 1/2048 | 12582912 | 68.3516 | 4.165 | $2.56857_{-7}$ | 0.251 |
| 10 | 1/4096 | 50331648 | 276.5470 | 4.045 | $6.42940_{-8}$ | 0.250 |

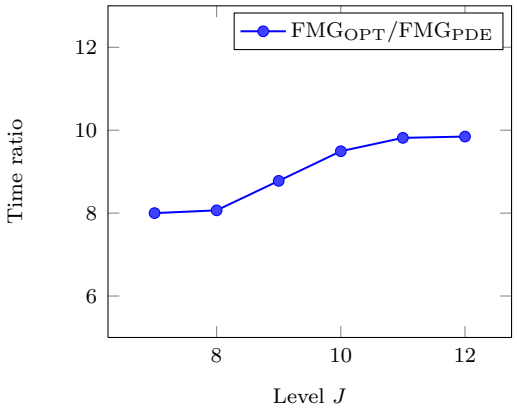



Figure 4.8: Ratio of wall-clock time in seconds for FMG/Opt and FMG/PDE.

small enough. For difficult problems it could be required to employ more iterations or a different cycle in order to ensure the convergence of FMG.[5] Table 4.2 shows the wall-clock time and the discrete $L^2$-error $e_h$ after one cycle of the FMG iteration. The discretization parameter $h_J$ is the mesh width on the finest level $J$, the total number of fine grid unknowns $n$ of the optimality system thus is given by $n = 3h_J^{-2}$. Since the number of unknowns quadruples from one level to the next, we expect a corresponding fourfold increase in computational time. This is clearly observed from the presented data, thus showing that the total cost for one FMG iteration is indeed $\mathcal{O}(n)$. The employed discretization is superconvergent with second order in the discrete $L^2$-norm. Therefore, we should expect a decrease of the error with a factor of 4, if the mesh resolution of the finest level is doubled. In the last two columns of Table 4.2 we indeed see that this is the case. After one FMG iteration, the error is of the order $\mathcal{O}(h_J^2)$. All in all, we conclude that the FMG solves the discrete optimal control problem up to discretization error accuracy with optimal complexity $\mathcal{O}(n)$.

[5]The convergence proof for FMG relies on assumptions on the reduction factor of the underlying multigrid iteration, cf. [85, 147].

In order to give a rough estimate of the computational cost needed to solve the optimality system compared to the cost needed to solve the constraint PDE only, let us now consider the cost for one application of our smoothing method in more detail. To this end, recall that $N = h_J^{-2}$ is the number of each of the three unknowns $y_J, u_J, p_J$ appearing in the discretized optimal control problem. The cost $C_N$ for the relaxation of the state and adjoint equation by (4.4) is $\mathcal{O}(N)$. Note that this is also the cost order for the smoothing iteration when solving the constraint PDE with a multigrid method. A rough (lower bound) estimate of the overall solution cost of the optimality system is as follows: From the definition of the smoother (4.27) it follows that a cost of at least $2C_N$ operations incurs by the matrix-vector product with $K_J$. Here, we have neglected all operations not involving the discretized PDE operator $L_J$, such as multiplications with the mass matrices $M_J$. The application of the preconditioner $B_{K_J}$ again contributes a cost of $2C_N$ for the constraint blocks given by $L_J$ and $L_J^T$. Finally, the conjugate gradient iteration with $\hat{H}_Z$ adds a cost of $2C_N$ for the first iteration and again a cost of $2C_N$ for the initialization of the cg method. Thus we obtain a cost count of $8C_N$ as a lower bound estimate for the total cost for one application of the smoother, counting only the operations which involve the constraint blocks $L_J$ and $L_J^T$. Figure 4.8 shows the ratio of the wall-clock times needed for the FMG solution of the optimal control problem versus the solution time of the underlying PDE problem, again using FMG. Here we see values between 8 and 10. We conclude that these values which were achieved with our implementation are more than reasonable since we have to bear in mind that the cost estimate neglects several operations such as operations involving $M_j$, inner products and the higher cost of the grid transfer operators $P^j_{j-1,\mathrm{BL}}$, $R^{j-1}_{j,\mathrm{FPA}}$ relative to their scalar counterparts. The constant of proportionality appears particularly attractive compared to the computational cost of the preconditioned GMRES method, cf. (3.30). In summary, we have seen that the discretized optimal control problem can be solved with optimal complexity at a small multiple of the cost which is required for the solution of the underlying PDE alone.

### 4.5.2 Example: General Diffusion Equation with Full Tensor

In this section we consider an optimal control problem where the equality constraints are given by the general diffusion equation

$$-\nabla \cdot D\nabla y = u, \tag{4.64}$$

completed with homogeneous Dirichlet boundary conditions. We minimize again the tracking type functional (2.1), all other parameters stay the same as before unless noted otherwise. First, we consider the diffusion tensor

$$D = \begin{bmatrix} 11 & 9 \\ 9 & 13 \end{bmatrix}. \tag{4.65}$$

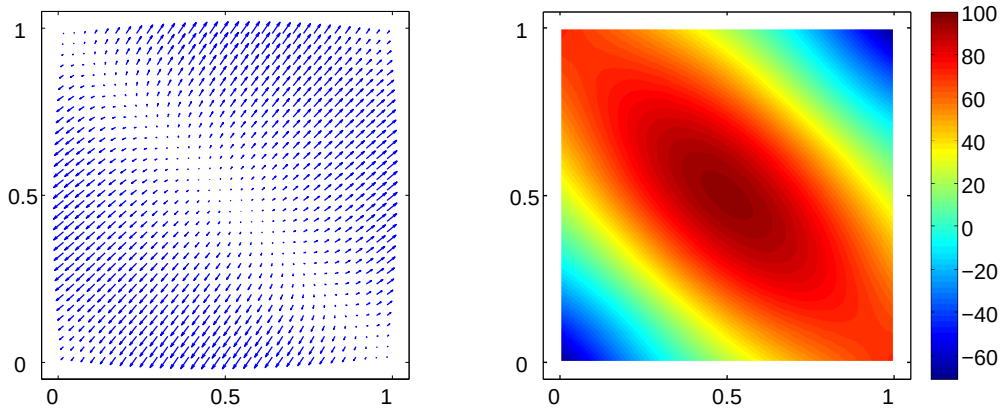



Figure 4.9: Darcy velocity $v_h$ (left) and right hand side $f_h$ (right) for the diffusion problem with the conductivity tensor $D$ given by (4.65).

Note that this choice of $D$ introduces a strong diagonal component of the flux, i.e. there are large tangential fluxes across element boundaries. In Figure 4.9 on the left we show the Darcy velocity $v_h$ which is obtained when solving the diffusion equation with the right hand side $f_h$, which is also displayed in Figure 4.9. Here, $f_h$ is the discretization of the continuous right hand side $f$ which was chosen such that the exact pressure is given by $y^* = (x_1 - x_1^2)(x_2 - x_2^2)$. The stencil resulting from the discretization of (4.64) with (4.65) is a 9-point stencil in 2 dimensions. We remark that the resulting matrix $L_h$ is not symmetric due to the implementation of the boundary conditions, cf. Section 2.3.2, however the nonsymmetry is restricted to a region close to the boundary. Furthermore, as discussed in Section 2.3.1, by following the discretize-then-optimize approach we nevertheless obtain a symmetric KKT matrix $K_h$.

It is well known that the convergence rates of multigrid methods with standard coarsening and pointwise smoothing can deteriorate when solving problems with mixed derivatives. This occurs both for seven- and nine-point discretizations. Basically two strategies can be pursued in order to improve the convergence rates: either different coarsening strategies, such as semi-coarsening or algebraic multigrid are employed, in which case a cheap pointwise smoother can be applied, or the standard coarsening approach with a more robust smoothing iteration has to be employed. Here, we follow the latter approach. Smoothers which are particularly well-suited for problems with mixed derivatives include those based on incomplete LU-factorizations, since they reduce also some lower frequency error components which are not captured well by the coarse grid correction. Here, we use the ILU(0) method with zero fill-in as constraint smoother, i.e. as the preconditioner in the iteration (4.4). ILU-based smoothers have been successfully applied as robust smoothers for anisotropic problems, in particular in fluid flow applications. For their derivation and further details we refer to [154].

For the optimal control test problem we set the right hand side $b = (\bar{y}, 0, f)$ such that

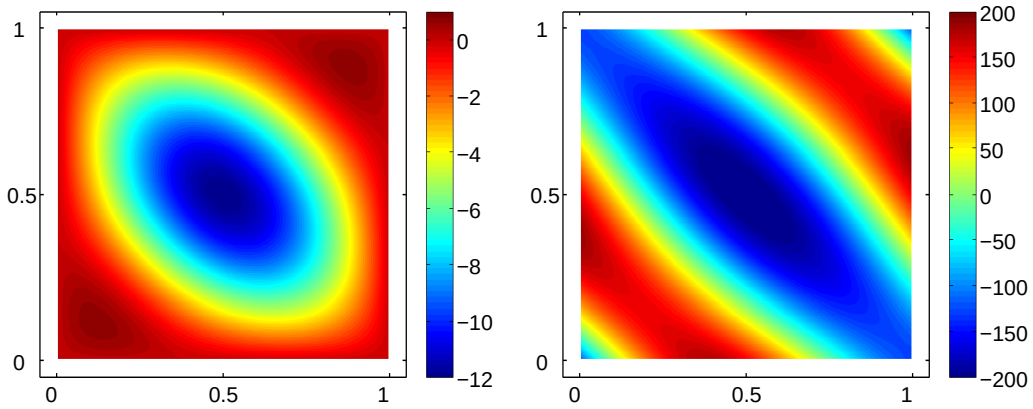



Figure 4.10: Computed optimal control $u_h$ (left) and discretized target state $\bar{y}_h$ (right) for the optimal control problem with constraints given by (4.64) and (4.65), shown for a finest mesh with $h = 2^{-8}$.

the exact solution $y^*$ of the state equation is the same as above for the scalar problem. To this end, we set $f = -\nabla \cdot D\nabla y^* - u^*$ with $u^* = \big(-\nabla \cdot D\nabla y^*\big)\big(\sin(\pi x_1)\sin(\pi x_2)\big)$. From the optimality condition follows $p^* = \sigma u^*$ and we define the target state $\bar{y} = y^* - \nabla \cdot D\nabla p^*$. Figure 4.10 shows color plots of the computed approximation $u_J$ to the optimal control $u^*$ (on the left) and the discretization of $\bar{y}$ (on the right). Both approximations are computed on a grid with mesh size $h_J = 2^{-8}$.

For the multigrid solution of the optimal control problem we employ (4.27) with $\alpha = \beta = 1$ and test the $V_{1,1}$- as well as the $W_{1,1}$-cycle, both with GS-LEX and ILU(0) as constraint smoother. For comparison, we solved the underlying PDE problem (4.64), (4.65) with multigrid, again using the $V_{1,1}$- and the $W_{1,1}$-cycle both with GS-LEX and ILU(0) smoothing. The multigrid iteration was stopped as soon as $\|r_J\| \leq 10_{-12}$. In some cases the stopping criterion could not be satisfied within the maximum number of 50 iterations, instead the residual norm stagnated with $\|r_J^m\| \sim \mathcal{O}(1_{-12})$. This happened shortly after the algebraic error had reached the level of discretization accuracy and a further decrease of the error would be smaller than the relative precision $\varepsilon_{\text{mach}} \sim 1.1_{-16}$ of the double precision floating point calculations, rendering the error constant. In Table 4.3 we give the reduction factors $\varrho^m$ for levels $J = 5, \ldots, 10$. As before, we set $h_c = 1/4$ such that $h_J = 2^{-J+2}$. Since the iterations with ILU smoothing satisfied the stopping criterion in considerably less iterations than with GS-LEX smoothing, in this case we state $\varrho^{10}$. For GS-LEX smoothing, we give $\varrho^{20}$ as before. The results obtained for the optimal control problem are denoted by "KKT", the reference results of the constraint PDE are denoted by "PDE". We observe that, similar to the case of the model problem, we obtain reduction factors when solving the KKT system which are in very good agreement with those for the multigrid solution of the underlying PDE. This holds regardless of the employed smoothing

Table 4.3: Reduction factors $\varrho^{10}$ (ILU(0)) and $\varrho^{20}$ (GS-LEX) of the residual norm for the scalar PDE-only problem and the optimal control problem with constraints (4.64), (4.65).

| | | | $J$ | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 5 | 6 | 7 | 8 | 9 | 10 |
| $V_{1,1}$ | GS-LEX | PDE | $4.24_{-1}$ | $4.32_{-1}$ | $4.36_{-1}$ | $4.35_{-1}$ | $4.38_{-1}$ | $4.38_{-1}$ |
| | | KKT | $4.32_{-1}$ | $4.34_{-1}$ | $4.42_{-1}$ | $4.37_{-1}$ | $4.39_{-1}$ | $4.40_{-1}$ |
| $W_{1,1}$ | GS-LEX | PDE | $3.22_{-1}$ | $3.26_{-1}$ | $3.27_{-1}$ | $3.28_{-1}$ | $3.28_{-1}$ | $3.29_{-1}$ |
| | | KKT | $3.17_{-1}$ | $3.28_{-1}$ | $3.29_{-1}$ | $3.29_{-1}$ | $3.29_{-1}$ | $3.29_{-1}$ |
| $V_{1,1}$ | ILU | PDE | $3.99_{-2}$ | $5.15_{-2}$ | $5.65_{-2}$ | $5.81_{-2}$ | $5.92-2$ | $5.96_{-2}$ |
| | | KKT | $4.40_{-2}$ | $5.65_{-2}$ | $5.84_{-2}$ | $6.01_{-2}$ | $6.14_{-2}$ | $6.23_{-2}$ |
| $W_{1,1}$ | ILU | PDE | $3.66_{-2}$ | $4.32_{-2}$ | $4.31_{-2}$ | $4.37_{-2}$ | $4.40_{-2}$ | $4.39_{-2}$ |
| | | KKT | $4.01_{-2}$ | $4.18_{-2}$ | $4.43_{-2}$ | $4.48_{-2}$ | $4.54_{-2}$ | $4.58_{-2}$ |

iteration for the constraint PDE and demonstrates once more that a robust smoother for the constraint PDE is the optimal candidate for the constraint smoothing step within $\mathcal{S}^{\nu}_{j,\alpha,\beta}$. For both the KKT and the PDE problem, the reduction factors $\varrho^{10}$ obtained with ILU-smoothing are better by almost an order of magnitude, compared to $\varrho^{20}$ obtained with GS-LEX smoothing. Note that the gain in convergence speed of the $W_{1,1}$-cycle over the $V_{1,1}$-cycle is minute and even more so when ILU-smoothing is employed. Figure 4.11 shows the convergence histories for the optimal control problem computed on level $J = 10$ with respect to different quantities. Starting from top left, in clockwise direction we show the reduction of the 2-norm of the residual $\|r_J^m\|$ and the discrete $L^2$-errors with respect to the solution of the state equation $e_J^{y,m}$, the adjoint unknown $e_J^{p,m}$ and the control unknown $e_J^{u,m}$. We note that switching from GS-LEX smoothing to ILU(0) reduces the number of iterations required to reach the discretization accuracy roughly by a factor of 2.5. For example for the $W_{1,1}$-cycle, the error $e_J^{y,m}$ reaches the level of discretization accuracy for $m = 6$ in the case of ILU(0)-smoothing and for $m = 15$ when using GS-LEX smoothing. The results with respect to the other errors are similar. In agreement with the data from Table 4.3 the improvement is somewhat larger for the $V_{1,1}$-cycle, where we have $m = 7$ vs. $m = 21$. Due to the very good smoothing properties of the ILU(0)-method for this problem the convergence rate of the $V_{1,1}$-cycle is already close to the two-grid convergence factor and thus can not be improved much by the $W_{1,1}$-cycle. For this problem clearly the cheaper $V_{1,1}$-cycle is more efficient than the $W_{1,1}$-cycle. This can also be seen in Figure 4.12 on the left. There, for the tested cycles the residual reduction is plotted over the execution time measured in seconds. In both cases, the slightly higher cost per iteration for the ILU(0)-smoothing compared to the cheaper GS-LEX smoothing pays off due to the large reduction of the convergence factor.

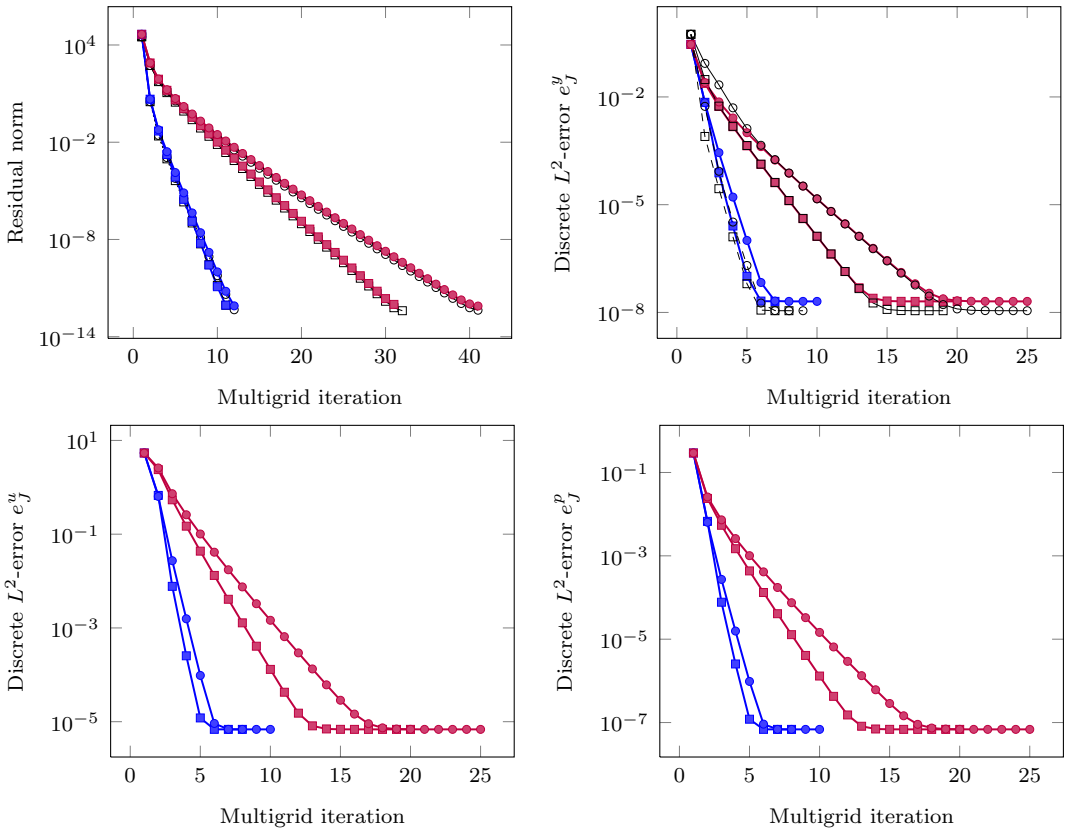



Figure 4.11: Clockwise from top left: reduction of the residual norm $\|r_J^m\|$, the state error $e_J^{y,m}$, the adjoint error $e_J^{p,m}$ and the control error $e_J^{u,m}$, $J = 10$. Shown are results for $V_{1,1}$- and $W_{1,1}$-cycles with GS-LEX as constraint smoother, $V_{1,1}$- and $W_{1,1}$-cycles with ILU(0) as constraint smoother. For reference, we also show results for multigrid applied to (4.64), (4.65) with $V_{1,1}$- and $W_{1,1}$-cycles with GS-LEX smoothing and $V_{1,1}$- and $W_{1,1}$-cycles with ILU(0)-smoothing.

In Figure 4.12 to the right we show the reduction of the total discrete $L^2$-error $e_J^m$ computed with the $V_{1,1}$-cycle and ILU(0) constraint smoothing for different fine grid mesh sizes $h_J$. We observe that the rate of convergence is independent of the number of levels $J$ and thus $h_J$. The final attained error $e_J^{\bar{m}}$ equals the discretization error, which we expect to decrease quadratically with respect to $h_J$. This is confirmed in Figure 4.13. There, we show the error $e_J^{\bar{m}}$ for the final multigrid iteration $\bar{m}$ plotted over the total number of unknowns which is given by $n_J = 3N_J$. Since in two dimensions $N_J = h_J^{-2}$, a slope of -1 corresponds to the expected second order convergence in the discrete $L^2$-norm.

Let us briefly consider another example, namely the homogeneous optimal control

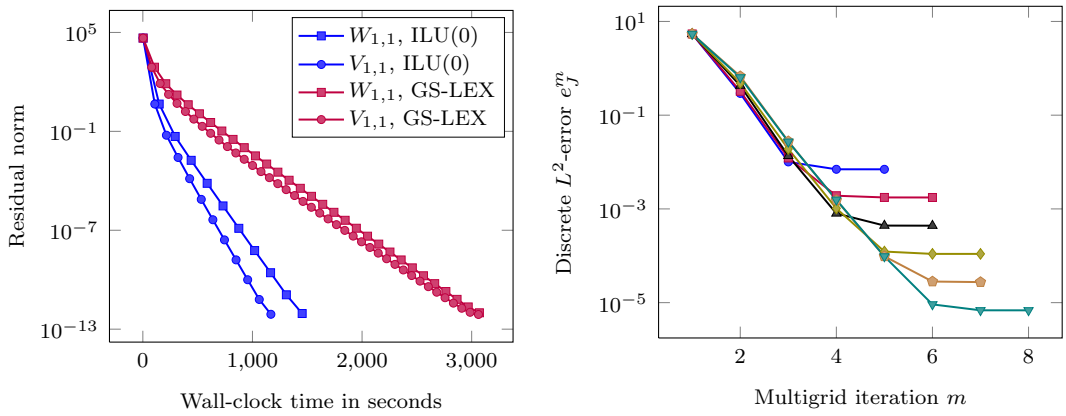



Figure 4.12: Left: Wall clock times for different multigrid cycles and choices of constraint smoothing, level $J = 10$. Right: Discrete $L^2$-error $e_J^m$ obtained with the $V_{1,1}$-cycle and ILU(0) for constraint smoothing, levels $J = 5, \ldots, 10$.

Table 4.4: Reduction factors $\varrho^{10}$ (ILU(0)) and $\varrho^{20}$ (GS-LEX) of the residual norm for the scalar PDE-only problem and the optimal control problem with constraints (4.64), (4.66).

| | | $J$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | 5 | 6 | 7 | 8 | 9 | 10 |
| GS-LEX | PDE | $3.98_{-1}$ | $4.26_{-1}$ | $4.30_{-1}$ | $4.38_{-1}$ | $4.43_{-1}$ | $4.43_{-1}$ |
| GS-LEX | KKT | $4.07_{-1}$ | $4.27_{-1}$ | $4.35_{-1}$ | $4.42_{-1}$ | $4.49_{-1}$ | $4.50_{-1}$ |
| ILU(0) | PDE | $6.46_{-2}$ | $6.81_{-2}$ | $7.83_{-2}$ | $7.66_{-2}$ | $7.99_{-2}$ | $8.12_{-2}$ |
| ILU(0) | KKT | $6.60_{-2}$ | $6.73_{-2}$ | $7.79_{-2}$ | $7.65_{-2}$ | $7.99_{-2}$ | $8.13_{-2}$ |

problem with (4.64) as constraints, but this time with the varying diffusion coefficient

$$D = \begin{bmatrix} (x_1+2)^2 + y^2 & \sin(x_1 x_2) \\ \sin(x_1 x_2) & 1 \end{bmatrix}. \tag{4.66}$$

Due to the results obtained for the previous example concerning the efficiency of $V$- and $W$-cycles, here we only employ the $V_{1,1}$-cycle, both with GS-LEX and ILU(0) as constraint smoothing. Again we computed the scalar PDE-only problem for comparison. The reduction factors $\varrho^{10}, \varrho^{20}$ with respect to $\|r_J\|$ are given in Table 4.4. Also for this problem we see a good agreement of the reduction factors for the optimal control problem and the underlying PDE problem. The reduction factors for multigrid iterations with GS-LEX smoothing are similar to those of the previous example. As before, the robust ILU(0)-smoothing yields a considerable improvement and overall excellent reduction factors are obtained. In Figure 4.14 we show a surface plot of the control error $u_J$ as computed after 15 iterations (in which the $V_{1,1}$-ILU cycle satis-

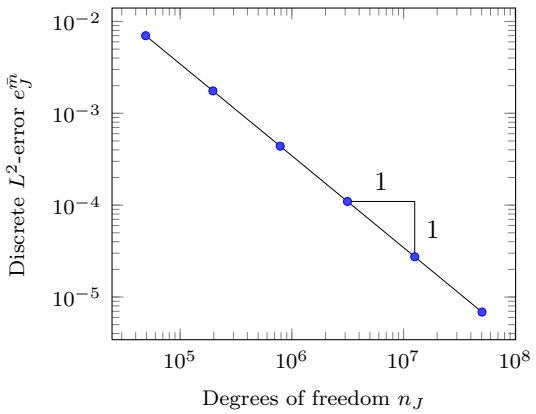



Figure 4.13: Final discrete $L^2$-error $e_J^{\bar{m}}$ obtained with the $V_{1,1}$-cycle and ILU(0) for constraint smoothing plotted over the number of degrees of freedom $n_J = 3h_J^{-2}$ for levels $J = 5, \ldots, 10$.

fied the stopping criterion), on the left for GS-LEX smoothing and on the right with ILU(0) smoothing. In both cases, the error exhibits the characteristic features of the error for anisotropic diffusion problems, namely slowly varying in one space direction and more oscillatory in the other one. The effect is much more pronounced though in the GS-LEX case since the Gauss-Seidel iteration is not effective in reducing the error along its slowly varying directions. Noting in addition the magnitude of both errors, the superior smoothing property of the ILU(0)-smoother is obvious.

Finally we point out that this test problem highlights the versatility offered by our approach. As we have seen in the case of the model problem, the multigrid solution of the KKT system is achieved with the same convergence speed as that for the underlying constraint-PDE allows. The results presented here show that this is also the case if the underlying PDE requires more robust smoothing iterations in order to obtain optimal reduction factors. Simply by employing the appropriate smoothing iteration as constraint smoother we can obtain the optimal convergence speed. As far as we know, no other published full space multigrid method for optimal control problems exhibits the same feature. In particular, is is not obvious how ILU-smoothing could be included in the collective smoothing approach CGSM presented in [33, 35].

### 4.5.3 Example: Anisotropic Diffusion Equation on Non-Uniform Grids

In this section we consider optimal control problems where the constraints (4.64) are discretized on non-uniform meshes. This is another example for a situation in which standard multigrid with pointwise smoothing fails already for the solution of the constraint PDE.

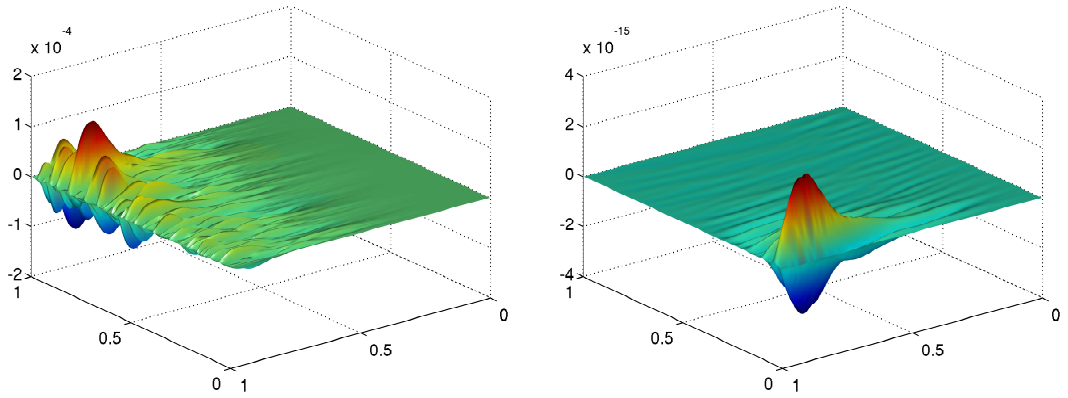



Figure 4.14: Error of the control component $u_J$ for $J = 6$ after 15 iterations of the $V_{1,1}$-cycle with GS-LEX as constraint smoother (left) and ILU(0) (right).

The discretization of the diffusion equation based on the method of Section 2.3.2 on non-uniform grids results in a transformation $\mathcal{D}$ of the diffusion tensor $D$ which is full, regardless whether $D$ is diagonal or not. Furthermore, the transformed tensor $\mathcal{D}$ introduces strong anisotropies into the discrete operator[6] which are solely due to the mapping $\phi : \hat{\Omega} \to \Omega$, therefore for our purpose here it is sufficient to consider $D = I$. Again, similar to the setting in the previous section, two possibilities exist in order to establish linear multigrid convergence: semi-coarsening or algebraic coarsening in conjunction with pointwise smoothing or standard coarsening in combination with a smoothing method that introduces a stronger coupling of the unknowns. In addition to the ILU(0)-smoothing we will consider line-relaxation as constraint smoother. In particular, we employ the alternating line Gauss-Seidel (ALGS) method. The ALGS relaxation can be considered as a specific block Gauss-Seidel method. For the lexicographical unknown ordering, one block consists of a single line of unknowns in either horizontal or vertical direction. The block Gauss-Seidel iterates sequentially over the blocks, solving for all unknowns within a block simultaneously. Here, for each block of unknowns a tridiagonal system has to be solved. This can be done efficiently with the Thomas algorithm. The ALGS performs horizontal and vertical sweeps in an alternating fashion. A similar comment concerning ILU-constraint smoothing and the CGSM is in order here. Although one can think of extending the collective smoothing to a line-relaxation, one has to consider that the resulting systems are not tridiagonal anymore but can have considerable bandwidth. Thus, a cheap method such as Thomas' algorithm can not be used. Instead, probably one has to resort to one-dimensional

[6]For example, a standard discretization of $-\Delta$ on a mesh with spacing $\epsilon h, \epsilon \ll 1$ in one coordinate direction and $h$ in the other direction yields the same discrete operator as discretizing $-\nabla \cdot D\nabla$ on a uniform mesh with $D = \begin{pmatrix} 1 & 0 \\ 0 & \epsilon \end{pmatrix}$.

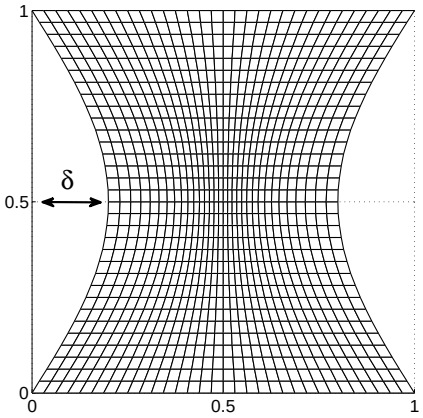
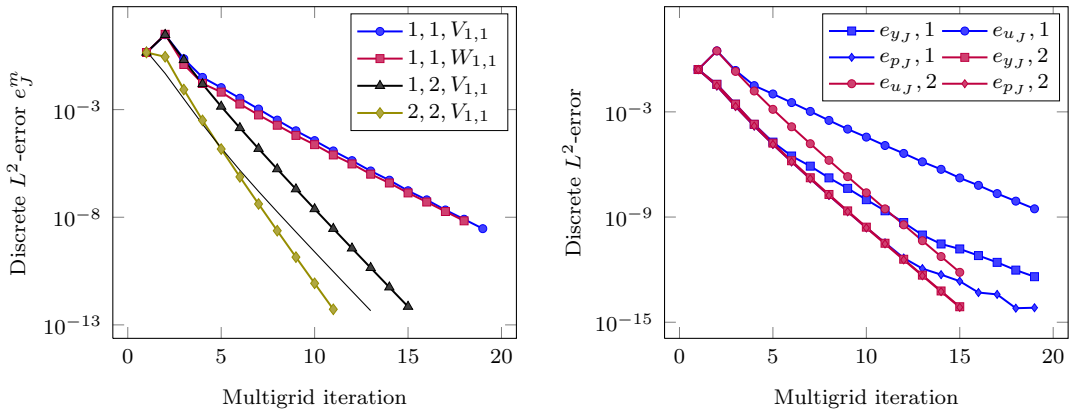


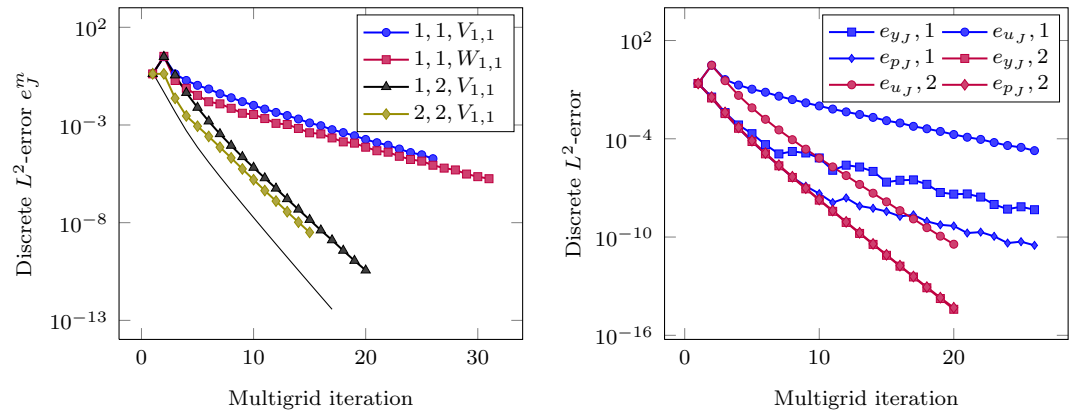

Figure 4.15: Mesh $G_\delta$ with indentation parameter $\delta = 0.25$, $h = 2^{-5}$.

multigrid methods in order to solve the subsystems with optimal complexity. All in all, the smoothing step is likely to require a significant computational effort.

As a first example we consider the homogeneous optimal control problem discretized on a deformation of the unit square with a contraction in horizontal direction. The deviation from the unit square boundary on each side is given by a parameter $\delta$. The resulting mesh $G_\delta$ for $\delta = 0.25$ is shown in Figure 4.15. We consider three different cases: $\delta = 0.05, 0.15$ and $0.25$. The test cases considered so far have been well-behaved in the sense that the smoothing step with respect to $u_j$ has not been a limiting factor for the convergence speed and thus essentially for the KKT problem the same reduction factors were obtained as for the PDE-only problem. In other words, the reduction with respect to $y_j$ and $p_j$ dominated the iteration. This behavior changes depending on the value of the contraction $\delta$. In Figure 4.16 on the left we show the reduction of the total discrete $L^2$-error $e_J^m$ obtained for $J = 10$. Different values $\alpha, \beta$ for the smoothing iteration $\mathcal{S}_{j,\alpha,\beta}^\nu$ have been employed. We observe that for $\alpha = \beta = 1$ the convergence speed for the KKT system is significantly lower than for the corresponding PDE-only problem, which is indicated by —). The $W$-cycle does not improve the rate of convergence. However increasing the value of $\beta$ to 2 restores the optimal rate and increasing $\alpha$ yields additional improvement. The reason for this becomes apparent when looking at the right part of Figure 4.16. There, we show the reduction of the individual error components $e_{y_J}^m, e_{u_J}^m$ and $e_{p_J}^m$ corresponding to the $V$-cycle with $\alpha = \beta = 1$ and $\alpha = 1, \beta = 2$. In particular in the case of $\delta = 0.25$ it is noticeable that for $\beta = 1$ in the first few iterations $e_{y_J}^m$ and $e_{p_J}^m$ decrease faster than $e_{u_J}^m$ but eventually are limited in convergence and pick up the lower rate of $e_{u_J}^m$. Increasing $\beta$ to 2 improves the smoothing effect for (RHA), cf. Algorithm 5, and restores the near optimal convergence speed close to that of the underlying PDE problem. In Table 4.5 we give the average reduction factors $\varrho_{\text{avg}}$ with respect to $\|r_J\|$ for levels $J = 5, \ldots, 10$. For all cases we employed the $V_{1,1}$-cycle. The different values of $\alpha, \beta$ are noted in the




(a) $\delta = 0.15$.




(b) $\delta = 0.25$.

Figure 4.16: Reduction of discrete $L^2$-errors for the optimal control problem discretized on $G_\delta$ with $\delta = 0.15, 0.25$ and smoothing iteration $\mathcal{S}^\nu_{j,\alpha,\beta}$. Left: reduction of total error $e_J^m$, — is the PDE-only reference, $\alpha, \beta \in \{1, 2\}$. Right: error components $e^m_{y_J}, e^m_{u_J}$ and $e^m_{p_J}$, $\alpha = 1, \beta = 1, 2$.

second column. As expected, for increasing $\delta$, the convergence speed degrades, both for the scalar PDE-only problem and the KKT problem. Still, the reduction factors are more than acceptable. We remark that a different type of cycle does not yield significant better results, e.g. for the PDE-only problem for $\delta = 0.25$ the $W_{1,1}$-cycle drops the average reduction factor on level $J = 10$ from 0.186 to 0.136 and the higher cost compared to the $V_{1,1}$-cycle does not pay off. For the easiest case $\delta = 0.05$, the reduction factors are very good when employing ALGS as constraint smoother and do not deviate too much from the reference case $\delta = 0$ when employing just the pointwise GS-LEX. In all cases $\varrho_{\text{avg}}$ is bounded independent of $J$. A further increase of $\alpha, \beta$ would yield even better reduction rates but already for $\alpha = 1, \beta = 2$, the optimal rates of the PDE-only reference are obtained for the most difficult case $\delta = 0.25$. We point

Table 4.5: Reduction factors $\varrho_{\text{avg}}$ obtained for the mesh deformations $\delta = 0.05, 0.15$ and 0.25. In all cases a $V_{1,1}$-cycle with the smoother $\mathcal{S}_{j,\alpha,\beta}$ was used. For comparison also the rates of the PDE-only problem are given.

| | | $J$ | | | | | |
|---|---|---|---|---|---|---|---|
| $\delta$ | $\alpha,\beta$ | 5 | 6 | 7 | 8 | 9 | 10 |
| 0.05 | 1,1[a] | $9.84_{-2}$ | $1.01_{-1}$ | $1.02_{-1}$ | $1.02_{-1}$ | $1.04_{-1}$ | $1.04_{-1}$ |
| | 1,1[b] | $5.06_{-2}$ | $5.22_{-2}$ | $5.19_{-2}$ | $5.16_{-2}$ | $5.18_{-2}$ | $5.33_{-2}$ |
| | —[c] | $4.86_{-2}$ | $5.03_{-2}$ | $5.21_{-2}$ | $5.25_{-2}$ | $5.41_{-2}$ | $5.44_{-2}$ |
| 0.15 | 1,1[a] | $6.73_{-1}$ | $5.03_{-2}$ | $5.21_{-2}$ | $5.25_{-2}$ | $5.41_{-2}$ | $5.44_{-2}$ |
| | 1,2[b] | $7.24_{-2}$ | $8.34_{-2}$ | $9.21_{-2}$ | $9.21_{-2}$ | $9.68_{-2}$ | $9.88_{-2}$ |
| | —[c] | $7.17_{-2}$ | $8.20_{-2}$ | $9.22_{-2}$ | $9.49_{-2}$ | $9.83_{-2}$ | $1.02_{-1}$ |
| 0.25 | 1,1[a] | $6.73_{-1}$ | $6.73_{-1}$ | $6.77_{-1}$ | $6.75_{-1}$ | $6.72_{-1}$ | $6.49_{-1}$ |
| | 1,2[b] | $1.93_{-1}$ | $1.78_{-1}$ | $1.71_{-1}$ | $1.67_{-1}$ | $1.76_{-1}$ | $1.82_{-1}$ |
| | —[c] | $1.12_{-1}$ | $1.41_{-1}$ | $1.62_{-1}$ | $1.71_{-1}$ | $1.79_{-1}$ | $1.86_{-1}$ |

[a]Constraint smoothing with pointwise GS-LEX
[b]Constraint smoothing with ILU(0)
[c]Reference computation of PDE-only problem

out again that just the cheap $V_{1,1}$-cycle was used in all cases.

Let us now consider two more mappings $\phi : \hat{\Omega} \to \Omega$ which generate deformations in both coordinate directions. The first deformation is given by

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \phi_1 \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} \hat{x}_1 + 0.1 \sin(2\pi\hat{x}_1)\sin(2\pi\hat{x}_2) \\ \hat{x}_2 + 0.1 \sin(2\pi\hat{x}_1)\sin(2\pi\hat{x}_2) \end{pmatrix}, \tag{4.67}$$

the second one by

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \phi_2 \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} \hat{x}_1 + 0.1 \cos(3\hat{x}_2) \\ \hat{x}_2 + 0.1 \sin(6\hat{x}_1) \end{pmatrix}. \tag{4.68}$$

The grids generated by $\phi_1$ and $\phi_2$ for $h = 1/32$ are shown in Figure 4.17. The more difficult problem is presented by $\phi_1$ due to larger variations of the local grid size. The ratio $\max|T_i| \,/\, \min|T_i|$ is about 4.4, whereas for $\phi_2$ we obtain a value of roughly 1.44. This manifests itself in the convergence rates, which are given in Table 4.6. Due to the findings for the first problem in this section we set $\alpha = 1, \beta = 2$ and employed the $V_{1,1}$-cycle. For $\phi_1$ we tested both ILU(0) and ALGS as constraint smoother, for $\phi_2$ only the results for ILU(0) are given. In both cases, the rates for ILU(0) are excellent and again the rates for the KKT problem are close to those of the PDE problem. The reduction factors are considerably worse for ALGS smoothing in both cases, but still match very good.

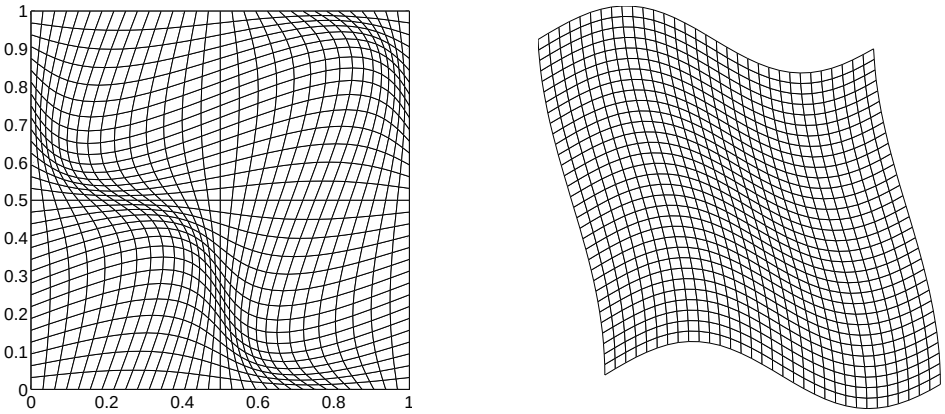



Figure 4.17: Grids generated by mappings $\phi_1$ (left) and $\phi_2$ (right) for $h = 1/32$.

Table 4.6: Reduction factors $\varrho_{\text{avg}}$ obtained for the meshes generated by the mappings $\phi_1$ and $\phi_2$. In all cases a $V_{1,1}$-cycle with the smoother $\mathcal{S}_{j,1,2}$ was used.

| | | | $J$ | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 5 | 6 | 7 | 8 | 9 | 10 |
| $\phi_1$ | ALGS | KKT | $2.53_{-1}$ | $3.29_{-1}$ | $3.86_{-1}$ | $4.22_{-1}$ | $4.44_{-1}$ | $4.51_{-1}$ |
| | | PDE | $2.44_{-1}$ | $3.25_{-1}$ | $3.89_{-1}$ | $3.93_{-1}$ | $4.11_{-1}$ | $4.17_{-1}$ |
| | ILU(0) | KKT | $1.91_{-1}$ | $1.61_{-1}$ | $1.51_{-1}$ | $1.30_{-1}$ | $1.18_{-1}$ | $1.07_{-1}$ |
| | | PDE | $8.81_{-2}$ | $9.30_{-2}$ | $1.01_{-1}$ | $1.03_{-1}$ | $1.04_{-1}$ | $1.05_{-1}$ |
| $\phi_2$ | ILU(0) | KKT | $3.43_{-2}$ | $4.02_{-2}$ | $5.18_{-2}$ | $6.04_{-2}$ | $6.12_{-2}$ | $6.83_{-2}$ |
| | | PDE | $2.99_{-2}$ | $3.89_{-2}$ | $5.24_{-2}$ | $6.15_{-2}$ | $6.26_{-2}$ | $6.99_{-2}$ |

The results of this section show that our overall method easily allows to exploit the knowledge of sophisticated multigrid solution methods for the constraint PDE by adapting the smoothing iteration for the inner linear systems in a suitable way. In this sense, again the result of the smoothing analysis performed in Section 4.3 is corroborated by the numerical results. Furthermore, the same comment as at the end of the previous section applies: although it is in principle possible to extend the CGSM approach to include line-smoothing methods, the resulting systems are considerably more expensive to solve, in particular they are not tridiagonal anymore. Thus, a cheap $\mathcal{O}(N)$ method like the Thomas algorithm is not available but multigrid itself should be applied. We remark that to our knowledge in previous publications the full space multigrid approach has been applied only to the model problem. Additional difficulties such as non-uniform meshes and the resulting anisotropies have not been considered before. For more difficult problems such as discontinuous coefficients or nonsmooth mappings $\phi$, we can not expect acceptable convergence rates with the presented approaches, due to the difficulties of standard multigrid for the PDE-only

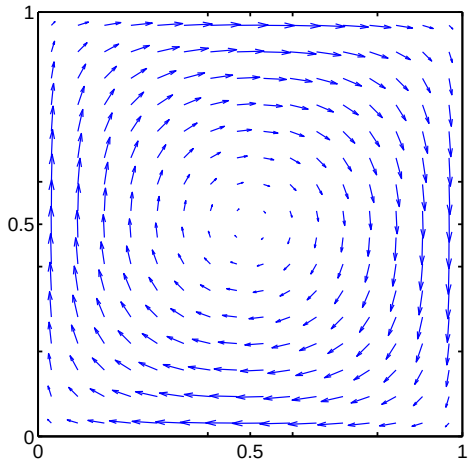



Figure 4.18: The circular convective velocity field $w_h$ on a mesh with $h = 2^{-5}$.

problem. For these cases, techniques such as operator-dependent interpolations or operator-dependent coarsening (viz. algebraic multigrid) are required. We expect that employing the appropriate techniques for the underlying PDE problem within our approach would yield again satisfactory convergence rates for the multigrid solution of the KKT problem, although this has not been confirmed yet by numerical experiments and applying algebraic multigrid to KKT systems is still an open question.

### 4.5.4 Example: Convection-Diffusion with Circular Wind

In this section we study optimal control of the steady-state convection-diffusion equation

$$-\varepsilon\Delta y + w \cdot \nabla y = u \tag{4.69}$$

with homogeneous Dirichlet boundary conditions and a given convection velocity or "wind" $w$. Here we consider the circular wind

$$w_1(x_1, x_2) = 8(x_2 - 0.5)(0.25 - (x_1 - 0.5)^2) \tag{4.70}$$

$$w_2(x_1, x_2) = -8(x_1 - 0.5)(0.25 - (x_2 - 0.5)^2), \tag{4.71}$$

see Figure 4.18. It is well-known (see e.g. [134]) that discretization of the convective term in the singular perturbed equation (4.69) with central finite differences or standard finite elements yields unstable approximations on coarse meshes. In particular, the matrix representing the discrete operator looses the $M$-matrix property if

$$\frac{h}{\varepsilon} \max_{\Omega_h} \|w\|_\infty > 2, \tag{4.72}$$

where the left side in (4.72) is the mesh-Péclet number $Pe$. Many practical problems are dominated by convection, i.e. $\varepsilon \ll 1$, and consequently it is difficult to satisfy

$Pe < 2$ on coarse meshes. For multigrid solvers, unacceptably high resolutions for the coarse meshes would be required in order to avoid instabilities due to the coarse grid correction or smoothing on coarse meshes. Therefore, upwind difference schemes or stabilized upwind Galerkin/streamline diffusion elements are the methods of choice for the discretization of (4.69). Here, we apply a simple first-order upwind scheme and just remark that for practical problems the implied reduction to first-order accuracy $\mathcal{O}(h)$ due to the large numerical viscosity might not be appropriate to obtain accurate solutions. In these cases, higher order upwind schemes have to be employed [134]. In any case, the upwind discretization results in a nonsymmetric matrix and we remind the reader of Section 2.3.1. Here, we employ the discretize-then-optimize scheme, which means that the discrete adjoint operator is given by $L_h^T$ and thus depends on the upwind scheme leading to $L_h$. However, the resulting KKT matrix is symmetric and discrete gradients are consistent.

Before we embark on the optimal control problem, some comments on the multigrid solution of the convection-diffusion equation are in order. Convection-dominated problems are in general classified as "difficult" problems to solve and as such require specifically tailored multigrid components in order to achieve acceptable convergence rates. This impacts all components of the multigrid solver. The first issue is the choice of the smoothing iteration. For a constant wind in the horizontal direction and for $\varepsilon \to 0$, the lexicographical Gauss-Seidel iteration degenerates to an exact solver due to the fact that the upwind operator becomes essentially tridiagonal. ILU-decompositions behave in a similar way. This observation spurred the idea of downstream node ordering, such that pointwise Gauss-Seidel smoothers are close to an exact solver along the characteristics of the flow. For a general wind with possibly closed characteristics downwind ordering is a complex problem and it is more common to employ four-direction point Gauss-Seidel, line smoothers or alternating modified ILU. Here, we chose the alternating line Gauss-Seidel method as constraint smoother of which we remark that it is not completely robust in $h$ and $\varepsilon$ for circular convection.

A further problem which is not so obvious is the coarse grid correction. In fact, a two-grid local mode analysis shows that for first-order upwind discretization and standard coarsening, the two-grid convergence factor is limited by 0.5 for $\varepsilon \to 0$, regardless of the number of smoothing steps.[7] Consequently, even worse multigrid convergence factors have to be expected for the standard approach. Although this can be considered as a worst-case scenario which is not observed in all practical applications, it is indeed observed for convection-dominated recirculating flows. A variety of modifications, none of which works equally well for all problems, has been proposed in the literature: Galerkin coarse grid operators constructed with operator-dependent transfer [56] or upwind-interpolation operators [54], acceleration by Krylov methods [128]

[7] Due to a grid-dependent amount of artificial viscosity, smooth modes which are constant along the characteristics are approximated only with zero relative order by the coarse grid operator [147, 156].

Table 4.7: Average residual reduction factors $\varrho_{\text{avg}}$ obtained for the optimal control of circular convection. The smoothing iteration is $\mathcal{S}_{j,1,\beta}$. For $\varepsilon = 1_{-4}, 1_{-5}$ a $W_{1,1}$-cycle is used as preconditioner for flexible GMRES(20), whereas for $\varepsilon = 1_{-1}, 1_{-2}$ and $1_{-3}$ the multigrid cycle is employed as solver.

| | | $J$ | | | |
|---|---|---|---|---|---|
| $\varepsilon$ | $\beta$ | 5 | 6 | 7 | 8 |
| $1_{-1}$ | 2 | $3.76_{-2}$ | $3.65_{-2}$ | $3.61_{-2}$ | $3.58_{-2}$ |
| $1_{-2}$ | | $6.48_{-2}$ | $5.43_{-2}$ | $4.22_{-2}$ | $4.06_{-2}$ |
| $1_{-3}$ | | $1.50_{-1}$ | $1.53_{-1}$ | $1.20_{-1}$ | $1.05_{-1}$ |
| $1_{-4}$ | 4 | $3.21_{-1}$ | $2.98_{-1}$ | $2.77_{-1}$ | $3.13_{-1}$ |
| $1_{-5}$ | 5 | $5.58_{-1}$ | $3.87_{-1}$ | $3.27_{-1}$ | $3.64_{-1}$ |

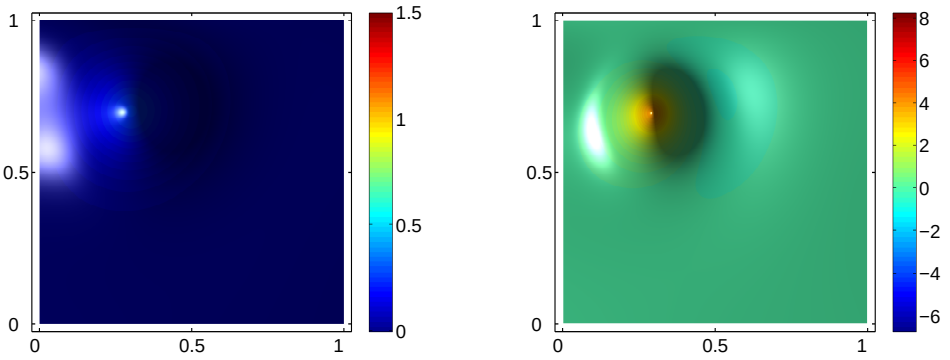
and employing $W$-cycles is common. A uniformly convergent method without restrictions on $h_0$ has been proposed in [103]. There, a monotone discretization is employed within a multigrid preconditioner for the standard second-order Galerkin finite element discretization. The first-order scheme is solved with a multigrid method using crosswind reordering and the whole scheme is employed as preconditioner for GMRES. Further it is shown that all components are required in order to obtain uniform convergence without any constraints on the coarse grid size. Here, we consider a $W$-cycle as preconditioner for GMRES, since this is easily accessible in our framework.

Let us now return to the optimal control problem. To this end, we consider the tracking problem with the Gaussian target state

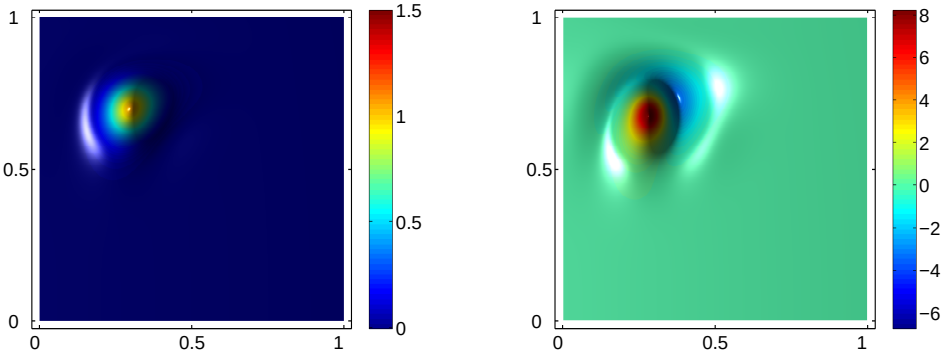
$$\bar{y}(x_1, x_2) = 2.5 \exp(-(400(x - 0.3)^2 + 250(y - 0.7)^2)) \tag{4.73}$$

and with a decreasing sequence of parameters $\varepsilon$. The regularization parameter is fixed at $\sigma = 1_{-2}$. In Figure 4.19 we show the computed optimal controls $u_h$ and associated optimal states $y_h$ for $\varepsilon = 1_{-l}, l = 1, \ldots, 4$ on a mesh with $h = 2^{-8}$. The tracking "works against" convection and diffusion and we clearly observe the stronger influence of the convection term on $u_h$ and $y_h$ for decreasing $\varepsilon$, while at the same time the diffusive effects are less pronounced. Results for $\varepsilon = 1_{-5}$ are visually indistinguishable from those for $\varepsilon = 1_{-4}$ and thus are not shown.

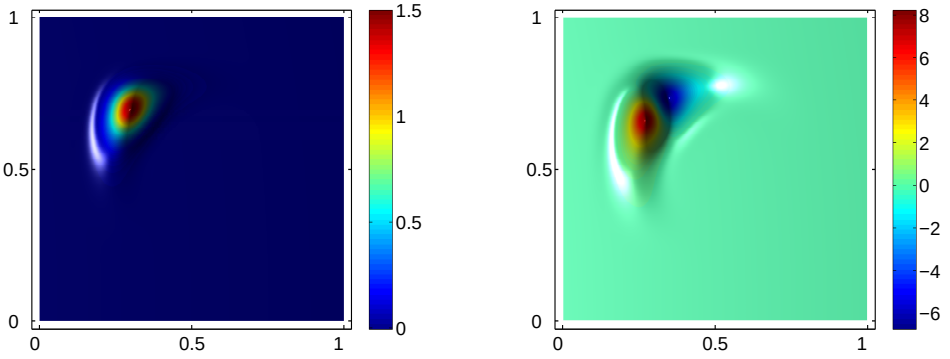
For our computations, we use the coarsest mesh size $h_0 = 1/16$, thus the finest mesh for level $J$ has a mesh size of $h_J = 2^{-J+4}$. We give the results for $J = 4, \ldots, 7$. Note that for $\varepsilon = 1_{-4}, 1_{-5}$ the Péclet condition (4.72) is violated on all meshes including the finest level $J = 8$. For those two cases, we employ the multigrid cycle as preconditioner for a (flexible) GMRES(20) iteration, whereas for $\varepsilon = 1_{-1}, 1_{-2}, 1_{-3}$ we use the multigrid cycle as solver. The control smoothing step, i.e. the approximate solution of (RHA), is done here with the FGMRES method instead of the conjugate gradient iteration to prevent possible instabilities due to nonsymmetry of $\hat{L}_j$. In Table 4.7 we give




(a) $\varepsilon = 1_{-1}$.

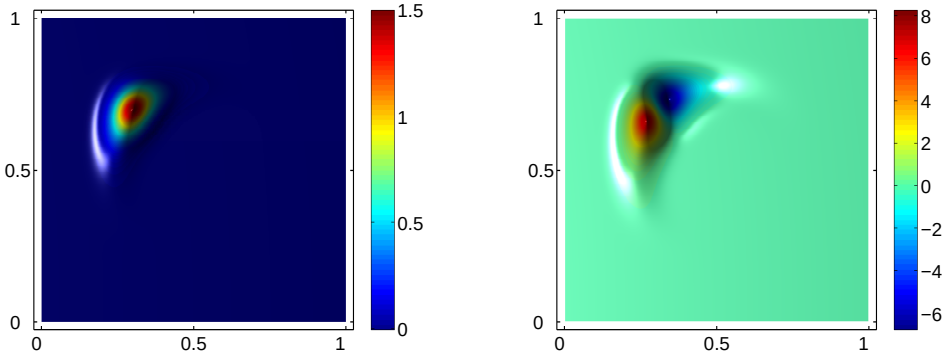



(b) $\varepsilon = 1_{-2}$.




(c) $\varepsilon = 1_{-3}$.

Figure 4.19: Computed optimal states $y_h$ (on the left) and associated optimal controls $u_h$ (to the right) for the tracking problem with the target state $\bar{y}$ given by (4.73). Visualizations of $y_h$ and $u_h$ are shown for the finest grid with a mesh size $h_J = 2^{-8}$ and varying convection parameter $\varepsilon$.




(d) $\varepsilon = 1_{-4}$.

Figure 4.19: Computed optimal states $y_h$ (on the left) and associated optimal controls $u_h$ (to the right) for the tracking problem with the target state $\bar{y}$ given by (4.73). Visualizations of $y_h$ and $u_h$ are shown for the finest grid with a mesh size $h_J = 2^{-8}$ and varying convection parameter $\varepsilon$.

the average convergence rates. In all cases the reduction factors are more or less independent of the fine grid level $J$, however note that for $\varepsilon = 1_{-4}$ we use $\beta = 4$ and for $\varepsilon = 1_{-5}$ we use $\beta = 5$. Furthermore, in these cases the multigrid cycle is accelerated by GMRES(20) on the finest mesh. Still, a noticeable decrease in convergence speed is observed. This is in agreement with the literature on multigrid for convection-diffusion problems, where one speaks of strongly convection dominated for problems with $\varepsilon < 1_{-4}$, which goes hand in hand with a visible change in the flow regime. Still, the rates are acceptable, albeit they come at larger computational cost. Clearly the (approximated) reduced Hessian depends now also on $\varepsilon$ and the shifting character from elliptic to hyperbolic for the state operator for $\varepsilon \to 0$ has an even stronger impact on $H_Z$ and consequently $\hat{H}_Z$. Thus, the smoothing step (RHA) requires considerably more effort, necessitating the increase of $\beta$. Note however that in all cases we still use just one constraint smoothing step, i.e. $\alpha = 1$. We remark that for $\varepsilon = 1_{-4}$ acceleration with a Krylov method is not required for convergence, however without Krylov acceleration we observe the aforementioned limiting rate of 0.5. For $\varepsilon = 1_{-5}$ multigrid as solver was not convergent for all levels and Krylov acceleration is necessary.

Looking closely at the average rates one observes that in some cases the convergence for finest levels $J = 4, 5$ is actually slower than for $J = 6, 7$. A possible explanation might be that the deficiency of the coarse grid correction, which is more pronounced for smaller $\varepsilon$, is to some extent corrected by the smoothing iteration. In particular, the Krylov iterations on (RHA) operate on the whole spectrum of $\hat{H}_Z$ and thus also reduce those smooth modes which are not corrected by the coarse grid correction. In this case, adding more levels has a similar effect as increasing $\beta$ for a fixed level, since

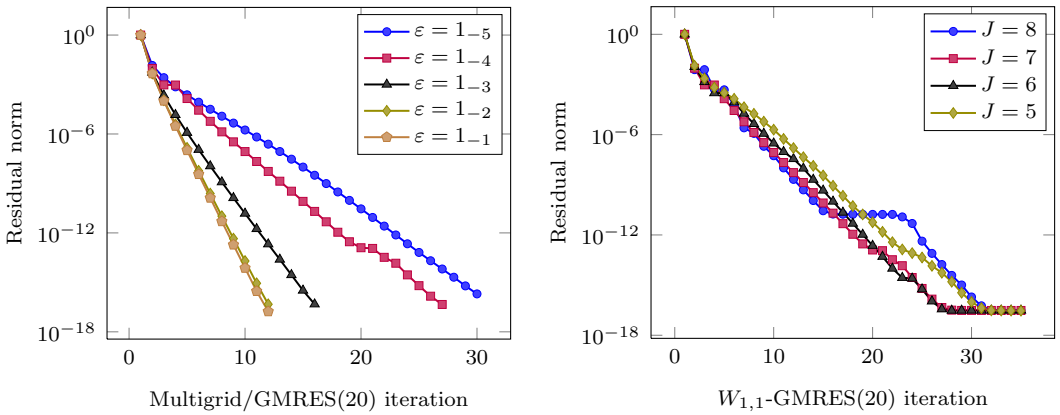



Figure 4.20: Convergence histories for $\varepsilon = 1_{-1}, \dots 1_{-5}$ and $J = 6$, i.e. $h_J = 2^{-10}$ (left), and $\varepsilon = 1_{-4}$ and $J = 5, 6, 7, 8$ (right). For $\varepsilon = 1_{-4}, 1_{-5}$, a $W_{1,1}$-cycle is employed as preconditioner for a GMRES(20) iteration, whereas for $\varepsilon = 1_{-3}, 1_{-4}, 1_{-5}$ the multigrid cycle is used as solver.

the affected smooth modes are treated on every level $j = 1, \dots, J$. Of course this increase in speed is limited by the underlying multigrid convergence for the constraint PDE and therefore the reduction factor does not further improve from $J = 6$ to $J = 7$ (in fact, from multigrid experience for convection-diffusion problems we would expect an increase again for finer levels, which should be mild however due to employing $W$-cycles and Krylov acceleration). Another fact influencing the measurements of the average reduction factor is the presence of stagnation plateaus, which are often observed due to the outer Krylov iteration. Such a plateau is visible e.g. for $J = 8$ in Figure 4.20 right. There, we show the convergence history of the $W_{1,1}$-preconditioned GMRES(20) iteration for $\varepsilon = 1_{-4}$ and $J = 5, \dots, 8$. We observe that for a fixed $\varepsilon$ the convergence speed is nearly independent of $J$, with a mild sensitivity due to the aforementioned plateaus of the outer iteration and the correction behavior of the smoother with respect to smooth error components. On the left side in Figure 4.20 we show the residual reduction for fixed $h_J = 2^{-10}$ and $\varepsilon = 1_{-1}, \dots, 1_{-5}$. Clearly the rate of convergence is not robust with respect to $\varepsilon$, nevertheless almost linear convergence is observed and even for the difficult cases $\varepsilon = 1_{-4}, 1_{-5}$ acceptable convergence rates are obtained.

All in all, we point out that the current implementation of our multigrid solver is not optimally adapted to treat strongly convection-dominated problems. Still, keeping in mind the shortcomings with respect to proper treatment of the constraint PDE, the obtained results are very promising. We have shown that even for fairly difficult constraint PDEs, where the development of a robust multigrid solver for the constraints alone is far from trivial, we obtain acceptable rates of convergence for the associated optimal control problem.

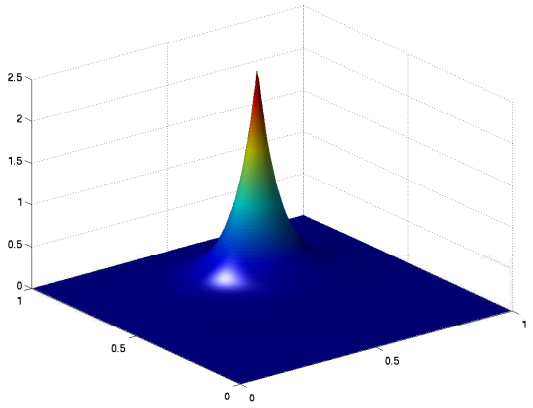



Figure 4.21: Approximation $\bar{y}_h$ of the target state $\bar{y}$ given by (4.75) on a mesh with $h = 2^{-8}$.

### 4.5.5 Robustness and the Role of Regularization

We will now discuss the effect of the regularization parameter $\sigma$ on the solution of the optimal control problem and the convergence behavior of our multigrid method. For the $L^2$-tracking functional (2.1), the value of $\sigma$ puts a weight on the cost of the control required to expend in order to minimize the (discrete) tracking error

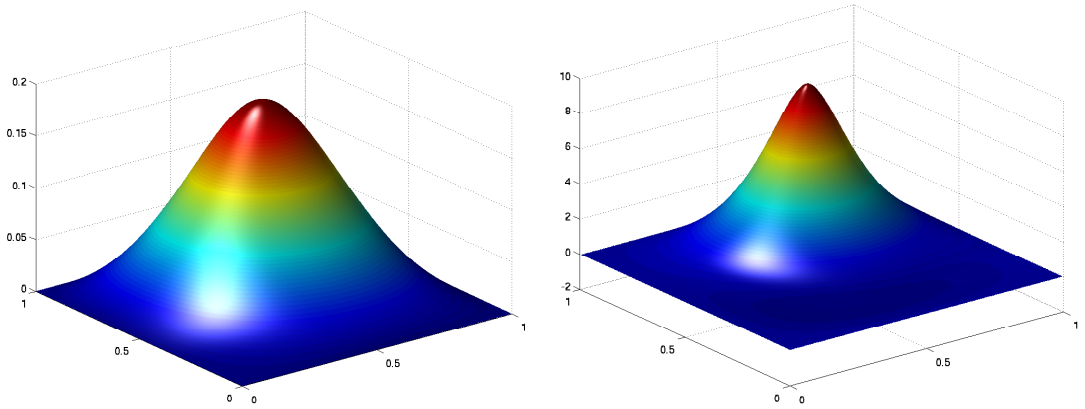
$$e_{h,\mathrm{tr}} = \|y_h - \bar{y}_h\|^2_{L^2,h}. \tag{4.74}$$

Figure 4.21 depicts the discrete approximation $\bar{y}_h$ of a target state $\bar{y}$ which is given by a localized peak

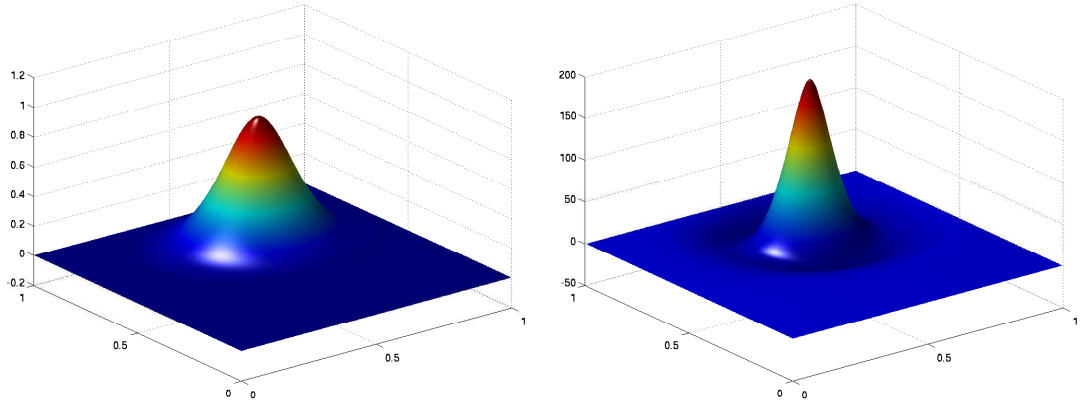
$$\bar{y} = 2.5 \exp\Big(-\sqrt{350(x-0.7)^2 + 200(y-0.6^2)}\Big), \tag{4.75}$$

on a uniform mesh with $h = 2^{-8}$. Figure 4.22 gives a visual impression of the impact which $\sigma$ has on the solution of ($\mathrm{LQP}_h$). The figure shows the computed optimal states $y_h$ on the left and the associated optimal controls $u_h$ on the right for the finest grid level with mesh size $h_J = 2^{-8}$. From top to bottom, the regularization parameter $\sigma$ takes on the values $1_{-3}, 1_{-5}, 1_{-7}$ and $1_{-9}$. Clearly, the smaller $\sigma$, the better is the tracking of the desired target state. However, at the same time, the optimal control degenerates to an unbounded point source. Recall from Chapter 2 that the tracking problem in the $L^2$-setting does not possess a bounded solution for $\sigma = 0$. For unbounded $\mathcal{U}_{\mathrm{ad}}$, the existence of a solution to the optimal control problem in the case of vanishing $\sigma$ could only be established in a stronger norm, which in turn required a higher regularity of the target state $\bar{y}$, cf. Remark 2.7. In the control-constrained case, which is discussed in Chapter 5, for $\sigma \to 0$ one obtains a so-called bang-bang control, that is, a control function that takes on its admissible bounds almost everywhere.

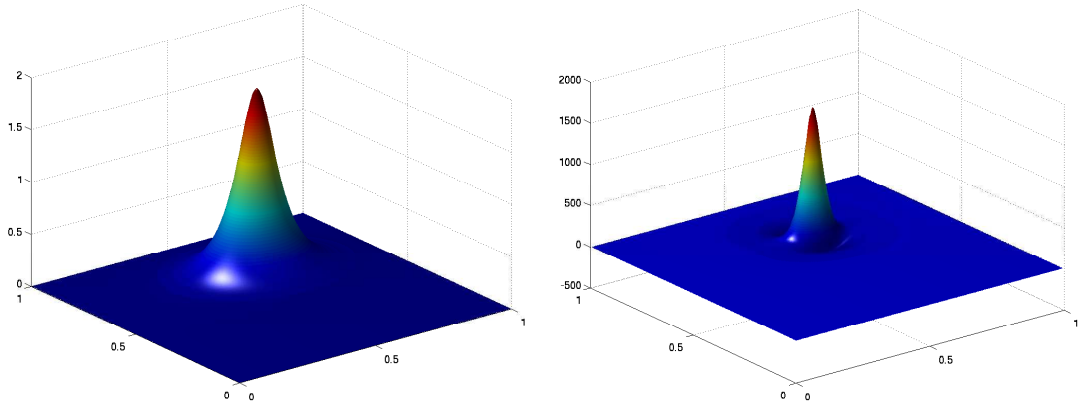
In Figure 4.23, for the same problem we show the tracking error (4.74) as well as



(a) $\sigma = 1_{-3}$.

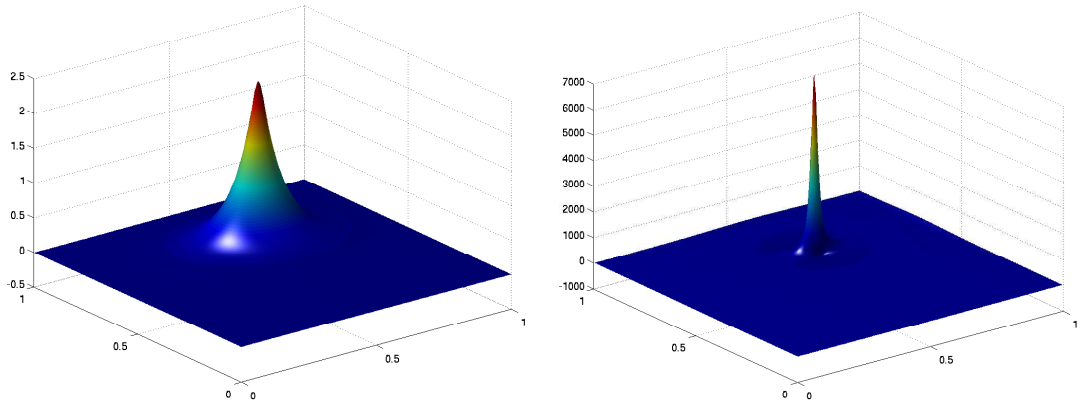


(b) $\sigma = 1_{-5}$.



(c) $\sigma = 1_{-7}$.

Figure 4.22: Computed optimal states $y_h$ (on the left) and associated optimal controls $u_h$ (to the right) for the tracking problem with the target state $\bar{y}$ given by (4.75). Visualizations of $y_h$ and $u_h$ are shown for the finest grid with a mesh size $h_J = 2^{-8}$ and varying regularization parameter $\sigma$.



(d) $\sigma = 1_{-9}$.

Figure 4.22: (cont'd) Computed optimal states $y_h$ (on the left) and associated optimal controls $u_h$ (to the right) for the tracking problem with the target state $\bar{y}$ given by (4.75). Visualizations of $y_h$ and $u_h$ are shown for the finest grid with a mesh size $h_J = 2^{-8}$ and varying regularization parameter $\sigma$.

the value of the discrete approximation to the objective functional (2.1),

$$\mathcal{J}_h(y_h, u_h) = \frac{1}{2} e_{h,tr} + \frac{\sigma}{2} \|u\|^2_{L^2,h}. \tag{4.76}$$

Furthermore, the true cost $\|u\|^2_{L^2,h}$ as well as the $\sigma$-weighted cost $\frac{\sigma}{2}\|u\|^2_{L^2,h}$ of the control are shown. The value of $\mathcal{J}_h$ is dominated by the tracking error and decreases at the same rate as $e_{h,\mathrm{tr}}$ for $\sigma \to 0$. At the same time, the true cost increases monotonically, while the weighted cost has a peak for $\sigma \sim 1_{-4}$.

These results show that it might be necessary to compute solutions for quite small $\sigma$ if a small tracking error $e_{\mathrm{tr}}$ is the aim. On the other hand, in many practical settings, the target state $\bar{y}$ is the result of measurements which are perturbed by noisy data. Therefore, in order to obtain meaningful results, the regularization parameter has to be linked to the uncertainty level in the given data. Several strategies such as the L-Curve method, the discrepancy principle or generalized cross-validation exist to tackle this task. We refer to [153] for further discussion of these and other methods, since the topic is too complex to justify even a brief survey in this thesis. Nevertheless, the mathematical question of robustness in the limit case of $\sigma \to 0$ is interesting in itself and we will discuss it later in this and the next section.

The method developed so far converges with rates independently of $\sigma$ for large to moderate values of $\sigma$. Here, the precise meaning of "large to moderate" depends on the condition (4.58) in case of the smoothing iteration $\mathcal{S}_\alpha$. For $\mathcal{S}_{\alpha,\beta}$, the range of convergence with respect to $\sigma$ is somewhat larger.

In general we would expect the multigrid iteration to converge if all smoothing factors on all levels are strictly less than unity, i.e. relation (4.58) holds for all mesh sizes. Thus, since $\sigma > h_1^4/4$ obviously implies (4.58) for all levels $j > 1$, we expect

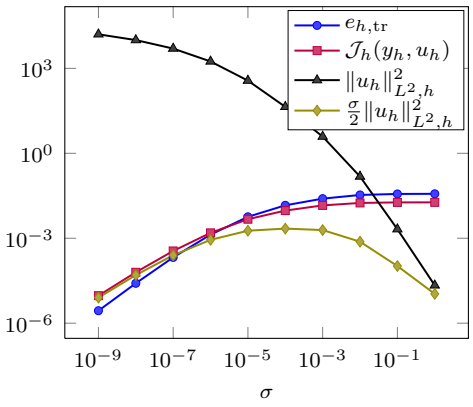



Figure 4.23: Tracking error $e_{h,\mathrm{tr}}$, value of discrete objective functional $\mathcal{J}_h(y_h,u_h)$, weighted cost $\frac{\sigma}{2}\|u_h\|^2_{L^2,h}$ and true cost $\|u_h\|^2_{L^2,h}$ for different values of $\sigma \to 0$ and $h = h_J = 2^{-8}$.

it to be a sufficient condition for convergence. Note that (4.58) does not need to be satisfied on the coarsest level $j = 0$ where we solve directly. However, rather than a sharp switching from convergence to divergence, we find in numerical experiments that a gradual deterioration of the convergence rate occurs, mainly due to the following two reasons: first, the LFA neglects the influence of the boundary conditions, which can be strong on coarser levels with few interior points, and second, in practice a loss of the smoothing property on coarser levels is compensated by a better coarse grid correction due to more frequent visits to coarse grids and to some extent by (additional) smoothing on finer levels, in particular when employing F- or W-cycles. In any case, based on the findings in Section 3.3 we expect $\mathcal{S}_{\alpha,\beta}$ to exhibit stronger convergence properties for smaller values of $\sigma$ than $\mathcal{S}_\alpha$ due to the better approximation property with respect to $H_Z$. These preliminary considerations are confirmed by the following numerical experiments.

We consider the model problem ($\mathrm{LQP}_h$) with zero right hand side and random initial guess. The mesh width on the coarsest level is $h_0 = 1/2$. Then, according to (4.58), the critical value of $\sigma$ such that the smoothing factor $\rho_j$ stays bounded from above by unity on all levels $1 \le j \le J$ is given by

$$\sigma_{\mathrm{crit}} \sim \frac{1}{4}\Big(\frac{1}{4}\Big)^4 \sim 9.765_{-4}. \tag{4.77}$$

In Figure 4.24 on the left we show the convergence history of a multigrid $V_{1,1}$-cycle with smoothing iteration $\mathcal{S}_\alpha$ for different values of $\sigma$ and a mesh size of $h_J = 2^{-10}$. To the right, the eigenvalues of the corresponding multigrid iteration matrix $M_G$ for $h_J = 2^{-5}$ are shown. We see that for $\sigma \to 0$ an isolated negative eigenvalue with largest absolute value dominates the iteration, in this respect the multigrid iteration behaves similar to the Richardson iteration $\mathcal{S}_\alpha$, cf. Figure 4.4. Furthermore, recall from Section 4.3

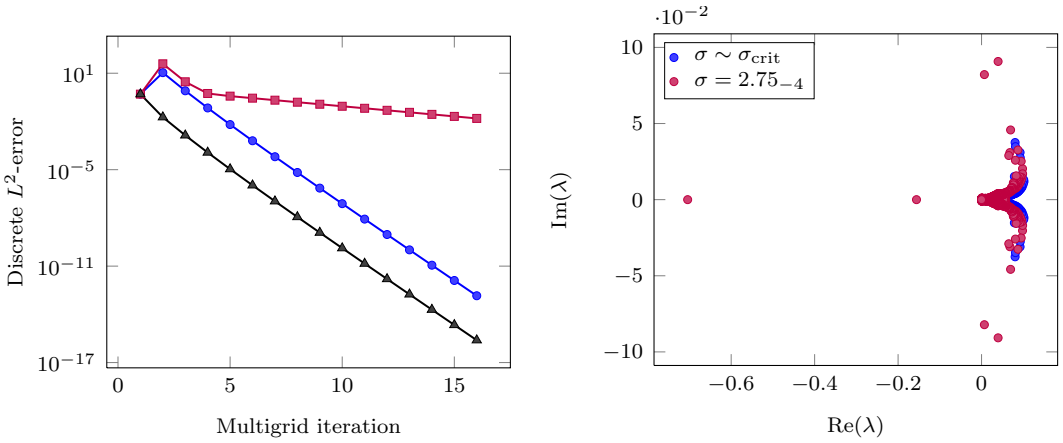



Figure 4.24: Convergence of the $V_{1,1}$-cycle with smoothing iteration $\mathcal{S}_{j,1}$ with $h = 2^{-10}$ (left) and eigenvalues of the corresponding multigrid iteration matrix $M_G$ with $h = 2^{-5}$ (right). The different values of $\sigma$ are $\sigma = \sigma_{\text{crit}}$, $\sigma = 2.75_{-4}$ and as reference with optimal rate $\sigma = 1.0$.

that the absolute value of this isolated eigenvalue quickly grows unbounded and correspondingly for the smoothing factor (4.57) one obtains $\rho_{\text{LFA}} \gg 1$. This explains the rapid onset of divergence for $\sigma < 2.75_{-4}$ (for $\sigma = 2.5_{-4}$ (not shown in the figure), the iteration is divergent). The large smoothing factor indicates that $\mathcal{S}_\alpha$ is strongly diverging and explains why the onset of divergence is not prevented when using an $F$- or $W$-cycle. A slight improvement can be achieved with acceleration by an outer Krylov iteration, see Figure 4.25. For the same reasons as discussed in Section 3.3 a flexible variant of GMRES was chosen as outer iteration. Furthermore, we set the restart value to 15. Here, the phenomen discussed in Section 3.3 arises again, namely that a relatively small restart value yields faster convergence.[8] However, finding the optimal restart value is an open question. For moderate values of $\sigma$ the acceleration yields a significant improvement, however already for $\sigma = 5_{-5}$, no satisfactory convergence is achieved. Also shown in Figure 4.25 is the unaccelerated $V_{1,1}$-cycle, but this time with an increased resolution of the coarsest mesh, $h_0 = 1/4$. Note that perfect linear and robust convergence is regained due to the increase of $h_0$.

For the numerical examples in Sections 4.5.1 to 4.5.3 we have used the smoothing iteration $\mathcal{S}_{\alpha,\beta}$, which we expect to converge for smaller values of $\sigma$ than $\mathcal{S}_\alpha$. This is confirmed by the results presented in Figure 4.26. We clearly see that the $V_{1,1}$-cycle with the smoothing iteration $\mathcal{S}_{j,1,1}$ produces almost acceptable convergence rates for $\sigma = 2.75_{-4}$, whereas $\mathcal{S}_{j,1}$ was close to divergence in this case, cf. Figure 4.24. Furthermore, regarding the eigenvalue distribution of the iteration matrix, we gather

[8] In this context we remark that Krylov acceleration for multigrid is closely related to the method of iterant recombinations, see [147]. There it is noted that a smaller number of iterates in the recombination process often produces better results.

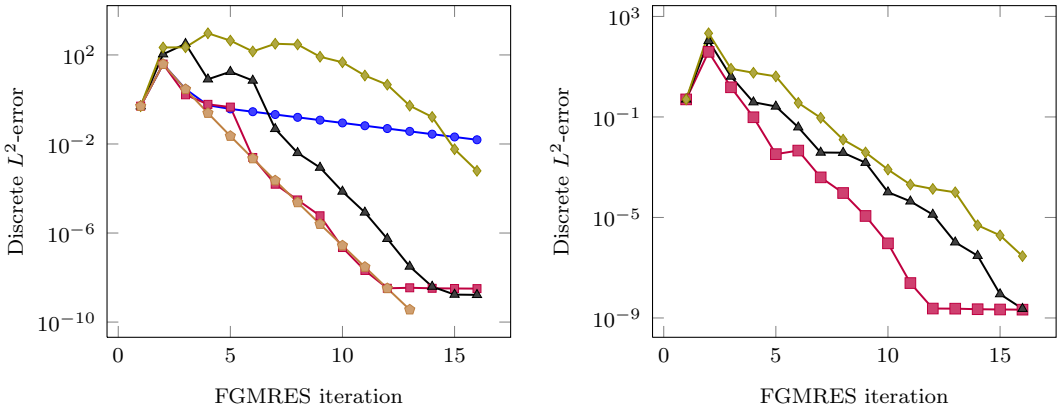



Figure 4.25: FGMRES(15) preconditioned with a $V_{1,1}$-cycle. On the left, the smoothing iteration is $\mathcal{S}_{j,1}$ for $\sigma = 2.75_{-4}$, $\sigma = 1_{-4}$ and $\sigma = 5_{-5}$. Also shown for comparison is the convergence of unaccelerated multigrid for $\sigma = 2.75_{-4}$ and $\sigma = 2.75_{-4}$ with a coarsest mesh size $h_0 = 1/4$. To the right, the smoothing iteration is $\mathcal{S}_{j,1,1}$ and same colors indicate same values of $\sigma$.

from Figure 4.26 (right) that the negative outliers due to small values of $\sigma$ are not as dominant as for $\mathcal{S}_{j,1}$. Instead, the eigenvalues are grouped more closely around the origin although the imaginary part of some eigenvalues has increased. This eigenvalue distribution proves to be beneficial for the convergence of preconditioned GMRES, see Figure 4.25 on the right. For instance, for $\sigma = 5_{-5}$, the smoothing iteration $\mathcal{S}_{j,1,1}$ results in a very good preconditioner for FGMRES, whereas $\mathcal{S}_{j,1}$ did not yield acceptable error reduction.

Also in contrast to the smoother $\mathcal{S}_{j,1}$, for $\mathcal{S}_{j,1,1}$ a cycle with better smoothing properties proves to be beneficial and the onset of divergence is delayed. In Figure 4.27 we show the convergence histories for the $W_{1,1}$- and $W_{2,2}$-cycle. Robust convergence is achieved for values down to $\sigma \sim 2_{-5}$. For smaller values of $\sigma$, soon also the $W$-cycle becomes divergent. Acceleration by a Krylov method does not improve the situation. This is due to the fact that the reduction rates for the convergent $W$-cycle are already very good and can not be reduced further by much. When the $W$-cycle diverges, acceleration by an outer Krylov method does not restore the convergence. On the right side of Figure 4.27 we plot the eigenvalues of the multigrid iteration matrix for a $W_{1,1}$-cycle and a value of $\sigma = 5_{-5}$. Note the different scaling of the axis when comparing with the $V_{1,1}$-cycle, cf. Figure 4.26, or the smoothing iteration $\mathcal{S}_{j,1,}$, cf. Figure 4.24. The eigenvalues for the $W$-cycle are grouped more evenly around the origin and the extension of the spectrum along the real axis is smaller than in both other cases. In particular, the negative outliers are even less dominant than before.

Throughout this section we considered the coarsest level $h_0 = h_c = 1/2$ and a finest level with mesh width $h_J = 2^{-10}$. We point out that, although the value of the critical $\sigma$ for which robust convergence can be achieved depends on the relation of $\sigma$ to the

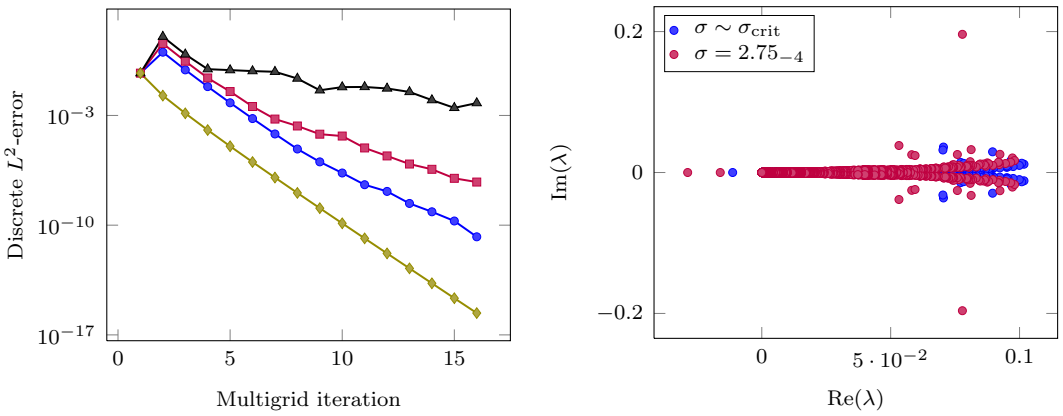



Figure 4.26: Convergence of the $V_{1,1}$-cycle with smoothing iteration $\mathcal{S}_{j,1,1}$ with $h = 2^{-10}$ (left) and eigenvalues of the corresponding multigrid iteration matrix $M_G$ with $h = 2^{-5}$ (right). The different values of $\sigma$ are $\sigma = \sigma_{\text{crit}}$, $\sigma = 2.75_{-4}$ $\sigma = 1_{-4}$ and as reference with optimal rate $\sigma = 1.0$.

next to coarsest mesh size $h_1$, the robustness results do neither depend on the number of levels, $J$, nor on the fine grid mesh size $h_J$. To demonstrate this, we consider a sequence of increasing coarsest mesh sizes $h_c$ and decreasing values of $\sigma$ such that (4.58) remains satisfied. We define $j_0 = -\log_2 h_c$ and consider coarse mesh sizes given by $j_0 = 2, \ldots, 6$. For $j_0 = 2$ we set $\sigma_{j_0} = 5_{-4}$ and for each increase of the coarsest mesh size, we reduce $\sigma_{j_0}$ by a factor of $1/16$, i.e. $\sigma_{j_0}$ is given by $\sigma_{j_0} = 5_{-4} \cdot 2^{-4(j_0-2)}$. The smallest value obtained is $\sigma_6 \sim 7.63_{-9}$. We distinguish two test cases. For the first test case, we fix the fine grid mesh size $h_J = 2^{-11}$. Thus, with increasing $j_0$, we obtain a decreasing number of levels in the multigrid hierarchy, i.e. $J = 11 - j_0$. For the second test case, we fix $J = 5$ and thus obtain a sequence of decreasing fine grid mesh sizes $h_J = 2^{-(j_0+5)}$. The corresponding convergence histories are shown in Figure 4.28 left and right, respectively. In all cases we used a $V_{1,1}$-cycle and the smoothing iteration $\mathcal{S}^{\nu}_{j,1,1}$. All in all we conclude that asymptotically for $h_J \to 0$ the reduction rates are independent of $\sigma$, $J$ and $h_J$ as long as (4.58) is satisfied.

Last we remark that the qualitative behavior does not change when solving the problems from Sections 4.5.2 or 4.5.3. Provided that appropriate values for $\alpha, \beta$ are chosen and ILU(0) or ALGS are employed as constraint smoother, then also for problems involving anisotropic constraints the robustness with respect to $\sigma$ is similar to the model problem case and analogous results are obtained.

## 4.6 Robustness Enhancement by Spectral Filtering

In this section we discuss the underlying reasons for the lack of robustness in the case of $\sigma \ll 1$. This issue is rarely discussed in the optimization and optimal control literature,

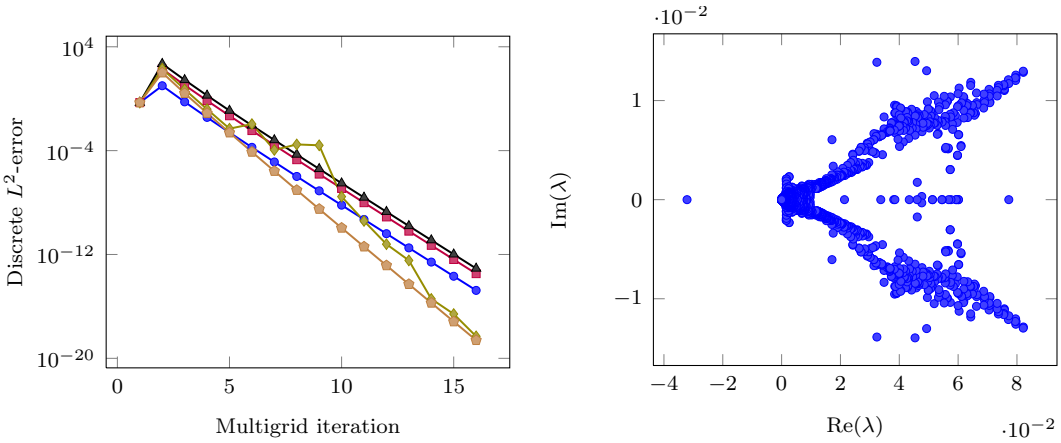



Figure 4.27: Left: Convergence of $W$-cycles with smoothing iteration $\mathcal{S}_{j,1,1}$. $W_{1,1}$-cycle with $\sigma = \sigma_{\text{crit}}$, $\sigma = 5_{-5}$ and $\sigma = 2_{-5}$. $W_{2,2}$-cycle with $\sigma = 2_{-5}$ and $\sigma = 1_{-5}$. On the right we show the eigenvalues of the multigrid iteration matrix $M_G$ for a $W_{1,1}$-cycle and $\sigma = 5_{-5}$.

in most cases the penalty parameter is a priori set to some value which is assumed to be appropriate for the problem at hand. Common values of $\sigma$ range between $1_{-2}$ and $1_{-6}$ [53], a range for which iteration numbers of most algorithms are still acceptable. To the contrary, the question of finding the optimal regularization parameter is an important subtopic in the inverse problems community. From the "inverse problem point of view", it is natural to expect a connection between $\sigma$ and $h = h_J$, since $\sigma$ is connected to the noise level in the data and less noise or higher fidelity of the data allows for finer features of the solution to be resolved, which in turn requires a finer mesh. However, as discussed in the previous section, our current implementation of the multigrid solution imposes a restriction on the next to coarsest mesh size $h_1$, which is introduced by insufficient smoothing with respect to $u_h$. Limitations on $h_1$ are quite common for multigrid algorithms when applied to difficult problems such as Fredholm equations of the second kind [84, 85] or convection-diffusion problems [103]. It is the aim of the discussion in this section to eliminate the condition (4.58) on $h_1$. As it will turn out, robust convergence results under a condition on the fine grid mesh size $h = h_J$ only. In fact, we believe that this is not a restriction at all, as one could argue that computing the solution to a regularized ill-posed problem for $\sigma \ll 1$ on a coarse mesh annihilates the effect of small $\sigma$, since the discretization itself can be interpreted as a regularization, see [126] and related publications on "self-regularization". In this respect, a small parameter $\sigma$ is only meaningful if it is accompanied by a suitable fine grid size $h = h_J$. We focus on the model problem, however all considerations also apply when the constraints are given by the general elliptic second-order BVP (2.20).

As can be gathered from the numerical experiments conducted in Section 3.3 and the smoothing analysis in Section 4.3, the critical component of the multigrid itera-

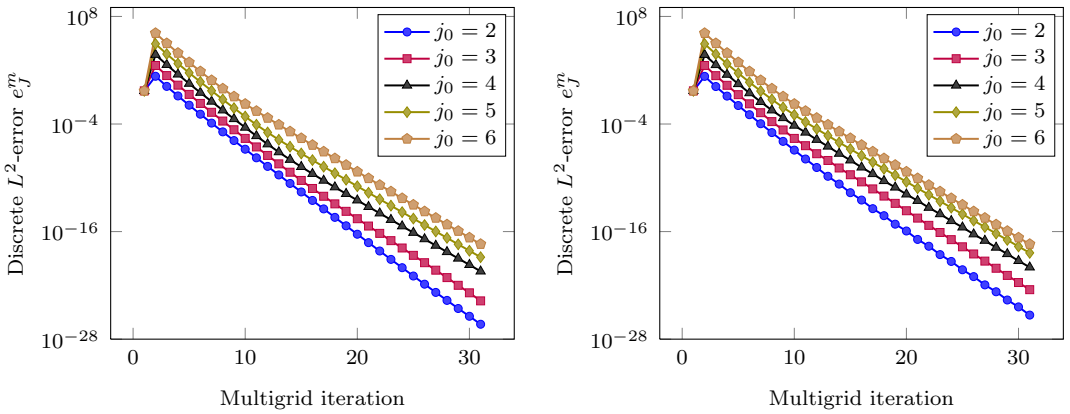



Figure 4.28: Convergence histories for $j_0 = 2, \ldots, 6$. Left: constant mesh size of finest level is $h_J = 2^{-11}$, the number of levels $J = 11 - j_0$ decreases with increasing $j_0$. Right: fixed number of levels $J = 5$, the fine grid mesh size $h_J = 2^{-(j_0+5)}$ decreases with increasing $j_0$. In both cases the regularization parameter is given by $\sigma_{j_0} = 5_{-4} \cdot 2^{-4(j_0-2)}$.

tion which can lead to divergence for small $\sigma$ is the smoothing step with respect to the control component $u_h$, i.e. the smoothing step (RHA) (cf. step 3 in Algorithm 5, page 67). A careful look at the continuous formulation of (RHA) foreshadows difficulties for solvers of the corresponding discrete system. Recall from Chapter 2 that the continuous reduced Hessian $\mathcal{H}$ is obtained by computing the gradient of the reduced objective functional $\hat{\mathcal{J}}$ (2.5) and is given by

$$\mathcal{H}u = \sigma u + S^* S u, \tag{4.78}$$

where $S : L^2(\Omega) \to H_0^1(\Omega)$ is the control-to-state mapping or solution operator of the constraint PDE and $S^*$ denotes its adjoint. Since the embedding $H_0^1(\Omega) \hookrightarrow L^2(\Omega)$ is compact, it follows that $S : L^2(\Omega) \to L^2(\Omega)$ and consequently also its adjoint $S^*$ as well as $S^*S$ are compact. For sufficiently large $\sigma$, the operator $\mathcal{H}$ can be interpreted as a compact perturbation of the identity. The continuous analogon to (RHA) states a Fredholm equation of the second kind and in the unregularized case, i.e. for $\sigma = 0$, one obtains a Fredholm equation of the first kind. Such problems often arise in inverse problems, where the operator $S$ corresponds to an integral operator with a smooth kernel on a compact domain. For instance in image deblurring, under the assumption of translation invariance, the operator $S$ corresponds to the convolution with a point spread function. The connection to ill-posed integral equations also is evident when expressing $S$ as an integral operator via Green's function.

Solving Fredholm equations of the first kind is well-known to be ill-posed in the sense of Hadamard, since the solution does not depend continuously on the right hand side data. In order to overcome this difficulty, regularization is introduced and in this respect the objective functional (2.1) corresponds to an output least-squares

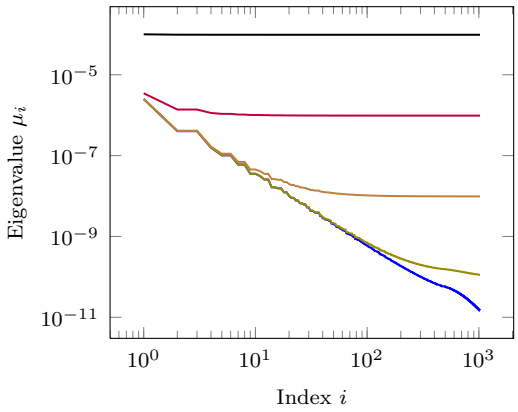



Figure 4.29: Eigenvalues $\mu_i, i = 1, \dots, 1024$ of $H_\sigma$ for the linear-quadratic model problem with $h = 1/32$ and — $\sigma = 0.0$, i.e. unregularized, — $\sigma = 1.0_{-1}$, — $\sigma = 1.0_{-3}$, — $\sigma = 1.0_{-5}$ and — $\sigma = 1.0_{-7}$.

or data-discrepancy functional with an additional $L^2$-Tikhonov regularization term. From the spectral theory for compact operators we deduce that the countably many isolated eigenvalues of the compact part of $\mathcal{H}$ converge to zero. We expect the discrete representation to reflect this property. The discrete approximation to $\mathcal{H}$ is given by (cf. (3.12))

$$H_\sigma = \sigma M_h + M_h^T L_h^{-T} M_h L_h^{-1} M_h, \tag{4.79}$$

i.e. the mass matrix $M_h$ has the role of the identity on $L^2(\Omega)$ and the discrete control-to-state mapping is given by $S_h = L_h^{-1} M_h$. Figure 4.29 shows the eigenvalues $\mu_i$ of $H_\sigma$ for different values of $\sigma$, including the unregularized case $H_0$, for a mesh with $h = 1/32$. The rapid decay of $\mu_i$ for small $\sigma$ is obvious. Correspondingly, the condition number $\kappa(H_\sigma)$ grows for $\sigma \to 0$, in fact

$$\kappa(H_\sigma) \leq \frac{\mu_{\max}(\sigma M_h) + \|S_h^T M_h S_h\|}{\mu_{\min}(\sigma M_h)} = \mathcal{O}\big(1/\sigma\big). \tag{4.80}$$

From (4.80) follows the worst-case estimate $\mathcal{O}(1/\sqrt{\alpha})$ for the number of iterations the cg method requires to solve (RHA). We would like to point out here that, although (4.80) holds independent of $h$ and consequently the number of cg iterations does not depend on $h$, the actual number of required iterations can be too large for practical problems, in particular if a single iteration is expensive (cf. the discussion in Section 3.3 and the results presented in Table 3.1).

As part of a smoothing step for (RHA) however, we are not interested in the asymptotic convergence behavior of the cg method, but instead it is of primary interest how the cg iteration acts with respect to the high frequency components of the error. The eigenvalues of $L_j$ discretized on a uniform grid are given by [86]

$$\lambda_{k,l} = 4h^{-2}(\sin^2(k\pi h/2) + \sin^2(l\pi h/2)), \tag{4.81}$$

and the corresponding discrete eigenfunctions are

$$\phi_{k,l} = \sin(k\pi x)\sin(l\pi y), \tag{4.82}$$

with $1 \le k, l \le N$ and $x, y$ range over the coordinates of the $N^2$ cell centers. Thus, high frequencies belong to large eigenvalues, which is the reason why classical iterative methods possess the smoothing property with respect to the model problem. This situation is completely different for $H_\sigma$, since its eigenvalues are given by

$$\mu_{k,l} = \sigma h^2 + \frac{h^6}{\lambda_{k,l}^2}. \tag{4.83}$$

However, the same eigenvector $\phi_{k,l}$ belongs to $\mu_{k,l}$ and $\lambda_{k,l}$. Thus, for decreasing eigenvalues $\mu_{k,l}$ the associated eigenfunctions become increasingly oscillatory which accounts for the failure of standard multigrid methods when employed to solve systems with coefficient matrix $H_0$ or $H_\sigma$ for $\sigma \ll 1$. Consequently, also the (RHA)-step within $\mathcal{S}^\nu_{j,\alpha,\beta}$ does not provide sufficient smoothing with respect to $u_h$.

In order to address the smoothing behavior of the cg method applied to (RHA), we recall the estimate (3.15). Denoting the minimal polynomial in the $m$-th iteration by $\pi_m$, the value $\pi_m(\mu_{k,l})$ gives the reduction (or growth) of the error component in the direction of the eigenfrequency $\phi_{k,l}$ in the $m$-th cg iteration. The normalizing condition $\pi_m(0) = 1$ already indicates that the value of $\pi_m$ will be close to unity for small eigenvalues $\mu_{k,l}$ and small iteration numbers $m$, and thus the reduction of the high frequencies[9] $\phi_{k,l}$, $N/2 < k, l \le N$ of the error will be minuscule. The precise reduction can be inferred from the Chebyshev polynomials of the first kind. The polynomial of degree $m$ is given by

$$T_m(\omega) = \frac{1}{2}\Big[\big(\omega + \sqrt{\omega^2 - 1}\big)^m + \big(\omega - \sqrt{\omega^2 - 1}\big)^m\Big]. \tag{4.84}$$

The polynomial which minimizes (3.15) is then

$$\pi_m(\lambda) = \frac{T_m\Big(\frac{\lambda_{\max}+\lambda_{\min}-2\lambda}{\lambda_{\max}-\lambda_{\min}}\Big)}{T_m\Big(\frac{\lambda_{\max}+\lambda_{\min}}{\lambda_{\max}-\lambda_{\min}}\Big)}, \tag{4.85}$$

where $\lambda_{\max}, \lambda_{\min}$ are the extremal eigenvalues of the matrix in question. Figure 4.30 shows the minimal polynomials of degree 1, 2 and 5, i.e. the polynomials which correspond to the cg iterations 1,2 and 5, over the interval $[0, \lambda_{\max}]$. Furthermore, the symbols • • • mark the value of $\pi_m$ on the eigenvalues $\mu_{k,l}$ of $H_\sigma$. The figure shows the situation for three different values of $\sigma$, namely $\sigma = 1_{-3}$ (top right), $1_{-5}$(bottom

[9]Recall from Section 4.1 that grid functions are high frequencies *with respect to a given fine grid* if they show up as spurious smooth modes on the next coarser grid due to aliasing.

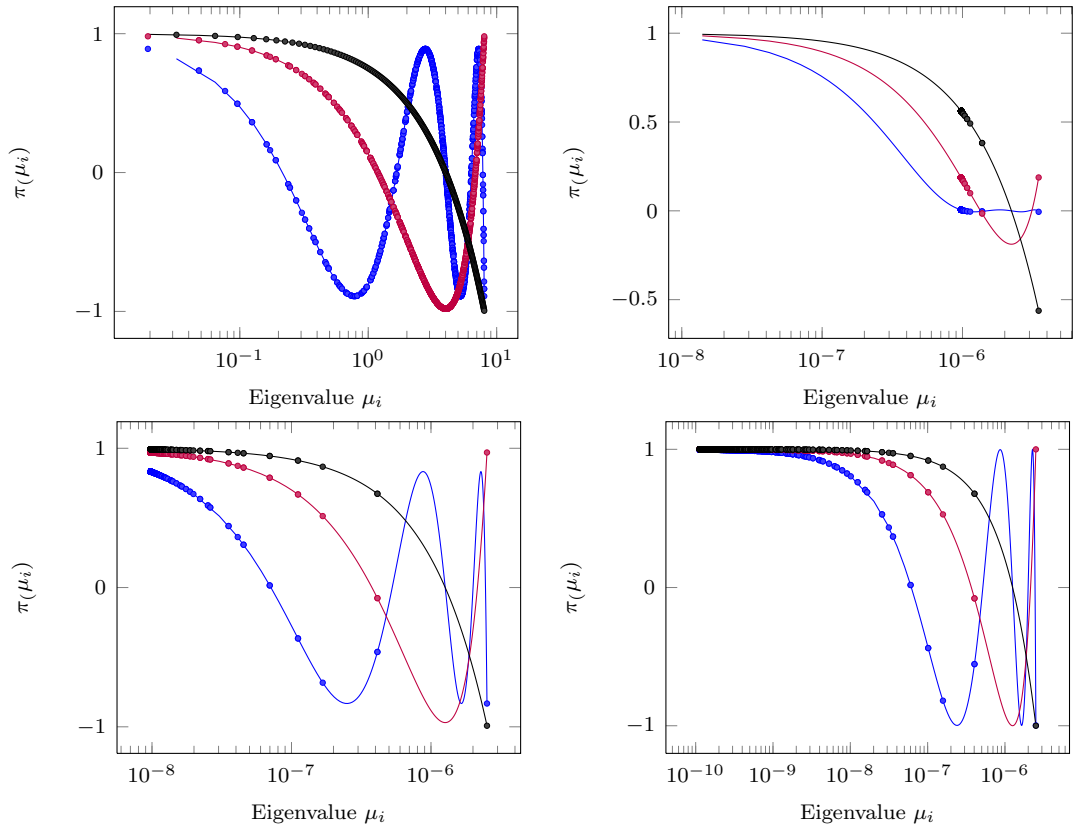



Figure 4.30: Values $\pi(\mu_i), i = 1, \ldots, 1024$ of the minimal polynomial at the eigenvalues $\mu_i, i = 1, \ldots, 1024$ of $H_\sigma$ for the linear-quadratic model problem with $h = 1/32$ and $\sigma = 1_{-3}$(top right), $\sigma = 1_{-5}$ (bottom left) and $\sigma = 1_{-7}$(bottom right). For comparison, $\pi(\mu_i)$ is shown for the eigenvalues of $L_h$ (top left). Shown are the polynomials — $\pi_1$, — $\pi_2$ and — $\pi_5$.

left) and $1_{-7}$(bottom right). All cases use a mesh size of $h = 1/32$. For comparison, the situation for the eigenvalues $\lambda_{k,l}$ of $L_j$ is shown in the top left. For a relatively large value of $\sigma$ such as $1_{-3}$, the operator $H_\sigma$ can be seen as a small perturbation of a scaled identity and thus the cg method converges rapidly. For smaller $\sigma$ however, the strong clustering of the $\mu_{k,l}$ close to zero poses a severe difficulty and even a higher order polynomial (degree 5 in the case of $\sigma = 1_{-5}, 1_{-7}$) does not generate significant reduction with respect to small $\mu_{k,l}$. Thus we expect that increasing $\beta$ in $\mathcal{S}^{\nu}_{j,\alpha,\beta}$, i.e. an increase in the number of cg iterations applied to (RHA), by a small amount only will not significantly ameliorate the smoothing behavior (on the other hand, a $\sigma$-dependent increase of $\beta$ will destroy the $\mathcal{O}(N)$ cost per iteration). Even to the contrary, the error reduction with respect to most of the larger eigenvalues or smooth error components, is stronger than the reduction of the high frequencies. Thus, when applied to $H_\sigma$ for $\sigma \ll 1$, the cg method acts as a rougher instead as a smoother. The consequence for

the multigrid iteration is that the complementarity between the smoothing iteration and the coarse grid correction does not hold anymore, leading at best to a decrease of convergence speed or worse to divergence of the whole multigrid process.

In order to restore the complementarity between smoothing and the coarse grid correction we construct a high-pass filter which removes the low frequency components from the control error and thus forces the cg iteration to act exclusively on the high frequency part of the spectrum. In a very general context considering a linear system $Ax = b$ with positive definite $A$, the family of "partial spectral factorization" algorithms [70] pursues a similar idea. There, a near-invariant subspace associated with the smallest eigenvalues of $A$ is removed from the iteration by orthogonal projections, resulting in so-called "deflated" Krylov iterations. Furthermore, several algorithms aimed specifically at solving Fredholm equations of the first and second kind involve some splitting of the approximation space into subspaces related to smooth and oscillatory components. Probably the first publications in this direction is [104], where a multilevel method for a Tikhonov-regularized first kind Fredholm integral equation is proposed. Later works building on these results include two- and multilevel preconditioners [57, 153] and wavelet-based approaches [133]. Analytical as well as numerical results in [57, 104, 133] indicate that the convergence factors do not depend on $h_J$, but on $J$ and thus implicitly on $h_c$. In this sense, the results are similar to our results as presented in Section 4.5.5. In our setting however, the idea of frequency filtering proves beneficial and yields a further improvement of our approach in the sense that it frees us from the constraint on the coarse grid size, which is in contrast to [57, 104, 133].

For all but the coarsest level we denote by

$$W_j^{\mathrm{LF}} \subset W_j, \quad j = 1, \ldots, J \tag{4.86}$$

a subspace of low frequency or smooth functions and by $W_j^{\mathrm{HF}}$ the orthogonal complement consisting of the oscillatory functions in $W_j$. Let $\Pi_j^{\mathrm{LF}} : W_j \to W_j^{\mathrm{LF}}$ be the orthogonal projection onto the low frequency subspace and $\Pi_j^{\mathrm{HF}} : W_j \to W_j^{\mathrm{HF}}, \Pi_j^{\mathrm{HF}} = I_j - \Pi_j^{\mathrm{LF}}$ the projection onto the high frequency subspace. Consistent with multigrid theory we envisage $W_j^{\mathrm{LF}}$ as the embedding of the coarse space $W_{j-1} \hookrightarrow W_j$ and consequently we will approximate $\Pi_j^{LF}$ by standard interpolation operators.

Using $I_j = \Pi_j^{\mathrm{LF}} + \Pi_j^{\mathrm{HF}}$, we write $H_j^\sigma$ as

$$H_j^\sigma = \Pi_j^{\mathrm{LF}} H_j^\sigma \Pi_j^{\mathrm{LF}} + \Pi_j^{\mathrm{HF}} H_j^\sigma \Pi_j^{\mathrm{HF}} + \Pi_j^{\mathrm{LF}} H_j^\sigma \Pi_j^{\mathrm{HF}} + \Pi_j^{\mathrm{HF}} H_j^\sigma \Pi_j^{\mathrm{LF}}. \tag{4.87}$$

The last two terms in (4.87) vanish identically for exact projections in the case of the model problem. To avoid solution of Gramian systems, we approximate the projection instead by standard interpolation operators, computing

$$\Pi_j^{LF} \approx P_{j-1}^j R_j^{j-1}. \tag{4.88}$$

Note that here, in contrast to the original multigrid algorithm, we need the unscaled restriction, i.e. the weights have to add up to 1 since we apply $\Pi_j^{LF}$ to a (discrete)

function instead of a residual vector. When using (4.88) the last two terms in (4.87) do not vanish but are still negligible, in particular since we do not intend to solve (RHA) but to reduce the high frequencies. To this end, we consider the low and high frequency order of the interpolation operators (condition (4.13) depends on the polynomial orders), which are defined as follows [41, 90]:

**Definition 4.1.** For a restriction operator $R_j^{j-1}$ the low frequency order $m_R^{\mathrm{LF}}$ and the high frequency order $m_R^{\mathrm{HF}}$ are defined as the largest numbers which satisfy

$$\hat{R}_j^{j-1}(\theta) = 1 + \mathcal{O}(h_j^{m_R^{\mathrm{LF}}}), \quad \theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2 \tag{4.89}$$

$$\hat{R}_j^{j-1}(\theta) = \mathcal{O}(h_j^{m_R^{\mathrm{HF}}}), \qquad \theta \in \left[-\pi, \pi\right)^2 \setminus \left[-\frac{\pi}{2}, \frac{\pi}{2}\right)^2 \tag{4.90}$$

for fixed $\theta$ and $h_j \to 0$, where $\hat{R}_j^{j-1}(\theta)$ is the symbol of $R_j^{j-1}$ (cf. Section 4.3.1).

A large $m_R^{\mathrm{LF}}$ means that smooth components are transferred to coarse grids with only small pollution by spurious high frequency components. A large $m_R^{\mathrm{HF}}$ means that high frequency components are filtered out well by $R_j^{j-1}$, i.e. they show up on the coarse grid as spurious smooth components with very small amplitude. The corresponding orders for a prolongation operator $P_{j-1}^j$ are defined as those of $R_j^{j-1} = (P_{j-1}^j)^*$. Therefore, a large $m_P^{\mathrm{HF}}$ indicates that spurious high frequencies generated by $P_{j-1}^j$ on the fine grid have very small amplitude, whereas a large $m_P^{\mathrm{LF}}$ implies that the amplitude of a prolongated smooth mode is not changed by much. For both $R_{j,\mathrm{FPA}}^{j-1}$ and $R_{j,\mathrm{BL}}^{j-1}$ holds $m_R^{\mathrm{LF}} = 2$. For $R_{j,\mathrm{FPA}}^{j-1}$, $m_R^{\mathrm{HF}} = 1$, whereas for $R_{j,\mathrm{BL}}^{j-1}$ holds $m_R^{\mathrm{HF}} = 3$.

Using (4.87), the system (RHA) decomposes into

$$\Pi_j^{\mathrm{LF}} H_j^\sigma w_j^{u,\mathrm{LF}} = \Pi_j^{\mathrm{LF}} g_j, \tag{4.91}$$

$$\Pi_j^{\mathrm{HF}} H_j^\sigma w_j^{u,\mathrm{HF}} = \Pi_j^{\mathrm{HF}} g_j \tag{4.92}$$

where $g_j = v_j^u + M_j \tilde{w}_j^p$ is the right hand side of (RHA) and $w_j^u = w_j^{u,\mathrm{LF}} + w_j^{u,\mathrm{HF}}$. The crucial idea now is to treat only (4.92) inside the smoothing iteration. On each level $j$, we iterate on (4.92) instead of on (RHA) and defer the solution of (4.91) to the next coarser level, until $w_0^{u,\mathrm{LF}}$ will be obtained by the coarse grid correction. For $j = 0$ we do not employ the splitting (4.91), (4.92), but solve the original system (RHA). Since $\Pi_j^{HF}$ has a non-trivial kernel, the system (4.92) is singular. In principle it is known that the conjugate gradient method can solve singular systems as long as the right hand side is consistent. However here, in order to avoid additional convergence difficulties due to non-symmetry and non-definiteness, we employ the more robust FGMRES method instead. This yields a stable algorithm in all our numerical experiments.

To summarize we note that the necessary modifications to $\mathcal{S}_{j,\alpha,\beta}^{\nu}$ are confined to step 3 in Algorithm 5, 67, where (RHA) is replaced by (4.92). In particular, the projection $\Pi_j^{\mathrm{HF}}$ is applied to the right hand side and in the course of the iterative process within the matrix-vector product for $\hat{H}_\sigma$, cf. Algorithm 6. Due to the approximation $\Pi_j^{\mathrm{LF}} \approx P_{j-1}^j R_j^{j-1}$ the required projections involve only a small computational overhead.

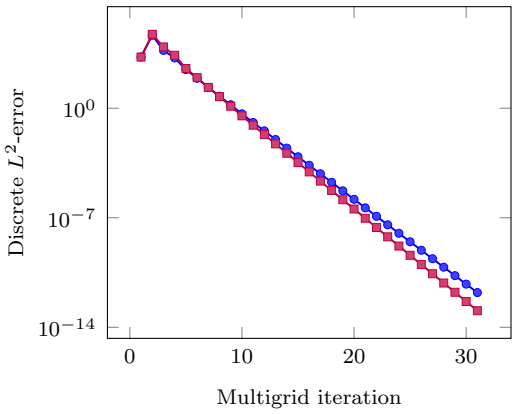



Figure 4.31: Error reduction for $\sigma = 1_{-10}$ and $J = 10$ for the cases $\Pi_j^{\mathrm{LF}} \approx P_{j-1,\mathrm{BL}}^{j} R_j^{j-1}$ with $R_j^{j-1} = R_{j,\mathrm{BL}}^{j-1}$ and $R_j^{j-1} = R_{j,\mathrm{FPA}}^{j-1}$

## 4.7 Numerical Results: Robustness Enhancement

All results presented in this section have been obtained with $\Pi_j^{\mathrm{LF}} \approx P_{j-1,\mathrm{BL}}^{j} R_{j,\mathrm{BL}}^{j-1}$. However, the effect of the larger $m_R^{\mathrm{HF}}$ for $R_{j,\mathrm{BL}}$ compared to $R_{j,\mathrm{FPA}}$ is only marginally, see Figure 4.31. There, we show the discrete $L^2$-error $e_h^m$ of the computed solution of the model problem ($\mathrm{LQP}_h$) on a grid with $h_J = 2^{-10}$ and $\sigma = 1_{-10}$. The projection $\Pi_j^{\mathrm{LF}}$ is approximated by $P_{j-1,\mathrm{BL}}^{j} R_j^{j-1}$ and we compare the two cases $R_j^{j-1} = R_{j,\mathrm{BL}}^{j-1}$ and $R_j^{j-1} = R_{j,\mathrm{FPA}}^{j-1}$. Recall that the low frequency order equals 2 for both restrictions. with $\Pi_h^{\mathrm{HF}} = P_{j-1,BL}^{j} R_{j,\mathrm{BL}}^{j-1}$.

Figure 4.32 shows the spectrum of the multigrid iteration matrix for a $W_{1,1}$-cycle on a finest mesh width $h_5 = 1/32$ and a regularization parameter $\sigma = 1_{-5}$. On the left, we see the eigenvalues for the unmodified method, which for this value of $\sigma$ is divergent. This is reflected by an eigenvalue $\mu$ with $|\mu| > 1$. Furthermore, several eigenvalues are located relatively far from the origin. Figure 4.32 on the right shows the eigenvalues for the improved method, symbolized with •, as well as those eigenvalues of the unmodified method which are located inside the shaded area of the left plot, again denoted by the symbol •. The tilted V-formation of the eigenvalues is left nearly unchanged by the frequency filtering, as these are the eigenvalues corresponding to the state and adjoint variables. However, the eigenvalues located along the axes most likely correspond to the control component, cf. Figure 4.4 on the right, where the spectrum of the smoothing iteration $\mathcal{S}_{j,1}$ was shown with direct factorization of state and adjoint unknowns. Thus, the filtering process essentially removes large eigenvalues, which correspond to high frequencies in the control unknowns, from the spectrum of the iteration matrix and leaves the rest of the spectrum nearly unchanged.

Table 4.8 lists the average error reduction factor $\varrho_{\mathrm{avg}}^{30}$ with respect to the discrete $L^2$-error $e_h^m$ obtained with the improved method. The test case is the homogeneous

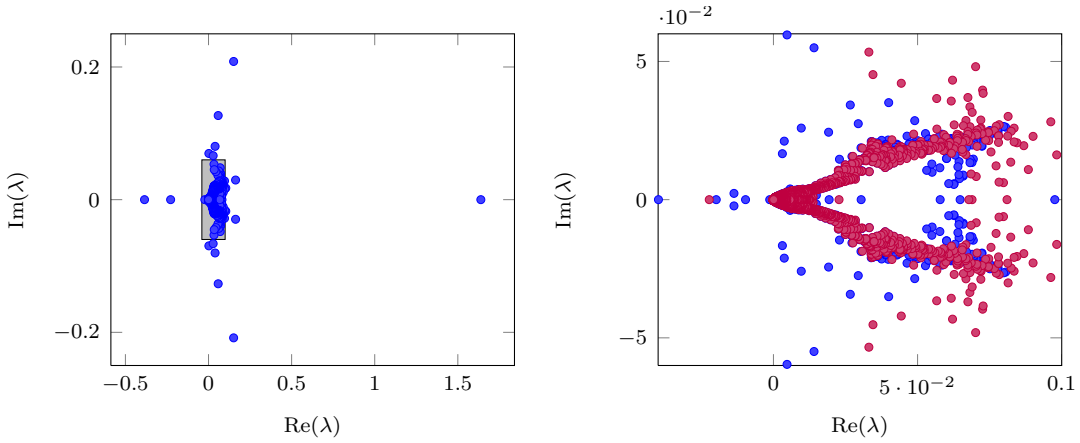



Figure 4.32: Eigenvalues of the $W_{1,1}$-cycle multigrid iteration matrix for $\sigma = 1_{-5}$, • unfiltered and • with frequency-filtering. The area shown on the left contains the complete spectrum of the unfiltered multigrid operator. The area shown on the right contains the complete spectrum of the multigrid operator employing the filtering step and corresponds to the shaded rectangle in the left figure.

linear-quadratic model problem ($\mathrm{LQP}_h$) and solutions for different values of $\sigma \to 0$ are computed. In all cases, the coarsest mesh size is $h_0 = h_c = 1/2$. Thus, the finest mesh size is $h = h_J = 2^{-J+1}$ and including the coarsest level we employ $J + 1$ levels. Note that for all values of $\sigma$ tested here the unmodified method diverges if the coarsest mesh width is $h_c = 1/2$. We tested both the $W_{1,1}$- and the $W_{2,2}$-cycle. $V$-cycle convergence is not satisfactory and for some combinations of small $\sigma$ and large $h_J$, divergence is observed. Both $W$-cycles achieve very good average convergence rates even for very small $\sigma$ down to $1_{-10}$. We see that $\varrho_{\mathrm{avg}}$ does neither depend on $J$ nor $\sigma$ for $J$ large enough. Roughly the condition $\sigma > ch_J^4$ for a constant $c < 1$ seems to hold. Recall the condition (4.58). For the improved version this condition has to hold for $J$ only and the reason is as follows: due to the inexact splitting (4.91), (4.92) the complementarity between smoothing and coarse grid correction is not perfect. Thus some modes are still not reduced quickly enough on intermediate grid levels $j < J$ and eventually have to be damped on the finest level $J$. The results suggest that these modes are ultimately responsible for the fact that the rates given in Table 4.8 are lower than the optimal rates obtained for the underlying PDE solution. Thus, similar to the case of the anisotropic problems, a further improvement can be achieved by increasing $\beta$. The corresponding results are shown in Figure 4.33, left. Here, for example the $W_{2,2}$-cycle with $\beta = 3$ or $\beta = 2$ yields the excellent rates $\rho_{\mathrm{avg}}^{30} = 7.83_{-2}$ and $\rho_{\mathrm{avg}}^{30} = 1.03_{-1}$, respectively.

We have also performed numerical experiments with the improved method applied to optimal control problems with anisotropic constraints as discussed in Sections 4.5.2, and 4.5.3. Exemplary we give the results for the model problem discretized on the

Table 4.8: Reduction factor $\varrho_{\text{avg}}^{30}$ with respect to the discrete $L^2$-error $e_h^m$ obtained with the improved method for ($\text{LQP}_h$) and $\sigma \to 0$.

| | | $J$ | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\sigma$ | | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| $1_{-10}$ | $W_{1,1}$ | $8.82_{-1}$ | $7.13_{-1}$ | $4.27_{-1}$ | $2.61_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ |
| | $W_{2,2}$ | $7.83_{-1}$ | $6.18_{-1}$ | $4.47_{-1}$ | $1.96_{-1}$ | $1.77_{-1}$ | $1.77_{-1}$ | $1.75_{-1}$ |
| $1_{-9}$ | $W_{1,1}$ | $7.73_{-1}$ | $5.14_{-1}$ | $2.68_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ |
| | $W_{2,2}$ | $6.11_{-1}$ | $4.51_{-1}$ | $2.23_{-1}$ | $1.80_{-1}$ | $1.77_{-1}$ | $1.77_{-1}$ | $1.75_{-1}$ |
| $1_{-8}$ | $W_{1,1}$ | $5.90_{-1}$ | $2.77_{-1}$ | $2.53_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ |
| | $W_{2,2}$ | $4.87_{-1}$ | $2.21_{-1}$ | $1.81_{-1}$ | $1.80_{-1}$ | $1.78_{-1}$ | $1.77_{-1}$ | $1.75_{-1}$ |
| $1_{-7}$ | $W_{1,1}$ | $2.89_{-1}$ | $2.54_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ |
| | $W_{2,2}$ | $2.47_{-1}$ | $1.83_{-1}$ | $1.81_{-1}$ | $1.81_{-1}$ | $1.78_{-1}$ | $1.77_{-1}$ | $1.75_{-1}$ |
| $1_{-6}$ | $W_{1,1}$ | $2.53_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ | $2.52_{-1}$ |
| | $W_{2,2}$ | $1.84_{-1}$ | $1.78_{-1}$ | $1.80_{-1}$ | $1.81_{-1}$ | $1.78_{-1}$ | $1.77_{-1}$ | $1.76_{-1}$ |

deformed unit square $G_\delta$ with $\delta = 0.25$ (see Figure 4.15) and $\sigma = 1_{-8}$. The coarsest mesh is again $h_c = 1/2$. We employed $\mathcal{S}_{j,1,2}^{\nu}$ with ALGS as constraint smoother. Furthermore, we used the $W_{1,1}$-cycle as preconditioner for a GMRES(15) iteration. The iteration history with respect to $e_h^m$ is shown in Figure 4.33 on the right and the mesh-independent convergence speed is readily observed. For comparison, we show the convergence of the $W_{1,1}$-cycle when used as a solver for the case $J = 10$. The Krylov acceleration yields a notable improvement compared to the unaccelerated version. The average reduction rates of the unaccelerated and the accelerated method are $2.68_{-1}$ and $1.41_{-1}$, respectively. These results show that in principle, the improved method is suitable also for the anisotropic problems discussed earlier. However we note that optimal rates without Krylov acceleration can only be expected if the anisotropies are reflected in (4.87) by e.g. employing operator-dependent interpolation operators.

Last but not least we remark that the results displayed in Figure 4.22 have been computed with the improved method employing the frequency filtering.

## Summary

We have developed a multigrid method which, using the full multigrid framework, solves discretized PDE constrained optimization problems with optimal cost $\mathcal{O}(n)$, where $n = 3h_J^{-2}$ is the total number of unknowns of the corresponding KKT system on the finest grid level $J$. The smoothing iteration is defined as an inexact variant of a constraint preconditioner, building on the results discussed in Section 3.2. A local mode analysis has shown that for discrete optimality systems smoothing rates can be

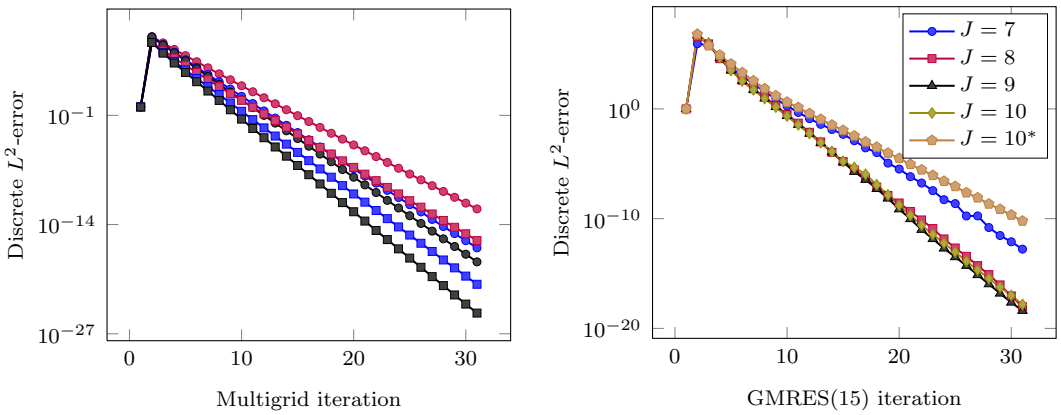



Figure 4.33: Left: Error reduction for $(\mathrm{LQP}_h)$ with $J = 10$, $\sigma = 1_{-10}$ and — $\beta = 1$, — $\beta = 2$, — $\beta = 3$ with • the $W_{1,1}$-cycle and ▪ the $W_{2,2}$-cycle. Right: Error reduction for test problem with $G_\delta, \delta = 0.25$, $\sigma = 1_{-8}$ and the $W_{1,1}$-cycle as preconditioner for GMRES(15). For comparison, for $J = 10$ the $W_{1,1}$-cycle is used as solver (marked with $^*$).

expected that are close to those of the underlying constraint PDE, at least for moderate values of the regularization parameter $\sigma$. These findings have been corroborated with detailed numerical experiments including anisotropic diffusion and convection-diffusion with a strong convection term as constraint, all of which require smoothing iterations more elaborate than pointwise Gauss-Seidel at the constraint level. Thus the flexibility of our approach, allowing to exploit properties of the constraint PDE, has been demonstrated. In all cases it has been clearly shown that discrete optimality systems can be solved with a small multiple of the computational cost required to solve the underlying constraint PDE. In the full multigrid framework the constant of proportionality has been found to range between 8 and 10.

Furthermore, the role of the regularization parameter $\sigma$ has been discussed in some detail. It has been shown that independent of $J$ and $\sigma$, but with a restriction on $h_1$, robust convergence rates are obtained. By a close inspection of the smoothing iteration with respect to the control unknown $u_h$, an improvement based on spectral filtering of the reduced Hessian $H_j^\sigma = \sigma M_j + S_j^* M_j S_j$ could be achieved. With minimal additional computational work, the (RHA)-step in Algorithm 5 is restricted to the high frequency components of the error, thus restoring complementarity of the smoothing step and the coarse grid correction. As a result, the restriction on the mesh size $h_1$ is removed. Numerical experiments show that for small enough $h_J$, robust convergence rates are obtained, which do not depend on the regularization parameter $\sigma$, the coarsest mesh size $h_0$ and the number of levels $J$.

In the two subsequent chapters the method presented here will be employed at the core of multigrid-based solution methods for control-constrained problems as well as semilinear-constrained problems.

# 5 A Primal-Dual Active-Set Multigrid Method for Control-Constrained Optimal Control Problems

In this chapter we consider optimal control problems with additional inequality constraints imposed on the control unknown $u$ and for their efficient solution we combine a primal-dual active-set strategy with the multigrid method developed in the previous chapter. Control-constraints are specified by the condition $u \in \mathcal{U}_{\text{ad}}$, where the set of *admissible controls* $\mathcal{U}_{\text{ad}} \subset L^2(\Omega)$ is a proper subset of $L^2(\Omega)$ and is assumed to be closed and convex. In particular, we consider the problem

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}(y,u) \\ \text{subject to} \quad & \mathcal{C}(y,u) = 0 \\ \text{and} \quad & u \in \mathcal{U}_{\text{ad}}, \end{aligned} \tag{CC}$$

where $\mathcal{J}$ and $\mathcal{C}$ are as in Example 2.12, i.e. $\mathcal{J}$ is the tracking-type functional (2.1) and the constraints $\mathcal{C}$ are given by the second-order linear elliptic partial differential equation (2.24). The set $\mathcal{U}_{\text{ad}}$ is defined by so-called *box-constraints*

$$u \in \mathcal{U}_{\text{ad}} = \{u \in L^2(\Omega) \mid u^\alpha \leq u \leq u^\beta \text{ a.e. in } \Omega\}, \tag{5.1}$$

with given functions $u^\alpha, u^\beta \in L^\infty(\Omega), u^\alpha \leq u^\beta$ a.e. in $\Omega$. Obviously the set $\mathcal{U}_{\text{ad}}$ defined by (5.1) is closed and convex. Existence and uniqueness of a solution then follows from Theorem 2.4. Theorem 2.8 yields the first-order conditions with the corresponding optimality system (OS), where the optimality condition is given by (2.17).

*Remark* 5.1 (Regularity of the Optimal Control). For $\mathcal{U}_{\text{ad}} = \{u \in L^2(\Omega) \mid u \geq 0 \text{ a.e. in } \Omega\}$ one obtains $u \in H_0^1(\Omega)$. In the case of the box-constraints (5.1) one obtains $u \in H^1(\Omega)$ if the bounding functions $u^\alpha, u^\beta$ are sufficiently regular, in particular $u^\alpha, u^\beta \in L^\infty(\Omega) \cap H^1(\Omega)$. Without additional assumptions on the regularity of $u^\alpha, u^\beta$ however, we can only expect $u \in L^2(\Omega)$, cf. [111, Ch. II, Remark 2.3].

## 5.1 Finite Dimensional Approximation

The approach most common for the finite dimensional approximation of control-constrained optimal control problems is the discretize-then-optimize methodology. To this end, the objective functional $\mathcal{J}$, the constraints $\mathcal{C}$ and the admissible set $\mathcal{U}_{\text{ad}}$

viz. the bounding functions $u^\alpha, u^\beta$ are approximated by discrete versions. Existence and uniqueness of a solution $y_h^*, u_h^*$ to the optimization problem in finite dimension immediately follows. We remark that several algorithms can be formulated in an appropriate function space setting and can be applied directly to (OS). A resulting sequence of infinite-dimensional systems then has to be discretized. Convergence can be proved using the interpretation as a semismooth Newton method, we refer to [91, 148] and to [92] for a mesh-independence result. In the present setting however, where we focus on devising an efficient algorithm, a discrete version of (OS) is the appropriate starting point and will be derived now. Discretizing $\mathcal{C}$ as in Section 2.3.2, we obtain the discretized state equation (2.50f) and the corresponding discrete adjoint equation (2.50b). It remains to discretize (5.1) and (2.17), which in the concrete setting is stated in (2.19). To this end, we discretize the bounding functions $u^\alpha, u^\beta$ by piecewise constants and obtain the discrete approximation to the set of admissible controls $\mathcal{U}_{\text{ad}}$,

$$\mathcal{U}_{\text{ad},h} = \left\{ v_h \in U_h \mid u_h^\alpha \leq v_h \leq u_h^\beta \right\}. \tag{5.2}$$

The $L^2$-inner products appearing in the variational inequality are discretized employing midpoint quadrature, resulting in the same mass matrix $M_h$ as in (2.50f), (2.50b). We then obtain

$$\left(\sigma M_h u_h - M_h p_h\right)^T (v_h - u_h) \geq 0, \quad v_h \in \mathcal{U}_{\text{ad},h} \tag{5.3}$$

as discrete optimality condition. The complete discrete optimality system corresponding to (OS), (2.17) is given by (2.50b), (2.50f) and (5.3). Error estimates have been given in Section 2.3.3.

## 5.2 Multigrid Methods for Variational Inequalities

The two most prominent methods to treat optimization problems with inequality constraints are *interior-point methods* and *active set strategies*. Both approaches yield a sequence of equality-constrained quadratic programming problems (QP) and therefore rely on an efficient solution method for such QPs. Currently the state-of-the-art methods discussed in Chapter 3 are frequently employed for their solution and the bulk of numerical methods is made up by their combination with interior-point or active-set methods. Only a few publications consider the application of multigrid within the context of inequality-constrained PDE constrained optimization problems.

The interior-point approach aims at maintaining the complementarity condition at all intermediate steps and introduces (logarithmic) barrier functions. As a result, the QPs can become increasingly ill-conditioned when the interior-point algorithm approaches the solution. The goal of active set methods is to enforce the feasibility of intermediate iterates with respect to the constraints. To this end, at each step the constraints are partitioned into sets of active and inactive constraints with respect to

the current iterate, hence the name of the method. A particular class of active-set methods is the primal-dual active-set (PDAS) strategy [24, 91]. The PDAS method is closely related to the projected Newton method of [25], and can be formulated as a semismooth Newton method in infinite dimension. As such, the PDAS approach shares important properties with Newton methods, namely superlinear or quadratic local convergence and a mesh-independence principle [92]. Therefore, we expect an inner-outer iterative method, defined by the combination of our multigrid method for the solution of the QPs and the semismooth Newton method to generate the QPs, to be an efficient and mesh-independent solution method for PDE constrained optimization problems with inequality constraints on the control. Furthermore, for control-constrained optimal control problems, in [23] the PDAS approach is shown to be more efficient than interior-point methods. For the combination of multigrid and interior-point methods, albeit not in the context of optimal control, we refer to [149]. Combinations of active-set strategies with multigrid have been proposed in [87] and later in [98] for minimal surface and other obstacle problems. In case the active and inactive sets generated by the PDAS method and the approach in [98], both outer iterations yield the same inner linear system. However, in contrast to [98], which is a primal method only, in [91] dual information is used to predict the active set.

In contrast to inner-outer iterative schemes one can define a variant of a multigrid method which treats the inequality constraints within the smoothing iteration. In the context of a semismooth Newton method, this amounts to a local linearization process in the smoother instead of a global one in an outer iteration. Thus, this multigrid approach is related to the Full Storage Approximation (FAS) scheme [40], which was mainly developed to treat nonlinear problems. In [42], the projected FAS (PFAS) has been developed and applied to complementarity formulations of free-boundary problems. In [33, 35], the collective Gauss-Seidel smoother of the CGSM multigrid approach (cf. Section 4.2) has been augmented by a local projection step $u_h = \Pi^{\mathrm{loc}}_{\mathcal{U}_{\mathrm{ad}}} \tilde{u}_h$ to allow for inequality constraints on $u_h$. To our knowledge, this is the only published application of a PFAS-like method in the context of PDE constrained optimization. Related approaches such as the monotone multigrid method [107], which also employ projected Gauss-Seidel smoothing, are applied mostly to variational inequalities derived from obstacle- and contact-problems in elasticity and mechanics.

A major advantage of treating the inequality constraints in an outer iteration is as follows: The resulting inner subsystem and in particular the smoother of a multigrid method applied there does not need to take the constraints into account. Thus, the inner systems to be solved are strictly linear and equality-constrained and the multigrid method of Chapter 4 can be applied as QP subsolver with only minor modifications. The connection to semismooth Newton methods lets us expect superlinear convergence and thus a fast detection of the active sets within the outer iteration.[1] In contrast, the

[1] In this respect the PDAS method differs significantly from primal active-set methods like the Simplex algorithm since many constraints per iteration can be identified.

next-neighbor related identification of active nodes within a projected Gauss-Seidel smoother can lead to slower identification of the active set.

## 5.3 The Primal-Dual Active Set Method

In this section we briefly describe the primal-dual active set strategy that will be used as outer iteration to handle the control-constraints [23, 24, 91]. To this end, we note that for the discrete variational inequality (5.3), an equivalent formulation is given by

$$\begin{aligned} \sigma M_h u_h - M_h p_h + \lambda &= 0, \\ \lambda = \max(\lambda + c(u_h - u_h^\beta), 0) + \min(\lambda + c(u_h - u_h^\alpha), 0), &\quad c > 0. \end{aligned} \tag{5.4}$$

Here, the unknowns $\lambda$ are the Lagrange multipliers associated with the inequality constraints. They satisfy the Karush-Kuhn-Tucker conditions

$$\begin{aligned} &\lambda \le 0 \ \text{ on } \ \mathcal{T}_{h,\mathcal{A}_-^*} = \{T_i \in \mathcal{T}_h \mid i \in \mathcal{A}_-^*\}, \ \mathcal{A}_-^* = \{i \mid u_h^* = u_h^\alpha \text{ on } T_i\}, \\ &\lambda \ge 0 \ \text{ on } \ \mathcal{T}_{h,\mathcal{A}_+^*} = \{T_i \in \mathcal{T}_h \mid i \in \mathcal{A}_+^*\}, \ \mathcal{A}_+^* = \{i \mid u_h^* = u_h^\beta \text{ on } T_i\}, \\ &\lambda = 0 \ \text{ on } \ \mathcal{T}_{h,\mathcal{I}^*} = \{T_i \in \mathcal{T}_h \mid i \in \mathcal{A}_-^*\}, \ \mathcal{I}^* = \{i \mid u_h^\alpha < u_h^* < u_h^\beta \text{ on } T_i\}. \end{aligned} \tag{5.5}$$

Here, $\mathcal{A}_-^*$ and $\mathcal{A}_+^*$ are the active sets and $\mathcal{I}^*$ is the inactive set at the (discrete) optimal solution $u_h^*$. To unburden the notation, active and inactive sets are not designated with the discretization index $h$, however, they always refer to the discrete unknowns.

The primal-dual active set strategy is an iterative algorithm that makes use of (5.4) to predict the active and inactive sets and treats an associated equality constrained optimization problem at each step. This leads to Algorithm 7. Note that for the case without control constraints, we have $\mathcal{A}_- = \mathcal{A}_+ = \emptyset$ and the overall algorithm reduces to just the solution of (EQP), which in turn reduces to the saddle point system (2.51). This concludes the description of the PDAS method. For details and convergence properties we refer to [24]. In particular, it is proved there that, if Algorithm 7 stops due to the criterion in line 4, the solution of (EQP) is the solution of the original optimality system. No additional stopping criterion was required in our implementation, in our numerical experiments Algorithm 7 always stopped due to the rule in line 4.

The main computational effort in this algorithm has to be spent for the solution of (EQP). In most publications this is done by solving the associated reduced system for the control unknowns $u_h^k$ by a conjugate gradient method, i.e. the methods discussed in Chapter 3 are applied to (EQP). In our context it is natural to employ the multigrid method developed in the preceding chapter for the solution of (EQP). To this end, we modify the system (EQP) in such a way that it can be formulated as a KKT system (2.51). We proceed as follows: First, we partition the control unknowns according to $u_h^k = [u_h^{\mathcal{I}^k} \ u_h^{\mathcal{A}_-^k} \ u_h^{\mathcal{A}_+^k}]$. The same partitioning applies to the Lagrange multipliers $\lambda^k = [\lambda^{\mathcal{I}^k} \ \lambda^{\mathcal{A}_-^k} \ \lambda^{\mathcal{A}_+^k}]$. Note that this partitioning induces corresponding $3 \times 3$

The Outer PDAS Iteration

1: Choose initial values $y_h^0, u_h^0, p_h^0, \lambda^0$ and set $k = 1$
2: **while** not converged **do**
3: predict $\mathcal{A}_-^k, \mathcal{A}_+^k, \mathcal{I}^k$ as follows:

$$\mathcal{A}_-^k = \{i \mid u_h^{k-1} + \frac{\lambda^{k-1}}{\sigma} < u_h^\alpha \text{ on } T_i\} \tag{5.6a}$$

$$\mathcal{A}_+^k = \{i \mid u_h^{k-1} + \frac{\lambda^{k-1}}{\sigma} > u_h^\beta \text{ on } T_i\} \tag{5.6b}$$

$$\mathcal{I}^k = \{i \mid i \notin \mathcal{A}_-^k \cup \mathcal{A}_+^k\} \tag{5.6c}$$

4: **if** $k \geq 2$ and $\mathcal{A}_-^k = \mathcal{A}_-^{k-1}, \mathcal{A}_+^k = \mathcal{A}_+^{k-1}, \mathcal{I}^k = \mathcal{I}^{k-1}$ **then**
5: converged = true
6: **else**
7: solve the equality-constrained problem

$$\begin{aligned} M_h y_h^k + L_h^T p_h^k &= M_h \bar{y}_h \\ \sigma M_h u_h^k - M_h^T p_h^k + \lambda^k &= 0 \\ L_h y_h^k - M_h u_h^k &= M_h f_h \\ \lambda^k &= 0 \qquad \text{on } \mathcal{T}_{h,\mathcal{I}^k} \\ u_h^k &= u_h^\alpha \qquad \text{on } \mathcal{T}_{h,\mathcal{A}_-^k} \\ u_h^k &= u_h^\beta \qquad \text{on } \mathcal{T}_{h,\mathcal{A}_+^k} \end{aligned} \tag{EQP}$$

8: $k = k + 1$

Algorithm 7: The Primal-Dual Active-Set Strategy as outer iteration.

block, $3 \times 1$ column and $1 \times 3$ row block partitions of the mass matrix $M_h$. Then, the system given by the first three lines in (EQP) can be written as

$$\left[\begin{array}{c|ccc|c} M_h & & & & L_h^T \\ \hline & \sigma M_h^{\mathcal{I}^k,\mathcal{I}^k} & \sigma M_h^{\mathcal{I}^k,\mathcal{A}_-^k} & \sigma M_h^{\mathcal{I}^k,\mathcal{A}_+^k} & -M_h^{\mathcal{I}^k,*} \\ & \sigma M_h^{\mathcal{A}_-^k,\mathcal{I}^k} & \sigma M_h^{\mathcal{A}_-^k,\mathcal{A}_-^k} & \sigma M_h^{\mathcal{A}_-^k,\mathcal{A}_+^k} & -M_h^{\mathcal{A}_-^k,*} \\ & \sigma M_h^{\mathcal{A}_+^k,\mathcal{I}^k} & \sigma M_h^{\mathcal{A}_+^k,\mathcal{A}_-^k} & \sigma M_h^{\mathcal{A}_+^k,\mathcal{A}_+^k} & -M_h^{\mathcal{A}_+^k,*} \\ \hline L_h & -M_h^{*,\mathcal{I}^k} & -M_h^{*,\mathcal{A}_-^k} & -M_h^{*,\mathcal{A}_+^k} & \end{array}\right] \left[\begin{array}{c} y_h^k \\ \hline u_h^{\mathcal{I}^k} \\ u_h^{\mathcal{A}_-^k} \\ u_h^{\mathcal{A}_+^k} \\ \hline p_h^k \end{array}\right] = \left[\begin{array}{c} M_h \bar{y}_h \\ \hline -\lambda^{\mathcal{I}^k} \\ -\lambda^{\mathcal{A}_-^k} \\ -\lambda^{\mathcal{A}_+^k} \\ \hline M_h f_h \end{array}\right]. \tag{5.7}$$

Now we utilize the last three equations in (EQP) to reduce (5.7) to a system for

$y_h^k, u_h^{\mathcal{I}_k}, p_h^k$, i.e. we eliminate $u_h^{\mathcal{A}_-^k}, u_h^{\mathcal{A}_+^k}$ and we consider the controls $u_h^k$ only on the inactive set $\mathcal{I}^k$. The solution of (EQP) then proceeds in two steps: First, the saddle point system

$$K_h^{\mathcal{I}^k} x_h^{\mathcal{I}^k} = r_h^{\mathcal{I}^k} \tag{5.8}$$

has to be solved, where

$$K_h^{\mathcal{I}^k} = \begin{bmatrix} M_h & & L_h^T \\ & \sigma M_h^{\mathcal{I}^k,\mathcal{I}^k} & -M_h^{\mathcal{I}^k,*} \\ L_h & -M_h^{*,\mathcal{I}^k} & \end{bmatrix}, \quad r_h^{\mathcal{I}^k} = \begin{bmatrix} M_h \bar{y}_h \\ -\sigma M_h^{\mathcal{I}^k,\mathcal{A}_-^k} u_h^\alpha - \sigma M_h^{\mathcal{I}^k,\mathcal{A}_+^k} u_h^\beta \\ M_h f_h + M_h^{*,\mathcal{A}_-^k} u_h^\alpha + M_h^{*,\mathcal{A}_+^k} u_h^\beta \end{bmatrix}, \tag{5.9}$$

and the vector of unknowns is given by $x_h^{\mathcal{I}^k} = [y_h^k \; u_h^{\mathcal{I}^k} \; p_h^k]$. Note that the KKT operator $K_h^{\mathcal{I}^k}$ and the right hand side vector $r_h^{\mathcal{I}^k}$ depend on the index $k$ of the outer iteration. In the second step, the Lagrange multipliers $\lambda^k$ are computed by

$$\begin{aligned} \lambda^{\mathcal{A}_-^k} &= M_h^{\mathcal{A}_-^k,*} p_h^k - \sigma M_h^{\mathcal{A}_-^k,\mathcal{I}^k} u_h^{\mathcal{I}^k} - \sigma M_h^{\mathcal{A}_-^k,\mathcal{A}_-^k} u_h^\alpha - \sigma M_h^{\mathcal{A}_-^k,\mathcal{A}_+^k} u_h^\beta, \\ \lambda^{\mathcal{A}_+^k} &= M_h^{\mathcal{A}_+^k,*} p_h^k - \sigma M_h^{\mathcal{A}_+^k,\mathcal{I}^k} u_h^{\mathcal{I}^k} - \sigma M_h^{\mathcal{A}_+^k,\mathcal{A}_-^k} u_h^\alpha - \sigma M_h^{\mathcal{A}_+^k,\mathcal{A}_+^k} u_h^\beta, \end{aligned} \tag{5.10}$$

compare the lines 3 and 4 in (5.7). On the inactive set, we just set $\lambda^{\mathcal{I}^k} = 0$. Note again that, for the case without control constraints, the system (5.8) reduces to (2.51), i.e. we just have $K_h^{\mathcal{I}^k} = K_h$, $r_h^{\mathcal{I}^k} = b_h$ and $x_h^{\mathcal{I}^k} = x_h$. Finally, we remark that in the context of a projected Newton method the operator $K_h^{\mathcal{I}}$ corresponds to the projected[2] Hessian [25].

## 5.4 A PDAS Multigrid Method for the Solution of Control-Constrained Optimal Control Problems

It remains to obtain the solution of (5.8) in a fast and efficient fashion. To this end, we proceed by modifying the multigrid method of Chapter 4. For a fixed index $k$ we set $\mathcal{I} = \mathcal{I}^k$ and define the smoothing iteration by the modification

$$w_j^{\mathcal{I},r+1} = w_j^{\mathcal{I},r} + (B_{K_j}^{\mathcal{I}})^{-1}(b_j^{\mathcal{I}} - K_j^{\mathcal{I}} w_j^{\mathcal{I},r}). \tag{5.11}$$

of iteration (4.23). The inexact constraint preconditioner (4.24) now takes the form

$$B_{K_j}^{\mathcal{I}} = \begin{bmatrix} & & \hat{L}_j^T \\ & \hat{H}_{Z_j}^{\mathcal{I}} & -M_j^{\mathcal{I},*} \\ \hat{L}_j & -M_j^{*,\mathcal{I}} & \end{bmatrix}, \tag{5.12}$$

[2] In the literature, the name reduced Hessian is often used and in this context is meant in the sense of reducing to inactive constraints, not to be confused with the reduced Hessian $H_Z$ in the sense of Chapters 3 and 4.

with the modified reduced Hessian given by

$$\hat{H}^{\mathcal{I}}_{Z_j} = M_j^{\mathcal{I},*} \hat{L}_j^{-T} M_j \hat{L}_j^{-1} M_j^{*,\mathcal{I}} + \sigma M_j^{\mathcal{I},\mathcal{I}} \tag{5.13}$$

instead of (4.26). Using (5.11)–(5.13), the smoothing step then follows from Algorithms 5 and 6 and is denoted in compact form as

$$\tilde{w}_j^{\mathcal{I}} = (\mathcal{S}^{\mathcal{I}}_{j,\alpha,\beta})^{\nu}(w_j^{\mathcal{I}}, b_j^{\mathcal{I}}) \tag{5.14}$$

with the same parameters $\nu, \alpha, \beta$ as for (4.27).

From the two-step solution of (EQP) it follows that the Lagrange multipliers $\lambda$ as well as the bounding functions $u^\alpha$ and $u^\beta$ need to be discretized on the finest grid level only. However, for the discretization of $x^{\mathcal{I}}_{h_j}, b^{\mathcal{I}}_{h_j}$ and the operator $K^{\mathcal{I}}_{h_j}$ in (5.8) on a grid level $j < J$, it is evident that we have to approximate the inactive set $\mathcal{I}$ on that grid level. This has to be done in each PDAS iteration, after the inactive and active set on the finest level $J$ have been detected by the algorithm and before the solution of the system (5.8). In the context of node-based discretizations for obstacle problems, two different strategies have been previously used. In [87], the inactive set shrinks when reduced to coarser grids and [98] later followed the same strategy. In [42], the size of the inactive set increases with reduced level index. The difficulty of representing $\mathcal{I}$ for levels $j < J$ has been circumvented altogether in [30] by employing cascadic multigrid. Here, the iteration starts on the coarsest level $j = 0$, and never returns to the coarser grids. On each grid level, the active and inactive sets are determined based on the current approximation. This leads to a fast detection of $\mathcal{I}$ due to the nested iteration approach, however the convergence speed of cascadic multigrid suffers from the non-existent coarse grid correction. A further strategy is the modification of coarse grid basis functions which is employed within monotone multigrid.

Consistent with the cell-centered discretization, here we proceed as follows: The third step of the PDAS algorithm yields index sets $\mathcal{I}$, $\mathcal{A}_-$ and $\mathcal{A}_+$ on the finest level $J$ and we denote the set of grid cells corresponding to $\mathcal{I}$ with $\mathcal{T}_{\mathcal{I}_J}$. Now, for given $\mathcal{T}_{\mathcal{I}_j}$ we define $\mathcal{T}_{\mathcal{I}_{j-1}}$ as the set of all coarser grid cells for which at least one fine-grid subcell is contained in $\mathcal{T}_{\mathcal{I}_j}$, i.e.

$$\mathcal{T}_{\mathcal{I}_{j-1}} = \big\{ T_i \in \mathcal{T}_{j-1} \mid T_i^s \in \mathcal{T}_{\mathcal{I}_j} \text{ for } s \in \{1,2,3,4\} \big\}, \quad j = J, \dots, 1. \tag{5.15}$$

In our numerical experiments, this "outer" approximation of $\mathcal{I}$ on coarser levels has shown to yield faster convergence than the alternative "inner" approximation. Most likely the difference is due to the coarse grid correction: when $\mathcal{I}$ shrinks with decreasing $j$, no coarse grid correction occurs for the current iterate on $\mathcal{I}_j \backslash \mathcal{I}_{j-1}$, and the error on $\mathcal{I}_j \backslash \mathcal{I}_{j-1}$ has to be improved exclusively by the smoothing iteration.

In any case, the coarsest grid has to be fine enough to avoid $\mathcal{A}_{-,0} = \mathcal{A}_{+,0} = \emptyset$, since otherwise the bounds $u^\alpha, u^\beta$ would be completely removed from the coarse grid correction process which necessarily leads to degradation of convergence. Note that

no representation of the active sets $\mathcal{A}_-$ and $\mathcal{A}_+$ is needed on coarser levels. From the sequence of meshes given by (5.15) we then obtain a sequence of operators $K_j^{\mathcal{I}}$ as before by direct discretization.

The intergrid transfer operators follow from (4.60) with minor modifications with respect to the control component $u_j^{\mathcal{I}}$. They are given by

$$\mathcal{R}_j^{j-1} = \begin{pmatrix} R_j^{j-1} & & \\ & R_j^{j-1,\mathcal{I}} & \\ & & R_j^{j-1} \end{pmatrix}, \quad \mathcal{P}_{j-1}^{j} = \begin{pmatrix} P_{j-1}^{j} & & \\ & P_{j-1}^{j,\mathcal{I}} & \\ & & P_{j-1}^{j} \end{pmatrix}. \tag{5.16}$$

In (5.16), the symbol $R_j^{j-1,\mathcal{I}}$ denotes the four-point average operator (4.12) giving values only for grid cells $T_i \in \mathcal{T}_{\mathcal{I}_{j-1}}$. When applying $R_j^{j-1,\mathcal{I}}$ to obtain a coarse grid value of the $u_{j-1}^{\mathcal{I}}$-component, fine grid values on the active set $\mathcal{T}_j \backslash \mathcal{T}_{\mathcal{I}_j}$ could be needed. However, on $\mathcal{A}_-$ and $\mathcal{A}_+$ the solution is fixed to the constraints $u^\alpha, u^\beta$, respectively and the corresponding residuals vanish. Thus, active nodes should provide no contribution here and consistently the corresponding stencil entries are set to zero. Similar considerations apply to the prolongation $P_{j-1}^{j,\mathcal{I}}$. Here, coarse grid values on the active set $\mathcal{T}_{j-1} \backslash \mathcal{T}_{\mathcal{I}_{j-1}}$ could enter the prolongation stencil when computing the correction for a fine grid value of $u_j^{\mathcal{I}}$. However again the corrections due to active nodes should be zero and the corresponding stencil entries are set to zero. With these modifications, the multigrid Algorithm 3 can be applied for the solution of (5.8), which concludes the description of the PDAS-multigrid method.

Some additional remarks are in order when using the full multigrid Algorithm 4 for the solution of (5.8). First we note that now the bounds $u^\alpha, u^\beta$ need to be discretized on all levels $0 \le j \le J$ for two reasons: first the right hand side vector $r_J^{\mathcal{I}}$ (5.9) needs to be constructed on all levels $j < J$, and second, the FMG prolongation $\tilde{\mathcal{P}}_{j-1}^{j}$ transfers the solution $u_j^{\mathcal{I}}$ which requires that the actual values of $u_j$ on $\mathcal{A}_-$ and $\mathcal{A}_+$ need to be taken into account. For these reasons, additionally the active sets $\mathcal{A}_-, \mathcal{A}_+$ need to be represented on each grid level. To this end, instead of proceeding as in (5.15), we restrict $\mathcal{A}_-$ and $\mathcal{A}_+$ according to

$$\mathcal{T}_{\mathcal{A}_{\pm,j-1}} = \left\{ T_i \in \mathcal{T}_{j-1} \mid \cup T_i^s \in \mathcal{T}_{\mathcal{A}_{\pm,j}} \text{ for } s \in \{1,2,3,4\} \right\}, \quad j = J, \dots, 1, \tag{5.17}$$

where $\mathcal{A}_\pm$ stands for $\mathcal{A}_-$ or $\mathcal{A}_+$. After the restriction step (5.17) we set

$$\mathcal{A}_{j-1} = \mathcal{A}_{-,j-1} \cup \mathcal{A}_{+,j-1} \text{ and } \mathcal{T}_{\mathcal{I}_{j-1}} = \mathcal{T}_{j-1} \backslash \mathcal{T}_{\mathcal{A}_{j-1}}. \tag{5.18}$$

Recall that for the conventional multigrid cycle only the sequence $\mathcal{T}_{\mathcal{I}_j}, j = J, \dots, 0$ was needed.

We note that this strategy differs from other nested iteration approaches to obstacle problems in the following sense: Here, we employ the full multigrid with the purpose of solving (5.8) for *fixed* sets $\mathcal{I}_J$ and $\mathcal{A}_{-,J}, \mathcal{A}_{+,J}$, as provided by the outer PDAS iteration on level $J$. From results presented in Section 4.5.1 we can expect that this is achieved

with optimal complexity. We do not employ the nested iteration strategy to change the current predictions of $\mathcal{I}_J$ and $\mathcal{A}_{-,J}, \mathcal{A}_{+,J}$. The latter strategy would be natural when using an FAS-based method.

## 5.5 Numerical Results

In this section we conduct several numerical experiments to test the convergence and efficiency of the proposed PDAS-multigrid method. We consider the model problem ($\mathrm{LQP}_h$) with different lower and upper bounds $u_J^\alpha, u_J^\beta$. The convergence factor of the inner multigrid iteration is measured analogously to Section 4.5.1. Here, $\varrho^m$ is based on the fine grid residual $res_J^{\mathcal{I}^k,m}$ of the EQP (5.8) which is defined as

$$res_J^{\mathcal{I}^k,m} = r_J^{\mathcal{I}^k} - K_J^{\mathcal{I}^k} x_J^{\mathcal{I}^k,m} \tag{5.19}$$

for each index $k$ of the outer PDAS iteration. As before, the superscript $m$ is the index of the multigrid iteration. With respect to the PDAS iteration, we use the fact that the optimal control $u^*$ satisfies the projection formula

$$u^* = \Pi_{\mathcal{U}_{\mathrm{ad}}}(\frac{1}{\sigma} p^*) \tag{5.20}$$

to construct an exact solution $u^*$ and measure the error $e_{u_J}^k$ in the discrete $L^2$-norm, cf. (3.24). Additionally we consider the violation of the bounds

$$e_\alpha^{u^k} = \max_{T\in\mathcal{T}_J}\big(u_J^\alpha - u_J^k\big), \quad e_\beta^{u^k} = \max_{T\in\mathcal{T}_J}\big(u_J^k - u_J^\beta\big). \tag{5.21}$$

For the multigrid solution of (5.8) we employ the smoothing iteration $\mathcal{S}_{j,1,1}^\nu$ and use GS-LEX as constraint smoother. Unless noted otherwise, the inner iterations are stopped as soon as $res_J^{\mathcal{I}^k,m} \le \max\big(1_{-16} res_J^{\mathcal{I}^k,0}, 1_{-12}\big)$ or $m = 20$. Furthermore, the coarse grid size is set to $h_0 = 1/8$.

In [23] different strategies are discussed for the initialization of the PDAS algorithm. It was noted that the algorithm is rather insensitive to the initial value $x_J^0$. A special initialization proposed requires the solution of the state and adjoint equation for a given feasible initial control $u_J^0$. The second strategy consists of solving the unconstrained problem and we will adopt this approach. To this end, we use a general initial value $x_J^0$ (cf. Section 4.5.1) and set

$$\mathcal{T}_{\mathcal{I}_J} = \mathcal{T}_J, \quad \mathcal{A}^0 = \emptyset, \quad \lambda \equiv 0. \tag{5.22}$$

With these choices, the first PDAS iteration yields the solution of the unconstrained problem since (5.8) reduces to (2.51).

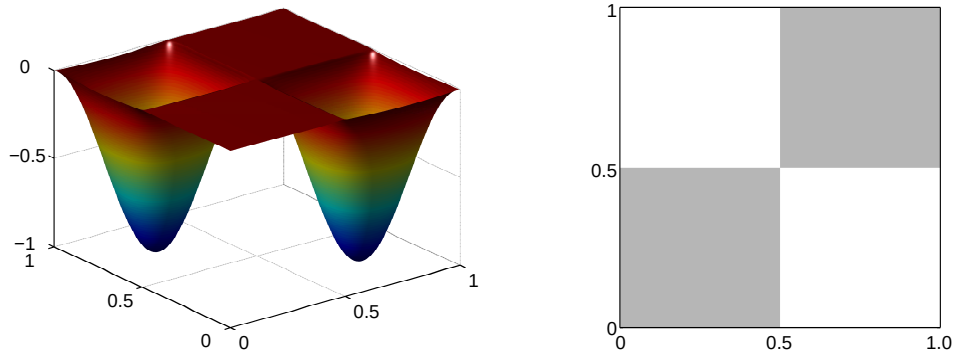



Figure 5.1: Computed optimal control $u_J^*$ (left) and corresponding active set $\mathcal{A}_+^*$ (shaded region, right) on a mesh with $h_J = 2^{-8}$ for ($\mathrm{LQP}_h$) with control-constraints (5.23).

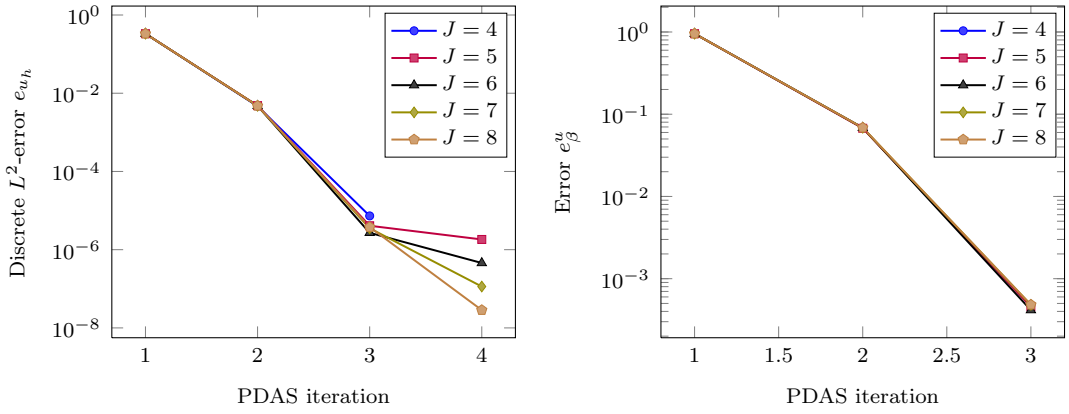



Figure 5.2: Discrete $L^2$-error of the control $u_J$ (left) and error $e_u^\beta$ (right) for the outer PDAS iteration and different number of levels $J$ with $h_J = 2^{-(J+3)}$.

### 5.5.1 A Model Problem

First we consider the unilaterally constrained problem with

$$u \leq u^\beta = 0. \tag{5.23}$$

The computed optimal control $u_J^*$ and the corresponding active set $\mathcal{A}^*$ are depicted in Figure 5.1 on a mesh with $h_J = 2^{-8}$. Figure 5.2 shows the iteration history of the outer PDAS iteration for different levels $J$ with a mesh size of the fine grid given by $h_J = 2^{-(J+3)}$. On the left, we plot the discrete $L^2$-error of the control $u_J$ and on the right we show the error $e_u^\beta$. Note that for $k = 4$ the computed solution satisfies $e_u^\beta = 0$. Further recall that in the first iteration, the unconstrained problem is solved. Thus, $\mathcal{I}^*$ and $\mathcal{A}^*$ are detected in 3 iterations, independent of the level number $J$ (disregarding that one less iteration is needed for $J = 4$, i.e. the lowest resolution of

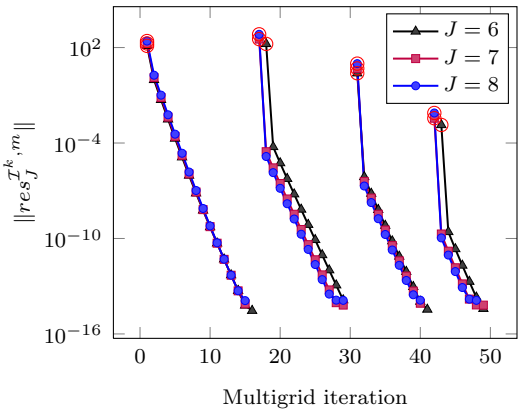



Figure 5.3: Reduction of the norm of the PDAS-residual $res_J^{\mathcal{I}^k,m}$ in each PDAS iteration $k$. The mark ○ indicates the respective initial residual $res_J^{\mathcal{I}^k,0}$.

Table 5.1: Average convergence rate of the $V_{1,1}$-multigrid cycle for the solution of (5.8) at each step $k$ of the PDAS iteration.

| $J$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 8 | $8.99_{-2}$ | $1.08_{-1}$ | $1.06_{-1}$ | $1.09_{-1}$ |
| 7 | $8.86_{-2}$ | $1.08_{-1}$ | $1.04_{-1}$ | $9.48_{-2}$ |
| 6 | $8.93_{-2}$ | $1.10_{-1}$ | $1.05_{-1}$ | $9.36_{-2}$ |
| 5 | $9.17_{-2}$ | $1.10_{-1}$ | $1.07_{-1}$ | $9.49_{-2}$ |
| 4 | $9.51_{-2}$ | $1.09_{-1}$ | $1.01_{-1}$ | — |

the fine mesh). The PDAS convergences at a superlinear rate, furthermore, for the final error obtained in iteration $k = 4$ we have $e_J^u \sim \mathcal{O}(h_J^2)$ (this will be confirmed shortly below). In Figure 5.3 the reduction of $\|res_J^{\mathcal{I}^k,m}\|$ is shown for each $k = 1, 2, 3, 4$. The symbol ○ denotes the begin of each PDAS step $k$. The corresponding average convergence rates for different $J$ and in each PDAS iteration are given in Table 5.1. We observe that the convergence of the inner multigrid iteration is independent of the outer iteration and for each system (5.8) corresponds to the rates which have been obtained for the unconstrained model problem in Section 4.5.1. The rather minuscule variations in $\varrho_{\text{avg}}$ are due the changing $\mathcal{I}$, which affects smoothing and more so the coarse grid correction. In Table 5.2 we present the discrete $L^2$-error $e_J^u$ obtained in the final PDAS step. In the second and third column, we give the errors and associated ratios which have been obtained by solving (5.8) with the standard multigrid solver, i.e. iterations with the $V_{1,1}$-cycle have been performed until the stopping criterion applied. Clearly second-order convergence is observed. In the fourth and fifth column we give the same data which has been computed by using just one iteration of the full multigrid for each PDAS step. Within the FMG, the same $V_{1,1}$-cycle has been used.

Table 5.2: Discrete $L^2$-error of the control $u_J^*$ for ($\mathrm{LQP}_h$) with control-constraints (5.23). Shown is the error for different levels $J$ which is obtained in the final step of the PDAS iteration with fully converged $V_{1,1}$-cycles, FMG and FMG plus one additional $V_{1,1}$-cycle.

| $J$ | $V_{1,1}$ | | FMG | | FMG + $V_{1,1}$ | |
|---|---|---|---|---|---|---|
| | $e^4_{u_J}$ | Ratio | $e^4_{u_J}$ | Ratio | $e^4_{u_J}$ | Ratio |
| 3 | $2.9225_{-5}$ | — | $3.1906_{-4}$ | — | $4.8750_{-5}$ | — |
| 4 | $7.3051_{-6}$ | $2.49_{-1}$ | $8.3699_{-5}$ | $2.62_{-1}$ | $1.2529_{-5}$ | $2.57_{-1}$ |
| 5 | $1.8262_{-6}$ | $2.50_{-1}$ | $2.1282_{-5}$ | $2.54_{-1}$ | $3.1626_{-6}$ | $2.52_{-1}$ |
| 6 | $4.5655_{-7}$ | $2.50_{-1}$ | $5.3510_{-6}$ | $2.51_{-1}$ | $7.9325_{-7}$ | $2.50_{-1}$ |
| 7 | $1.1414_{-7}$ | $2.50_{-1}$ | $1.3403_{-6}$ | $2.50_{-1}$ | $1.9854_{-7}$ | $2.50_{-1}$ |
| 8 | $2.8534_{-8}$ | $2.50_{-1}$ | $3.3527_{-7}$ | $2.50_{-1}$ | $4.9656_{-8}$ | $2.50_{-1}$ |

The absolute error is roughly one order of magnitude larger than for the fully converged multigrid solution, but it still reduces at the same rate. In the last two columns, we again present the same data, however this time the FMG has been followed by one additional $V_{1,1}$-cycle. This reduces the error to the same order of magnitude as that of the conventional multigrid solver. These results reflect the situation for scalar elliptic problems. Under the assumption that the convergence rate of the employed multigrid cycle is smaller than 1/6, one FMG iteration yields an approximate solution with an error of $(5/2)ch_J^2$ and one additional multigrid cycle reduces that error below $(1/2)ch_J^2$. Here, $c$ is the constant from the error estimate $\|u_J^* - u^*\| \leq ch^2$. Comparison of the relative performance in terms of wall-clock time will be given below for a different example.

As the second test case let us consider the bilaterally constrained problem with

$$u^\alpha = \begin{cases} -0.75 & \text{for } y \leq 0.5 \\ -0.9 & \text{for } y > 0.5 \end{cases} \quad \text{and } u^\beta = y^3 - 0.5. \tag{5.24}$$

The computed optimal control $u_J^*$ and $\mathcal{A}^* = \mathcal{A}_-^* \cup \mathcal{A}_+^*$ are depicted in Figure 5.4 left and right, respectively, computed on a mesh with $h_J = 2^{-8}$. In Figure 5.5 we show the discrete inactive and active set on levels $J = 0, 1, 2, 3$ as generated by the coarsening process (5.15). In Figure 5.6 the iteration history of the outer PDAS iteration is shown. The inner systems are solved with the FMG followed by an additional multigrid cycle. On the left, we show the error $e_J^{u^k}$, on the right, we show $e_\beta^{u^k}$. The error $e_\alpha^{u^k}$ vanishes already in the second iteration and is not displayed here. Again, superlinear convergence of the outer iteration is clearly visible. The final error obtained for $k = 4$ again is second-order convergent with respect to $h_J$. The corresponding data is given in Table 5.2, where we also list the computing time in seconds. The numbers confirm the optimal complexity $\mathcal{O}(3h_J^{-2})$ of the FMG solver.

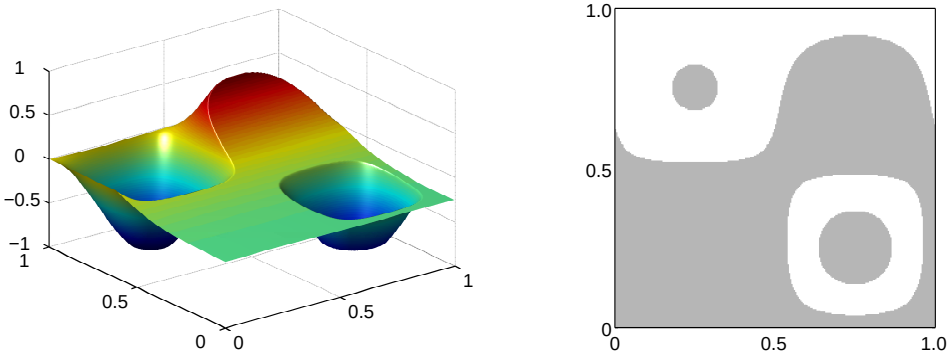



Figure 5.4: Computed optimal control $u_J^*$ (left) and corresponding active set $\mathcal{A}^* = \mathcal{A}_+^* \cup \mathcal{A}_-^*$ (shaded region, right) on a mesh with $h_J = 2^{-8}$ for ($\mathrm{LQP}_h$) with control-constraints (5.24).

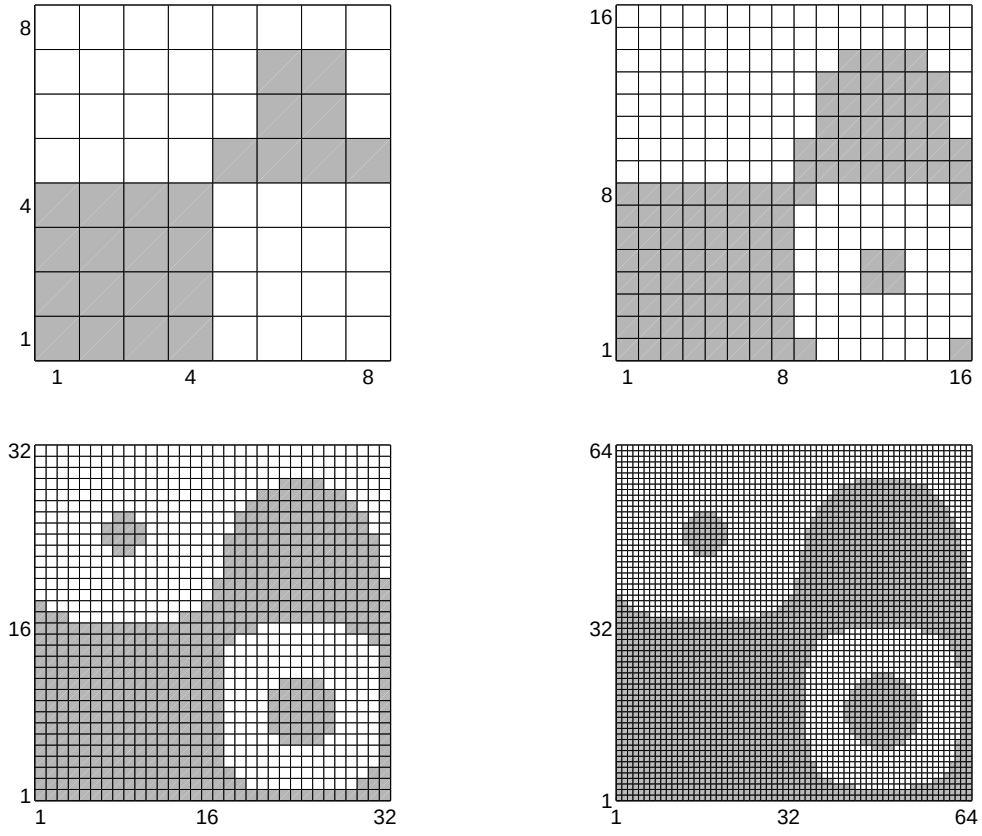



Figure 5.5: Inactive set $\mathcal{I}^*$ and active set $\mathcal{A}_+^*$ (shaded) generated by coarsening (5.15) on levels $J = 0, 1, 2, 3$ for ($\mathrm{LQP}_h$) with control-constraints (5.24).

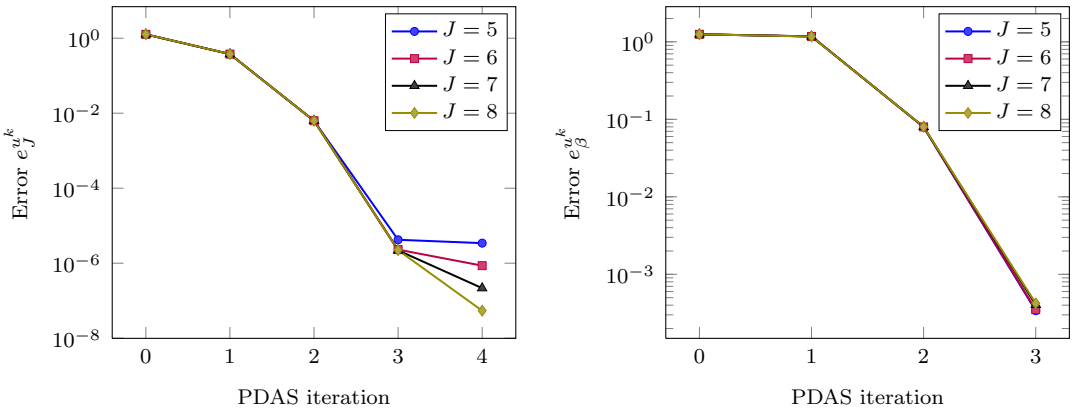



Figure 5.6: PDAS iteration history for ($\mathrm{LQP}_h$) and constraints (5.24). The inner system (5.8) was solved with FMG followed by one $V_{1,1}$-cycle. Shown are the discrete $L^2$-error of the control (left) and the violation of the upper bound (right).

Table 5.3: Discrete $L^2$-error of the control $u_J^*$ for ($\mathrm{LQP}_h$) with control-constraints (5.23) and wall-clock time for the solution with FMG plus one additional $V_{1,1}$-cycle.

| $J$ | $e^4_{u_J}$ | Ratio | time [s] | Ratio |
|---|---|---|---|---|
| 5 | $3.4246_{-6}$ | — | $1.2723_{+1}$ | — |
| 6 | $8.5909_{-7}$ | $2.51_{-1}$ | $5.9187_{+1}$ | 4.65 |
| 7 | $2.1570_{-7}$ | $2.51_{-1}$ | $2.6605_{+2}$ | 4.49 |
| 8 | $5.4015_{-8}$ | $2.50_{-1}$ | $1.1127_{+3}$ | 4.18 |
| 9 | $1.3530_{-8}$ | $2.50_{-1}$ | $4.5630_{+3}$ | 4.10 |

In the context of Newton-like methods, a different strategy is commonly employed to optimize the efficiency with respect to computing time. In inexact Newton methods (cf. Section 6.3.1), the accuracy requirement for the solution of the inner systems is coupled to the progress of the outer iteration. Here, we test how a fixed number of multigrid iterations, e.g. just one or two cycles per outer PDAS step, performs. This is a specific truncated (semismooth) Newton method. Naturally, we expect an increase in the number of PDAS iterations and at least for $m = 1$, we can only expect linear convergence. This is confirmed in Figure 5.7, where the error $e_J^{u^k}$ is plotted against the wall-clock time measured in seconds. Marks indicate a new PDAS step. However, measured in total units of multigrid cycles both truncated approaches are very competitive and the overall performance equals that of the full multigrid approach.

For fixed $\sigma$, the convergence of the outer PDAS iteration does not depend on $h_J$. Let us now consider the dependence on the regularization parameter $\sigma$. To this end, we consider the first test case with the unilateral bound (5.23) and vary $\sigma$ between $1_{-2}$ and $1_{-5}$. Figure 5.8 shows the discrete $L^2$-error $e_J^{u^k}$ on the left, and the bound

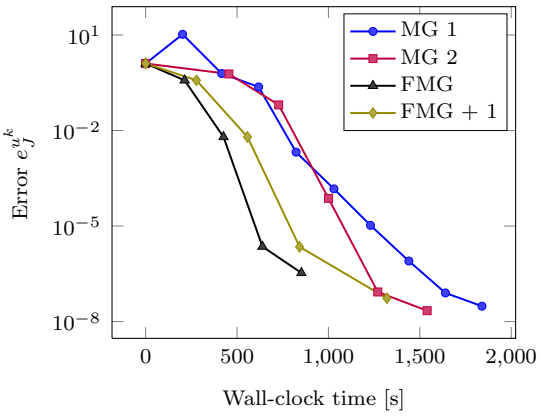



Figure 5.7: Performance comparison of FMG, FMG $+V_{1,1}$, and 1 and 2 $V_{1,1}$-cycles per PDAS iteration, respectively.

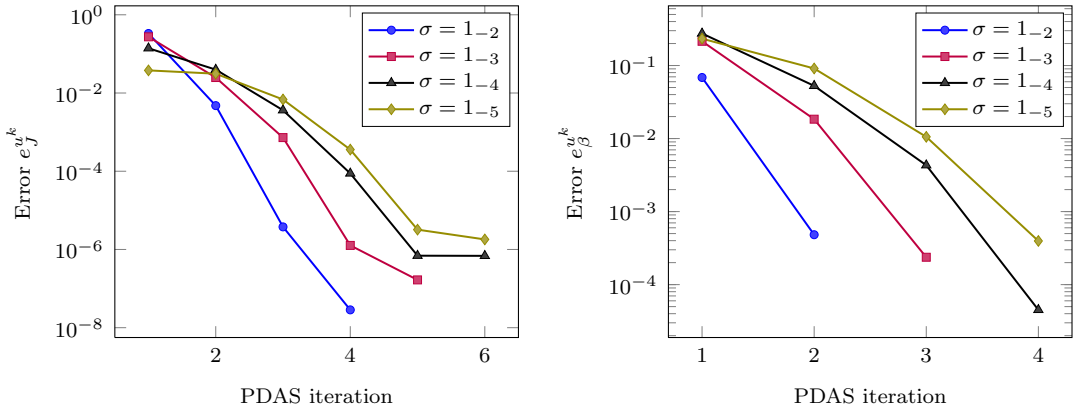



Figure 5.8: Convergence of the outer PDAS iterations for different values of the regularization parameter $\sigma$. $L^2$-error $e_J^{u^k}$ (left) and bound violation $e_\beta^{u^k}$ (right).

violation measured with $e_\beta^{u^k}$ on the right. We observe a slight increase in the number of iterations for decreasing $\sigma$. This is in accordance with results reported in [24] and is therefore not related to method used to solve (EQP). The given heuristic explanation is that for smaller $\sigma$, the constraints can act stronger and $\mathcal{A}^*$ is larger for smaller $\sigma$. Depending on the initial guess, more iterations are required until $\mathcal{A}^*$ is fully resolved. The size of the active set and the growth in each PDAS iteration is reported in Table 5.4 for $\sigma = 1_{-2}, 1_{-5}$ and $J = 6, 7, 8$. The $k$-th column contains the data for the $k$-th PDAS iteration. In the first column, we give the size of $\mathcal{A}^1$, each following column contains the growth, i.e. the increase in cell numbers, from $\mathcal{A}_+^k$ to $\mathcal{A}_+^{k+1}$. Furthermore, we report the error $e_\beta^{u^k}$.

Table 5.4: Size of the active set $\mathcal{A}_+^k$ and error $e_\beta^{u^k}$ for $J = 6, 7, 8$ and regularization parameters $\sigma = 1.0\text{e-}2$ and $\sigma = 1.0\text{e-}5$.

| $\sigma$ | $J$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| | 6 | 124952 | +6096 | +24 | — | — |
| | | $6.843_{-2}$ | $4.149_{-4}$ | 0.0 | — | — |
| $1_{-2}$ | 7 | 499670 | +24474 | +144 | — | — |
| | | $6.848_{-2}$ | $4.800_{-4}$ | 0.0 | — | — |
| | 8 | 1998192 | +98254 | +706 | — | — |
| | | $6.856_{-2}$ | $4.820_{-4}$ | 0.0 | — | — |
| | 6 | 101450 | +20528 | +7744 | +1330 | +20 |
| | | $2.338_{-1}$ | $9.117_{-2}$ | $8.090_{-3}$ | $3.379_{-4}$ | 0.0 |
| $1_{-5}$ | 7 | 405590 | +82330 | +30916 | +5342 | +110 |
| | | $2.326_{-1}$ | $8.836_{-2}$ | $9.050_{-3}$ | $3.848_{-4}$ | 0.0 |
| | 8 | 1622360 | +329360 | +123534 | +21352 | +546 |
| | | $2.331_{-1}$ | $9.109_{-2}$ | $1.055_{-2}$ | $3.960_{-4}$ | 0.0 |

Table 5.5: Number of cells in the final inactive set $\mathcal{I}^*$ (fraction of total cell number in parentheses) for different values of $\sigma$.

| $\sigma$ | 1.0 | $1_{-1}$ | $1_{-2}$ | $1_{-3}$ | $1_{-4}$ |
|---|---|---|---|---|---|
| $\lvert\mathcal{I}^*\rvert$ | 100288 ($3.83_{-1}$) | 15616 ($5.96_{-2}$) | 1152 ($4.4_{-3}$) | 64 ($2.44_{-4}$) | 0 |

## 5.5.2 Example: A Bang-Bang Control Problem

This test case is an example for a so-called *bang-bang control*. Such controls are almost everywhere equal to the bounding functions $u^\alpha, u^\beta$. We prescribe the target state

$$\bar{y} = 128\pi^2 \sin(4\pi x_1) \sin(4\pi x_2), \tag{5.25}$$

the lower and upper bounds are given by

$$u^\alpha = -1, \text{ and } u^\beta = 1, \tag{5.26}$$

respectively. In Figure 5.9 we show the computed optimal state $y_h^*$ for $\sigma = 1_{-4}$ on a mesh with $h = 2^{-9}$. Figure 5.10 shows the computed optimal controls $u_h^*$ on the same mesh for a decreasing sequence of values for $\sigma$. For $\sigma = 1_{-4}$, $u_h^*$ everywhere attains the values of the bounds $u_h^\alpha, u_h^\beta$. Correspondingly, the size of the inactive set, $|\mathcal{I}^*|$, is zero and the bound constraints are active in every cell. In Table 5.5 we list the number of cells and the fraction of the total number of cells (in parentheses) in $|\mathcal{I}^*|$ for the different values of $\sigma$. The given values confirm that the size of $\mathcal{I}^*$ shrinks with decreasing $\sigma$. The active set for this example was always detected in two PDAS iterations.

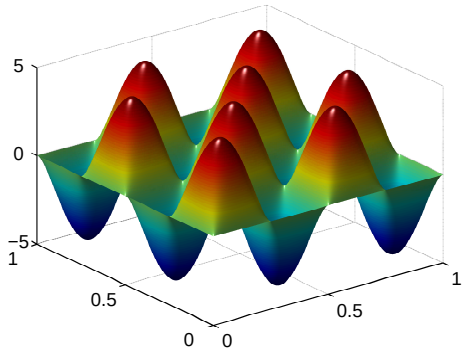
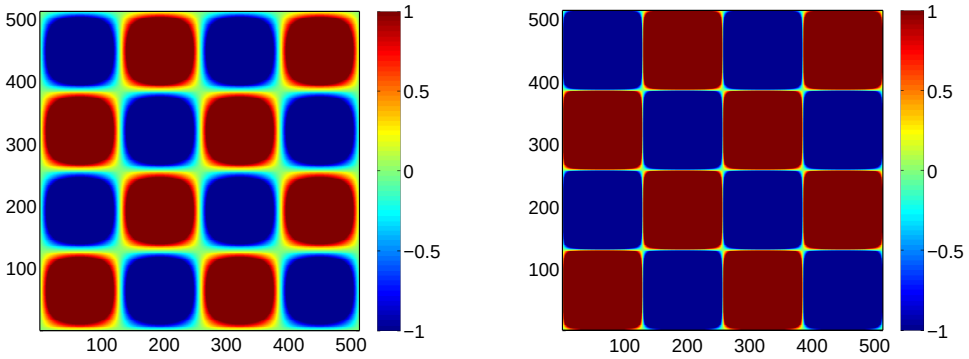
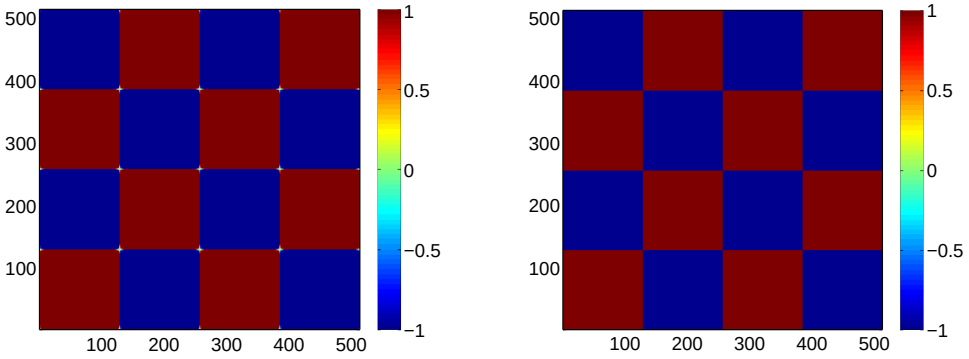



Figure 5.9: Computed optimal state $y_h^*$ for the example problem with an optimal control $u_h^*$ of bang-bang type and $\sigma = 1_{-4}$ on the finest mesh with $h_J = 2^{-9}$.




(a) $\sigma = 1.0$ (left) and $\sigma = 1_{-1}$ (right).




(b) $\sigma = 1_{-2}$ (left) and $\sigma = 1_{-4}$ (right).

Figure 5.10: Computed optimal controls $u_h^*$ for the finest grid with a mesh size $h_J = 2^{-9}$ and different regularization parameters $\sigma$.

## Summary

We considered the numerical solution of linear-quadratic PDE constrained optimization problems with additional pointwise inequality constraints imposed on the control function. The control constraints are treated in an outer iteration, which here is given by an implementation of the primal-dual active-set strategy (PDAS). The PDAS method generates a sequence of equality constrained problems. In order to solve these problems efficiently, we extended the multigrid approach devised for equality-constrained problems (cf. Chapter 4). The required modifications have been described in detail. We also adapted the full multigrid method in order to solve the PDAS subproblems with optimal complexity.

Several numerical examples have been discussed, including constant and non constant upper and lower bounds. It has been demonstrated that the PDAS-multigrid method yields an efficient solver for discrete optimality systems with pointwise inequality constraints on the control. For both the outer PDAS iteration and the inner multigrid method the convergence does not depend on the mesh size $h_J$ of the finest discretization level. It also has been demonstrated that the solution of the subproblems is achieved with optimal complexity, employing the full multigrid method. Overall, the resulting PDAS-multigrid method is a fast solver for linear-quadratic programming problems. A natural application for such methods is as solver for the quadratic subproblems which arise within a sequential quadratic programming (SQP) algorithm. This will be the topic of the next chapter.

# 6 A SQP Multigrid Method for Semilinear PDE Constrained Optimization

In this chapter we consider the numerical solution of optimization problems in the form

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}(y,u) \\ \text{subject to} \quad & \mathcal{C}(y,u) = 0 \\ \text{and} \quad & u \in \mathcal{U}_{\text{ad}}. \end{aligned} \tag{NP}$$

In particular, we consider problems (NP) where $\mathcal{J} : Y \times U \to \mathbb{R}$ is given by the tracking-type functional (2.1), the closed and convex set $\mathcal{U}_{\text{ad}} \subset U$ is defined by the box-constraints (5.1) and the equality constraints $\mathcal{C}(y,u) : Y \times U \to W$ represent a semilinear elliptic boundary value problem

$$-\Delta y + g(x,y) = u \text{ in } \Omega, \quad y = 0 \text{ on } \partial\Omega. \tag{6.1}$$

We note that throughout the Laplacian in (6.1) could be replaced by the the general second-order operator (2.20). The term $g(x,y)$, which defines a nonlinear function in the space variable $x$ and the state $y$, is called a superposition or Nemytzkij-operator [5]. To unburden the notation, we will omit the space variable $x$ when referring to $g$. Due to the nonlinearity, the problem (NP) is not convex and may admit multiple critical points and solutions. Necessary first-order and sufficient second-order conditions characterizing a local minimizer of (NP) will be given in Section 6.1.

The discretization of (NP) yields a large-scale nonlinear programming problem (NLP). Our numerical approach for the solution of this NLP is based on the sequential quadratic programming (SQP) method which is an iterative method that requires the solution of a quadratic programming problem (QP) at each iteration. The QPs are essentially obtained from a quadratic approximation of the Lagrangian and a linear approximation of the constraints. For the unconstrained problem $\mathcal{U}_{\text{ad}} = L^2(\Omega)$, the discretized QP is a KKT system of the form (2.51). In this case, the SQP approach is equivalent to the Lagrange-Newton method in the sense that the solutions of the QPs generate the same iterates as those which are obtained by a step of Newton's method applied to the nonlinear first-order conditions. Thus, under suitable conditions, the SQP method inherits the fast, i.e. superlinear or quadratic, local convergence of Newton's method. Furthermore, Newton's method often satisfies a mesh-independence principle [3] which carries over to the SQP approach as well. Thus, when combined

with a mesh-independent inner iteration for the solution of the QPs, such as our multigrid method from Chapter 4, the overall approach should result in an efficient and mesh-independent solution method.

In the control-constrained case with $\mathcal{U}_{\text{ad}}$ given by (5.1), we extend the above algorithm by adding the control-constraints to the QP subproblems. In this case, each subproblem is a quadratic complementarity problem corresponding to (CC) and can be solved by the PDAS-multigrid method as discussed in Chapter 5. Again, mesh-independence of inner and outer iteration are expected to yield an efficient solution method for the discretization of (NP).

Several further issues, such as globalization by line search or trust-region strategies, inexactness, different inner solvers, and differences in the specific formulation of the QP respective the Newton updates give rise to a number of variants of the SQP approach. Often the Lagrange-Newton method is denoted the SQP method, in particular in the PDE-constrained optimization context [66, 67, 89], whereas in the general nonlinear optimization context, the terminology SQP always is understood to include inequality constraints. For an interpretation as a semi-smooth Newton method in function space see [95] in the context of optimal control of the Navier-Stokes equations, and [146] for an application to control of Burgers' equation. For a survey of SQP methods we refer to [32].

## 6.1 Existence and Characterization of Solutions

In this section we discuss the necessary first-order and sufficient second-order conditions to characterize a solution of (NP). Different approaches exist in the literature to prove existence of a solution to (NP), we refer to [6, 112, 145] for optimal control problems, to [115, 157] for Lagrange multiplier theory and optimization in vector spaces and to [127] for the finite dimensional case. Here we state the relevant results.

Due to the nonlinearity of $\mathcal{C}$, the problem (NP) is not convex and uniqueness of a minimizer can not be expected. A point $(y, u)$ is called feasible or admissible, if $u \in \mathcal{U}_{\text{ad}}$ and $\mathcal{C}(y, u) = 0$. A feasible point $(y^*, u^*) \in Y \times \mathcal{U}_{\text{ad}}$ is a locally optimal solution of (NP), if there exists a neighborhood $\mathcal{N}_\varepsilon(y^*, u^*) \subset Y \times \mathcal{U}_{\text{ad}}$ such that

$$\mathcal{J}(y^*, u^*) \leq \mathcal{J}(y, u) \text{ for } (y, u) \in \mathcal{N}_\varepsilon(y^*, u^*). \tag{6.2}$$

If strict inequality in (6.2) holds, $(y^*, u^*)$ is called a strict locally optimal solution.

**Definition 6.1.** Let $(y^*, u^*)$ be a local solution of (NP). Then $p^* \in W'$ is denoted the associated Lagrange multiplier, provided

$$D_{(y,u)}\mathcal{L}(y^*, u^*, p^*)(y - y^*, u - u^*) \geq 0 \text{ for } (y, u) \in Y \times \mathcal{U}_{\text{ad}} \tag{6.3}$$

holds.

Here, $D_{(y,u)}\mathcal{L}$ denotes the first Fréchet-derivative of the Lagrangian (2.22) with respect to $y$ and $u$, which is assumed to exist. We remark that the usual complementarity condition does not apply here since the second equation in (NP) is an equality constraint. The existence of such an optimal multiplier $p^*$ requires a constraint qualification to hold. In the special case considered here the constraint qualification reads

$$D_{(y,u)}\mathcal{C}(y^*,u^*)C(y^*,u^*) = W, \tag{6.4}$$

where $C(y^*,u^*) = \{\lambda(y-y^*,u-u^*) \mid \lambda \geq 0, (y,u) \in Y \times \mathcal{U}_{\mathrm{ad}}\}$ denotes the conical hull of $Y \times \mathcal{U}_{\mathrm{ad}}$ in $(y^*,u^*)$. Then existence of $p^*$ follows from

**Theorem 6.2.** *Let $(y^*,u^*)$ denote a local minimizer of* (NP)*, let $\mathcal{J}$ and $\mathcal{C}$ be continuously Fréchet-differentiable in $\mathcal{N}_\varepsilon(y^*,u^*)$. If the constraint qualification* (6.4) *holds, then there exists an optimal multiplier $p^* \in W'$.*

Using the notation of Section 2.2, the first-order conditions (6.3) together with the constraints $\mathcal{C}(y^*,u^*) = 0$ yield the nonlinear optimality system

$$\begin{aligned} Ay^* + Bg(y^*) - Bu^* &= 0, \\ A'p^* + B'\big(D_y g(y^*)\big)'p^* &= E'(\bar{y} - Ey^*), \\ (\sigma u^* - B'p^*, u-u^*) &\geq 0, \quad u \in \mathcal{U}_{\mathrm{ad}}. \end{aligned} \tag{NOS}$$

consisting of the state equation, the adjoint equation and the optimality condition, cf. (OS). The variational inequality turns out to be the same as in the linear case since $\mathcal{C}(y,u)$ is linear with respect to $u$. In order to prove that a point $(y^*,u^*)$ which satisfies (6.3) is actually a minimizer, we need a second-order condition. To this end, we require that $\mathcal{J}$ and $\mathcal{C}$ are two times continuously Fréchet-differentiable.

**Theorem 6.3.** *If a critical point $(y^*,u^*)$ and its associated multiplier $p^*$ satisfy*

$$D^2_{(y,u)}\mathcal{L}(y^*,u^*,p^*)(y,u)(y,u) \geq \alpha(\|y\|_Y^2 + \|u\|_U^2), \quad (y,u) \in \ker D_{(y,u)}\mathcal{C}(y^*,u^*) \tag{6.5}$$

*for some $\alpha > 0$, then $(y^*,u^*)$ is a locally optimal solution of* (NP).

For the concrete problem at hand, the second Fréchet-derivative of the Lagrangian is

$$D^2_{(y,u)}\mathcal{L}(y^*,u^*,p^*)(y_1,u_1)(y_2,u_2) = (y_1,y_2) + \sigma(u_1,u_2) + \langle p^*, BD_y^2 g(y^*)y_1y_2\rangle, \tag{6.6}$$

and thus (6.5) reads

$$\|y\|^2_{L^2(\Omega)} + \sigma\|u\|^2_{L^2(\Omega)} + \big(p^*, BD_y^2 g(y^*)y^2\big) \geq \alpha\big(\|y\|^2_{H_0^1(\Omega)} + \|u\|^2_{L^2(\Omega)}\big) \tag{6.7}$$

for all $(y,u)$ satisfying the adjoint equation.

In order to justify above computations and results for concrete applications, we need to verify the corresponding differentiability properties in $Y = H_0^1(\Omega), U = L^2(\Omega)$ and $W = Y'$. Furthermore, due to reflexivity, $Y''$ is identified with $Y$, i.e. $p^* \in Y$. The tracking-type functional (2.1) satisfies the differentiability requirements. We refer to [6] for suitable conditions in the case of a more general $\mathcal{J}$ . The required Fréchet-differentiability of $\mathcal{C}$ implies the corresponding differentiability of the superposition operator $g(y)$ in $Y$. Appropriate growth conditions of the form

$$|g(y)| \leq \alpha + \beta |y|^{(p/q)}, \ \alpha \in L^q(\Omega), \ \beta \in L^\infty(\Omega), \ 1 \leq q \leq p < \infty, \tag{6.8}$$

ensure continuity and differentiability of $g(y)$ for appropriate indices $p, q$. Using Sobolev embeddings, one obtains that in the setting of Chapter 2, i.e. $Y = H_0^1(\Omega)$, differentiability is obtained for example for $g(y) = y^k, k \leq 5$, cf. [5, 145]. Stronger nonlinearities such as $g(y) = \exp(y)$ require additional Lipschitz and boundedness conditions as stated in

**Assumption 6.4.** Let $g : \Omega \times \mathbb{R} \to \mathbb{R}$ be a Carathéodory function of class $C^2$. Furthermore, we assume that for all $M > 0$ there exists a constant $C_M > 0$ such that

$$|D_y g(x, y)| + |D_{yy} g(x, y)| \leq C_M \qquad \text{for a.e. } x \in \Omega \text{ and } |y| \leq M \tag{6.9}$$

$$|D_{yy} g(x, y_1) - D_{yy} g(x, y_2)| \leq C_M |y_1 - y_2| \qquad \text{for } x \in \Omega \text{ and } |y_1|, |y_2| \leq M. \tag{6.10}$$

and

$$D_y g(x, y) \geq 0 \tag{6.11}$$

In this case, one applies formally the Lagrange principle and has to prove that the Lagrange multiplier is obtained as a solution of the adjoint equation (cf. also Section 2.2). The condition (6.11) in particular ensures that 0 is not an eigenvalue of $D_{(y,u)}\mathcal{C}(y^*, u^*)$, thus allowing to establish surjectivity and existence of optimal multipliers. At this point we remark that not all numerical examples considered below satisfy (6.9)–(6.11). For further details we refer to [6], where it is shown that under above assumptions, the state equation has a unique solution $y \in H_0^1(\Omega) \cap W^{2,p}(\Omega)$ for any $1 \leq p < \infty$ and a Lagrange multiplier is obtained as the unique weak solution of the adjoint equation, satisfying (6.3).

Verifying the second-order conditions a priori, e.g. based on (6.7), in general is not possible, since it requires knowledge of the exact solution. Still, second-order sufficient conditions play an essential role as they are needed to prove error estimates [6] as well as local convergence of Lagrange-Newton or SQP methods formulated in function space [95, 96]. We refer the interested reader to [135], where it is proposed to check a certain discretized second-order condition numerically. Frequently, the second-order conditions require a certain controllability of the state, i.e. a smallness assumption of the tracking error $\|y^* - \bar{y}\| \leq \varepsilon$, [95].

## 6.2 The Discrete Optimality System

Discretizing (NP) yields a nonlinear programming problem NLP

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}_h(y_h, u_h) \\ \text{subject to} \quad & \mathcal{C}_h(y_h, u_h) = 0 \\ \text{and} \quad & u_h \in \mathcal{U}_{\text{ad},h}. \end{aligned} \tag{NLP}$$

The necessary first-order conditions of (NLP), which correspond to (NOS), are given by the Karush-Kuhn-Tucker conditions. Hence, the discrete optimality system can be derived in a similar fashion as in the linear case, cf. Section 2.3.3. We consider first the case $\mathcal{U}_{\text{ad}} = U$, the box-constraints (5.1) will be incorporated by the SQP method as defined in the next section. Discretizing the state equation yields the nonlinear equation

$$\mathcal{C}_h(y_h, u_h) = L_h y_h + M_h g_h(y_h) - M_h u_h - M_h f_h = 0, \tag{6.12}$$

which differs from (2.35) by the additional nonlinear term $M_h g_h(y_h)$. The discrete Lagrangian is given by

$$\mathcal{L}_h(y_h, u_h, p_h) = \mathcal{J}_h(y_h, u_h) + p_h^T \mathcal{C}_h(y_h, u_h), \tag{6.13}$$

and following the discretize-then-optimize approach, we obtain the adjoint equation

$$D_{y_h}\mathcal{L}_h(y_h, u_h, p_h) = M_h y_h + \big(L_h^T + M_h g_{y,h}(y_h)\big)p_h - M_h \bar{y}_h, \tag{6.14}$$

where $L_h^T + M_h g_{y,h}(y_h)$ is the discrete version of the adjoint of the linearized state operator and $g_{y,h}$ denotes the derivative of $g_h$ with respect to $y$. The discrete optimality condition follows as

$$D_{u_h}\mathcal{L}_h(y_h, u_h, p_h) = \sigma M_h u_h - M_h p_h. \tag{6.15}$$

Noting that (6.12) corresponds to $D_{p_h}\mathcal{L}_h(y_h, u_h, p_h)$ and ordering the equations according to (6.14), (6.15), (6.12), we obtain the nonlinear system

$$\begin{aligned} D_{y_h}\mathcal{L}_h(y_h, u_h, p_h) &= 0 \\ D_{u_h}\mathcal{L}_h(y_h, u_h, p_h) &= 0 \\ D_{p_h}\mathcal{L}_h(y_h, u_h, p_h) &= 0, \end{aligned} \tag{NOS$_h$}$$

which is the discrete version of (NOS). Discretizations of (NP) have been considered for example in [6, 118]. In [6] it is shown that under Assumptions 6.4 and if (6.7) holds, a piecewise constant discrete control $u_h$ exists which solves (NLP), and furthermore, the optimal $L^2$ order of convergence

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch \tag{6.16}$$

holds.

## 6.3 Lagrange-Newton Methods and Sequential Quadratic Programming

As we have outlined in the introductory section, for the solution of ($\text{NOS}_h$) we devise a Newton-based algorithm, which employs the multigrid method of Chapter 4 for the solution of the arising linear Newton systems. A prominent alternative to treat nonlinear problems with multigrid is the Full Approximation Storage (FAS) approach, proposed in [40] for the solution of nonlinear elliptic PDEs. The FAS multigrid defers the linearization process to the smoothing iteration. Since in the nonlinear case the equivalence of the residual equation $Ae = r$ to the original problem $Ax = b$ does not, the FAS multigrid computes solution updates instead of error corrections. The FAS approach has been applied to optimal control problems in [33, 35]. There, a collective Gauss-Seidel smoother (cf. Section 4.2) is applied to the optimal control of the Bratu problem. To our knowledge, this is the only approach which treats nonlinearly constrained optimal control problems with a multigrid-related full-space approach.

Both FAS and Newton-multigrid approaches can be extremely efficient solvers for nonlinear equations, and a final answer to the question, which approach is more efficient or robust, can not be given. Further details can be found in the standard multigrid references, e.g. [85, 147].

In the following, let $F_h(w_h)$ with $w_h = (y_h, u_h, p_h)$ denote the nonlinear operator defined by ($\text{NOS}_h$). Solving ($\text{NOS}_h$) then amounts to finding the root of

$$F_h(w_h) = 0, \tag{6.17}$$

which is done by Newton's method. As Newton's method is only locally convergent, some measure for globalization is required to increase robustness when converging from remote starting points. To this end, two strategies are commonly employed: line search and trust-region methods. In the context of a full SQP method, applying a line-search method to the primal-dual iterate seems more natural than a trust-region strategy.[1] For an introduction to both variants we refer to [127]. The basic Newton-line search algorithm reads

*i*) compute the Newton step $\delta w_h^l$ by solving

$$DF_h(w_h^l)\delta w_h^l = -F_h(w_h^l), \tag{6.18}$$

*ii*) use a line search algorithm to compute a step length $\alpha_l$,

[1] Trust-region strategies are based on positive definite modifications of the Hessian. In the full space one has to deal with the indefinite KKT matrix, therefore trust-region methods are mostly applied to reduced space approaches.

*iii*) perform the Newton update

$$w_h^{l+1} = w_h^l + \alpha_l \delta w_h^l. \tag{6.19}$$

The line search algorithm in step *ii)* intentionally is left unspecified, since some details specific for constrained optimization problems will have to be considered, which we will do below. The original, undamped Newton method results if only unity steps are taken. As we will see, in order to obtain the full Newton efficiency it is important that the line search strategy eventually admits $\alpha_l = 1$ when the iterates approach the solution. Still, exact solves of (6.18) are not required for all $l$, in particular in the initial phase the accuracy can be considerably relaxed, leading to a further gain of efficiency. These details will be discussed in Section 6.3.1 and 6.3.2.

The gradient of the nonlinear function $F_h(w_h)$ is given by the Hessian of the Lagrangian,

$$DF_h(w_h) = D^2\mathcal{L}_h(x_h, p_h). \tag{6.20}$$

In iteration $l$, the system (6.18) is defined by a linear operator of KKT form

$$DF_h(w_h^l) = \begin{pmatrix} D^2_{yy}\mathcal{L}_h(w_h^l) & D^2_{yu}\mathcal{L}_h(w_h^l) & (D_y\mathcal{C}_h(x_h^l))^T \\ D^2_{uy}\mathcal{L}_h(w_h^l) & D^2_{uu}\mathcal{L}_h(w_h^l) & (D_u\mathcal{C}_h(x_h^l)^T \\ D_y\mathcal{C}_h(x_h^l) & D_u\mathcal{C}_h(x_h^l) & 0 \end{pmatrix} \tag{6.21}$$

and the right hand side is given by the nonlinear residual

$$-F_h(w_h^l) = \begin{pmatrix} D_y\mathcal{L}_h(w_h^l) \\ D_u\mathcal{L}_h(w_h^l) \\ \mathcal{C}_h(x_h^l) \end{pmatrix}. \tag{6.22}$$

For problem (NP) with (6.1) we obtain (cf. Section 6.2)

$$DF_h(w_h^l) = \begin{pmatrix} M_h + g_{yy,h}(y_h^l)p_h^l & 0 & L_h^T + g_{y,h}(y_h^l) \\ 0 & \sigma M_h & -M_h \\ L_h + g_{y,h}(y_h^l) & -M_h & 0 \end{pmatrix}, \tag{6.23}$$

and (6.22) given by ($\mathrm{NOS}_h$). If Assumptions 6.4 and the second-order sufficient condition (6.7) are satisfied, the operator $DF_h(w_h^l)$ will be uniformly bounded in a neighborhood of a minimizer. The Newton step (6.18) with (6.23) and (6.22) is then well defined and local convergence of Newton's method follows [129].

## 6.3.1 Inexact Newton Methods

In practice, the Newton system (6.18) can be solved only iteratively to a certain accuracy. This leads to so-called truncated Newton methods. Related to this are inexact Newton methods. Here, the accuracy of the Newton step $\delta w_h^l$ is deliberately reduced

in order to improve the efficiency. The underlying idea is that, far from a solution, the quality of the Newton step might be disputable, and thus solving with high accuracy is commonly referred to as "oversolving", a potential waste of computational resources. In fact, far from the solution an accurately computed $\delta w_h^l$ might even yield less progress towards the solution than a less accurate solution of (6.18). Close to the solution however it is important to compute accurate steps $\delta w_h^l$, such that the line search method can accept unity step lengths and the fast local convergence of Newton's method is preserved. The precise relationship between the outer iteration and the inner linear process has been analyzed in [55]. We define the inner, linear residual for the outer Newton step $l$ as

$$r_l^m = DF_h(w_h^l)\delta w_h^{l,m} + F_h(w_h^l), \tag{6.24}$$

where $m$ is the iteration index of the inner linear process employed for the solution of (6.18). The outer, nonlinear residual is given by $F_h(w_h^l)$. Inexact Newton methods arise, if the Newton system (6.18) is solved such that

$$\|r_l^m\| \leq \eta_l \|F_h(w_h^l)\| \tag{6.25}$$

holds with a nonnegative *forcing sequence* $\{\eta_l\}$. Choosing $\eta_l = 0$ yields Newton's method. In [55] it is proved[2] that the iterates generated by Newton's method which satisfy (6.25)

- converge to $w_h^*$ provided $\eta_l \leq \eta < 1$,
- converge to $w_h^*$ with a superlinear rate provided $\eta_l \to 0$,
- and converge to $w_h^*$ with a quadratic rate if $\eta_l = \mathcal{O}(\|F_h(w_h^l)\|)$.

Practical choices for $\eta_l$ are discussed in [60], in our implementation we use

$$\eta_l = \gamma\Big(\frac{\|F_h(w_h^l)\|}{\|F_h(w_h^{l-1})\|}\Big)^{\alpha}, \tag{6.26}$$

with the common choices $\alpha = 2$ and $\gamma = 1.0$ or $0.9$. Furthermore, we apply the safeguard

$$\eta_l \leftarrow \min\big(\eta_l, \eta_{\max}\big) \tag{6.27}$$

with $\eta_{\max} = 0.5$.

[2] Assuming the usual conditions ensuring local convergence of Newton's method, i.e. nonsingularity and Lipschitz continuity of $DF_h(w_h^*)$ and a starting point sufficiently close to $w_h^*$

### 6.3.2 Globalization and Merit Functions

It remains to define the line search procedure required in step *ii)* of Newton's method. To this end, a scalar function $\phi(w_h)$ which measures the progress of the algorithm is required. In unconstrained optimization, the obvious choice is the objective functional itself. Suitable choices for constrained minimization will be given below. Assuming that the computed step $\delta w_h^l$ is not degenerate, i.e. it is a descent direction of sufficient length, a line search method determines a step length $\alpha_l$ such that a sufficient decrease condition

$$\phi(w_h^l + \alpha_l \delta w_h^l) \leq \phi(w_h^l) + \alpha_l c_1 \nabla\phi(w_h^l)^T \delta w_h^l \tag{6.28}$$

is satisfied. The constant $c_1 \in (0,1)$ is typically chosen as $c_1 = 1_{-4}$ and (6.28) is known as the Armijo-condition. In practical implementations, $\alpha_l$ is evaluated by backtracking, in our implementation we chose the largest $\alpha \in (\alpha_{\min}, 1]$ such that (6.28) is satisfied, always testing the full Newton step $\alpha_l = 1$ first, and reducing $\alpha_l$ by a factor of 0.5 if the trial step fails (6.28).

The choice of merit function corresponding to unconstrained minimization would be $\phi(w_h) = \frac{1}{2}\|F_h(w_h)\|_2^2$. However, in constrained optimization, $\phi(w_h)$ needs to balance the objectives of decreasing $\mathcal{J}_h(y_h, u_h)$ and satisfying $\mathcal{C}_h(y_h, u_h) = 0$. Two common choices for $\phi(w_h)$ are the $l_1$-penalty function

$$\phi(w_h) = \mathcal{J}_h(y_h, w_h) + \rho\|\mathcal{C}_h(y_h, u_h)\|_1 \tag{6.29}$$

and the augmented Lagrangian merit function

$$\phi(w_h) = \mathcal{L}_{h,\varrho}(y_h, u_h, p_h) = \mathcal{L}_h(y_h, u_h, p_h) + \rho\|\mathcal{C}_c\|_2^2, \tag{6.30}$$

where $\mathcal{L}_h(y_h, u_h, p_h)$ was defined in (6.13). In both cases, $\rho \geq 0$ is a penalty parameter to be determined in the course of the iteration. Both merit functions are exact in the sense that any minimizer of (NLP) is also a local minimizer of $\phi(w_h)$, provided the penalty parameter $\rho$ is sufficiently large. The $l_1$-merit function can be evaluated comparatively cheap, but has the distinct drawback that sometimes unit step lengths close to a solution are rejected, preventing the fast local convergence of Newton's method. The augmented Lagrangian merit function does not have this effect, however it requires accurate estimates of the Lagrange multipliers and is more costly to evaluate. In many publications, in particular those concerned with reduced methods, the $l_1$ function is preferred and second-order corrections are applied to overcome the slow-down of convergence speed. Here, since we pursue the full space approach, we compute at every iteration estimates for the Lagrange multipliers. Hence, we opt for (6.30) as merit function. In particular, we treat $p_h$ as independent variable, and the computed step length explicitly applies to $p_h$ as well.

For the Armijo-test (6.28) we need to evaluate the directional derivative

$$\begin{aligned}\nabla\phi(w_h^l)^T \delta w_h^l &= F_h(w_h^l)^T \delta w_h^l + \rho\mathcal{C}_h(y_h, u_h)^T D\mathcal{C}_h(y_h, u_h)\delta x_h^l = F_h(w_h^l)^T \delta w_h^l \\ &+ \rho\big(L_h y_h^l + M_h g_h(y_h^l) - M_h u_h^l\big)^T \big((L_h^T + M_h g_{y,h}(y_h^l))\delta y_h^l - M_h \delta u_h^l\big)\end{aligned} \tag{6.31}$$

The penalty parameter $\rho$ is determined such that $\nabla\phi(w_h^l)^T\delta w_h^l + \bar{c} = 0$ for $\bar{c} > 0$, i.e. $\delta w_h^l$ is a descent direction for $\phi(w_h)$. This yields

$$\rho = \frac{F_h(w_h^l)^T\delta w_h^l + \bar{c}}{-\mathcal{C}_h(y_h, u_h)^T D\mathcal{C}_h(y_h, u_h)\delta x_h^l} \tag{6.32}$$

Note that for $C_h(y_h, u_h) = 0$, any choice of $\rho$ satisfies the descent property since then $\phi(w_h) = \mathcal{J}_h(x_h)$.

The Lagrange-Newton or SQP approach outlined is sometimes called a full-space SQP method, in contrast to the widely-used reduced space methods. These result from applying one of the Schur complement reduction techniques, in particular the null space method, discussed in Chapter 3 for solving (6.18). Reduced SQP methods have been used e.g. in [71, 95, 96, 146]. They are of particular interest, if the dimension of the control space is small, or if for certain reasons a quasi-Newton approximation to the reduced Hessian has to be maintained.

Further measures could be taken to improve robustness and performance of Newton's method. This includes heuristics such as the watchdog strategy, which can temporarily accept steps violating (6.28) to allow for hill-climbing. Further techniques include hybridization with a globally convergent first-order descent method: in case the Newton step due to (6.18) is not well-defined, e.g. far from a minimizer where the second-order condition does not hold, one reverts to the gradient step. Also in principle it would be possible to incorporate a Lanczos process in order to verify numerically that the second-order condition at a computed KKT point holds. We have not included such steps in our implementation and refer to [73, 127] for further details.

### 6.3.3 A Full SQP-Multigrid Method

It remains to incorporate the inequality constraint (5.1) in the solution algorithm. To this end, we exploit the Lagrange-Newton SQP equivalence, which states that (assuming the Newton step is well defined) the solution of (6.18) can be obtained equivalently as the solution of the minimization problem

$$\text{minimize} \quad \frac{1}{2}\delta x_h^T D_{xx}^2\mathcal{L}_h(x_h^l, p_h^l)\delta x_h + (D_x\mathcal{L}_h(x_h^l, p_h^l))^T\delta x_h \tag{6.33a}$$

$$\text{subject to} \quad D_x\mathcal{C}_h(x_h^l)\delta x_h = -\mathcal{C}_h(x_h^l). \tag{6.33b}$$

Hence, the system (6.18) can be interpreted as the first-order optimality condition of a linear-quadratic optimization problem, which is convex and thus has a unique solution. We now add the constraints (5.1) to (6.33) and solve (NLP) by solving the sequence of problems (6.33), (5.1).

We proceed analogously to Chapter 5. The optimality system corresponding to (6.33)

is augmented by the complementarity condition (cf. equation (5.4))

$$\begin{aligned} \sigma M_h \delta u_h^l - M_h \delta p_h^l + \lambda^l &= 0, \\ \lambda^l = \max(\lambda^l + c(\delta u_h^l - (u_h^\beta - u_h^l), 0) + \min(\lambda^l + c(\delta u_h^l - (u_h^\alpha - u_h^l), 0), &\quad c > 0. \end{aligned} \tag{6.34}$$

Here, $\lambda^l$ are the Lagrange multipliers associated with the constraints

$$\xi_h^{\alpha,l} \le \delta u_h^l \le \xi_h^{\beta,l}, \quad \xi_h^{\alpha,l} = u_h^\alpha - u_h^l, \quad \xi_h^{\beta,l} = u_h^\beta - u_h^l. \tag{6.35}$$

For the solution of (6.33), (6.34) we apply the PDAS-multigrid method Algorithm 7. For completeness, we state the equality-constrained QP which has to be solved in each iteration of the PDAS algorithm:

$$\begin{aligned} \big(M_h + M_h g_{yy,h}(y_h^l)p_h^l\big)\delta y_h^{l,k} + \big(L_h^T + M_h g_{y,h}(y_h^l)\big)\delta p_h^{l,k} &= -F_{h,1}(w_h^l) \\ \sigma M_h \delta u_h^{l,k} - M_h^T \delta p_h^{l,k} + \lambda^k &= -F_{h,2}(w_h^l) \\ \big(L_h + M_h g_h(y_h^l)\big)\delta y_h^{l,k} - M_h \delta u_h^{l,k} &= -F_{h,3}(w_h^l) \qquad \text{(SQPEP)} \\ \lambda^{l,k} &= 0 \qquad \text{on } \mathcal{T}_{h,\mathcal{I}^{l,k}} \\ \delta u_h^{l,k} &= \xi_h^{\alpha,l} \qquad \text{on } \mathcal{T}_{h,\mathcal{A}_-^{l,k}} \\ \delta u_h^{l,k} &= \xi_h^{\beta,l} \qquad \text{on } \mathcal{T}_{h,\mathcal{A}_+^{l,k}}. \end{aligned}$$

Here, $l$ is the index of the SQP step, $k$ is the iteration index of the PDAS method and $F_{h,i}, i = 1,2,3$ are the component functions of the nonlinear residual $F_h(w_h)$ with respect to $y_h, u_h, p_h$. Note that the sets $\mathcal{I}^{l,k}, \mathcal{A}_-^{l,k}$ and $\mathcal{A}_+^{l,k}$ are defined analogously to (5.6) (step 3 in Algorithm 7), but now with respect to the Newton increment $\delta u_h^l$ and the bounds $\xi_h^{\alpha,l}, \xi_h^{\beta,l}$ of (6.35). The system (SQPEP) is solved with the multigrid method, where again the partitioning with respect to $\mathcal{I}^{l,k}, \mathcal{A}_-^{l,k}, \mathcal{A}_+^{l,k}$ is exploited. We set $\lambda^{l+1,0} = \lambda^{l,\bar{k}}$, where $\bar{k}$ is the index of the final PDAS iteration in step $l$. This initialization for $\lambda^{l,0}$ corresponds to the so-called "warm-start" feature of active set strategies in conjunction with an SQP approach: close to a solution, $\lambda^l$ provides a good estimate of $\mathcal{A}_-^*, \mathcal{A}_+^*$ and $\mathcal{I}^*$ and only a few PDAS iterations are required. Naturally, $\delta w_h^{l,0} = 0$ provides a reasonable initial value for the Newton increments.

Enforcing (6.35) implies that the new iterate $w_h^{l+1}$ is feasible with respect to (5.1) on the current estimate of $\mathcal{A}_-^{l,\bar{k}}, \mathcal{A}_+^{l,\bar{k}}, \mathcal{I}^{l,\bar{k}}$. The line search and the merit function of the unconstrained case can be used here, and the line search is performed with respect to $\delta w_h^{l,\mathcal{I}} = (\delta y_h^l, \delta u_h^{l,\mathcal{I}}, \delta p_h^l)$.

The convergence of the SQP method is stated in

**Theorem 6.5** ([127]). *Let $x_h^* = (y_h^*, u_h^*)$ denote a local solution of* (NLP). *Assume that* (6.33) *in iteration $l$ is well-defined, i.e. the constraint Jacobian $D_x\mathcal{C}_h(x_h^l)$ has full row rank and $D_{xx}^2\mathcal{L}_h(x_h^l, p_h^l)$ is positive definite on $\ker D_x\mathcal{C}_h(x_h^l)$. Assume furthermore*

*that strict complementarity*[3] *holds at the solution and let* $\mathcal{A}^*$ *denote the active set of* $u_h^*$. *Then, if* $(x_h^l, p_h^l)$ *is close enough to* $(x_h^*, p_h^*)$, *there exists a solution of* (6.33), (6.35) *such that* $\mathcal{A}^{l,\bar{k}} = \mathcal{A}^*$.

This completes the description of the SQP-multigrid method, which is summarized in Algorithm 8.

Sequential Quadratic Programming

1: Set initial values $y_h^0, u_h^0, p_h^0, \lambda^0$, set $l = 0$
2: **while** $\|F_h(w_h^l)\| > \varepsilon$ **do**
3: Compute $\xi_h^{\alpha,l}, \xi_h^{\beta,l}$ according to (6.35)
4: Compute $\delta w_h^l, \lambda$ as solution to (6.33), (6.35) with Algorithm 7
5: Compute step length $\alpha_l$ according to (6.28)
6: Compute Newton update $w_h^{l+1} \leftarrow w_h^l + \alpha_l \delta w_h^l$
7: Set $\lambda^{l+1} = \lambda^{l,\bar{k}}$
8: $l \leftarrow l + 1$

Algorithm 8: The SQP algorithm.

## 6.4 Numerical Results

In this section we present numerical results obtained with our implementation of Algorithm 8. In all experiments, the outer iterations were stopped as soon as

$$\|F_h(w_h^l)\|_2 \leq 1_{-12}. \tag{6.36}$$

Here we chose a stringent termination criterion since the primary interest is to assess the performance of the SQP-multigrid, in practical settings, (6.36) might not be reasonable. The stopping criterion for the inner iterations has been chosen as

$$\|r_l^m\|_2 \leq 1_{-12}, \tag{6.37}$$

if "exact" solution of the Newton system (6.18) was desired. For the inexact Newton strategy as given in Section 6.3.1, we stop the inner iterations based on (6.25), (6.26) with $\alpha = 2$, $\gamma = 1.0$. Unless noted otherwise, we set the regularization parameter to $\sigma = 1_{-3}$ and the initial value to $y_J^0 = 1, u_J^0 = \min\big(\max(1, u_j^\alpha), u_J^\beta\big), p_J^0 = 0$. Note that $u_J^0$ is feasible with respect to the inequality constraints. The coarsest mesh size is always $h_0 = 1/8$, i.e. $h_J = 2^{-(J+3)}$. The QP subproblems are solved with the $V_{1,1}$-cycle, employing the smoothing iteration $\mathcal{S}_{j,1,1}$ and pointwise GS-LEX as constraint smoother.

[3] I.e., the inequalities in (5.5) are strict.

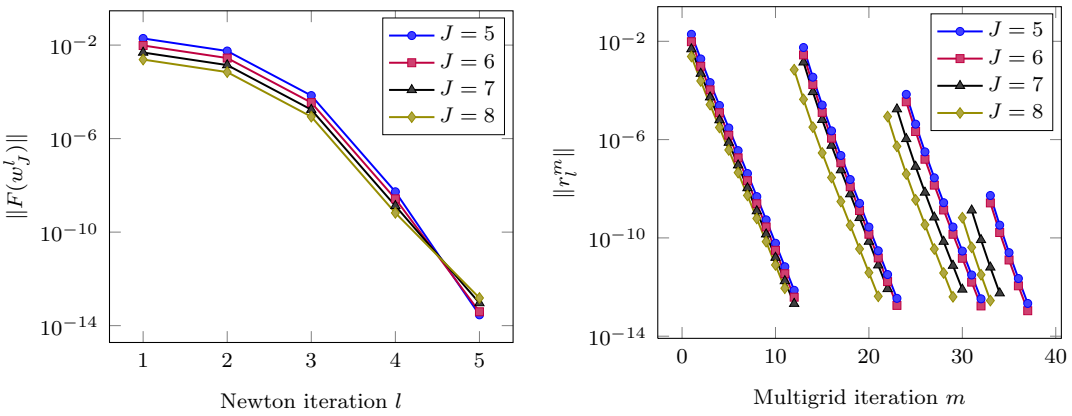



Figure 6.1: Convergence histories for test case 1 for $J = 5, 6, 7, 8$. On the left, we show the nonlinear residual $F(w_J^l)$, on the right, the linear residual $r_l^m$ for all multigrid steps is shown..

### 6.4.1 A Model Problem

In the first section, we consider test problems with nonlinear terms which satisfy Assumptions 6.4. We consider an equality constrained problem, i.e. $\mathcal{U}_{\text{ad}} = L^2(\Omega)$, and a control-constrained problem.

**The Equality Constrained Case** For this problem, the nonlinear term is given by $g(y) = y^3$ in (6.1) on $\Omega = (0, 1)^2$. The target state is

$$\bar{y} = 8 \sin(\pi x) \sin(\pi y) - 4, \tag{6.38}$$

which is not attainable due to the boundary conditions. The results are presented in Figure 6.1. The left part shows the iteration history of the nonlinear residual $\|F(w_J^l)\|$. We observe superlinear convergence, which furthermore does not depend on $J$. To the right, we show the corresponding iteration histories of the inner, linear residual (6.24). The convergence independent of $J$ is confirmed, and in this case, for all Newton steps the reduction factors are the same.

**The Inequality Constrained Case** We consider the nonlinear function $g(y) = y + y^3$, the additional box-constraint

$$-4 \leq u \leq 0$$

and

$$\bar{y} = \sin(2\pi x) \sin(2\pi y) \exp 2x/6. \tag{6.39}$$

All other parameters stay the same. This example has been considered in [92]. The computed optimal control $u_J^*$, the active and inactive sets $\mathcal{A}^*, \mathcal{I}^*$ and the optimal state $y_J^*$ are depicted in Figure 6.2. In Figure 6.3, we display results with respect

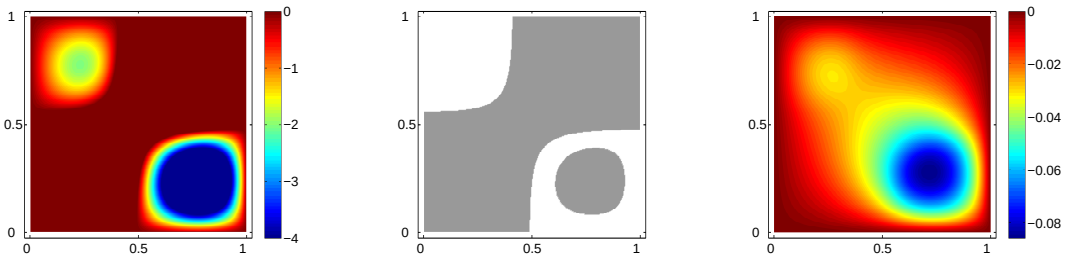


Figure 6.2: Computed optimal control $u_J^*$ (left), associated inactive set $\mathcal{I}^*$ and active set $\mathcal{A}^*$ (shaded region, middle) and optimal state $y_J^*$ for the control-constrained model problem on a mesh with $h_J = 2^{-8}$.

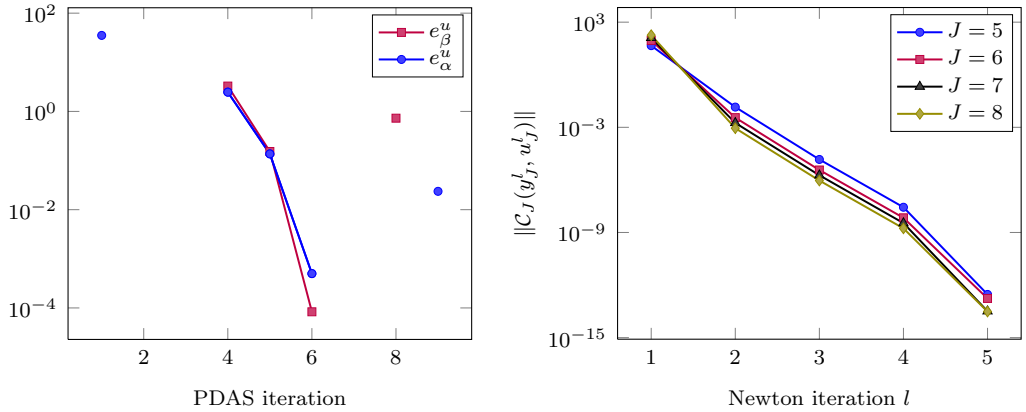



Figure 6.3: Iteration history of $e_\alpha^u, e_\beta^u$ for all QP substeps (left) and feasibility with respect to $\mathcal{C}_J(y_J^l, u_J^l)$ for all Newton steps (right).

to the feasibility of the intermediate iterates. On the left, we show the errors $e_\alpha^u, e_\beta^u$ corresponding to the inequality constraints for all intermediate QP steps $k$. Note that this error vanishes if an iterate is feasible. The error history is shown exemplary for the case $J = 8$, the results are visually indistinguishable for other values of $J$. On the right, we plot $\|C_J(y_J^l, u_J^l)\|$, i.e. the feasibility with respect to the equality constraint, for all Newton iterations $l$ and levels $J = 5, 6, 7, 8$. Superlinear convergence occurs in the final iteration, and the results do not depend on $J$.

### 6.4.2 Optimal Control of a Steady-State Solid Fuel Ignition Model

In this section we study the optimal control of the steady-state solid fuel ignition or Bratu problem. The constraint $\mathcal{C}(y, u)$ is defined as

$$\Delta y + \delta e^y = u \quad \text{in } \Omega \tag{6.40}$$

with homogeneous Dirichlet boundary conditions and $\delta \in (0, \delta^*]$. The Bratu problem appears in the theory of thermal self-ignition of an enclosed chemically reactive mixture. The solution $y$ represents temperature differences between the interior of $\Omega$ and $\partial\Omega$. For $\Omega = [0,1]^2$, equation (6.40) has at most two solutions. For $\delta = \delta^*$, the so-called turning point, (6.40) has exactly one solution. The value of $\delta^*$ has been estimated numerically and in several publications is reported as approximately 6.81. We remark that the difficulty of the Bratu problem consists in finding the second solution for $\delta$ close to $\delta^*$. Optimal control of the Bratu problem has been studied numerically in [35], where it has been reported that in the optimal control case even $\delta > \delta^*$ yields a solution (to the controlled problem).

The remaining data for the test case is as follows. We consider the target state

$$\bar{y} = \frac{45}{\pi^2} \sin(\pi x) \sin(\pi y). \tag{6.41}$$

As initial values we set $y_J^0 = u_J^0 = \bar{y}_J$ and $p_J^0 \equiv 0$. The regularization parameter is $\sigma = 8.25_{-5}$, and $\delta = 3.0$. The results are reported in Table 6.1. We have computed solutions for $J = 5, 6, 7, 8$, where $h_J = 2^{-J+3}$. For every computation, the outer Newton iteration converged in 8 iterations. For each $J$ and each $l = 0, \ldots, 8$, we report data as follows. In the first row, we give the nonlinear residual $\|F(w_J^l)\|$, in the second row we report $s_l = -\log_2 \alpha_l$, which is the number of backtracking trial steps in the line search, and $s_l = 0$ corresponds to acceptance of the full Newton step with $\alpha_l = 1$. In the third row the number of linear iterations, i.e. multigrid cycles, required to satisfy the stopping criterion $\|r_l^m\| \leq 1_{-12}$, is given. In the fourth and fifth row, the current values of the objective functional $\mathcal{J}_J(y_J^l, u_J^l)$ and of the constraint residual $\|\mathcal{C}_J(y_J^l, u_J^l)\|$ of (6.40) are given, respectively.

Due to the non-optimal choice of the initial value, in the initial phase of the iteration, full Newton steps are rejected by the merit function. Naturally, the iteration number of the inner solver differs for each $l$, depending on the operator in (6.18), which might be ill-conditioned. Correspondingly, the final number $\bar{m}$ of linear iterations necessary to obtain the desired decrease of $r_l^m$ is large and for some $l$ the maximum prescribed number of 20 is reached. Nevertheless, $\bar{m}_l$ does not depend on $J$, again showing the mesh-independent convergence of the inner multigrid solver. Note that in the final phase, full Newton steps are accepted by the line search and the convergence is rapid. Furthermore, in the neighborhood of the minimizer, the condition number of the operator in (6.18) seems to improve and the required number of multigrid iterations decreases. We remark that $\|F(w_J^l)\|$ increases in the first iteration, however $\mathcal{J}_J(y_J^l, u_J^l)$ decreases monotonically.

For comparison, in Table 6.2 we present the analogous results which have been obtained with the inexact Newton strategy according to Section 6.3.1. From the given data we observe that the line search method does not interfere with the inexactness of inner solves and again, once full Newton steps are accepted, rapid convergence occurs. Furthermore, also in the inexact case, neither the number of Newton steps nor the

Table 6.1: Results for the test problem with the constraints (6.40). Reported are the nonlinear residual $\|F(w_J^l)\|$, the number of trial steps $s_l$ in the backtracking line search, the number of multigrid cycles $\bar{m}_l$ in order to converge (6.18) to a tolerance of $\|r_l^m\| \leq 1_{-12}$, the value of the objective functional $\mathcal{J}_J(y_J, u_J)$, and the constraint residual $\|\mathcal{C}_J(y_J, u_J)\|$.

| $J$ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | $\|F(w_J^l)\|$ | $2.944_{-2}$ | $4.985_{-1}$ | $4.683_{-1}$ | $4.112_{-1}$ | $2.100_{-1}$ | $1.222_{-3}$ | $1.187_{-4}$ | $2.893_{-9}$ | $1.643_{-13}$ |
| | $s_l$ | — | 0 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 12 | 20 | 20 | 20 | 19 | 17 | 12 | 6 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.150_{-2}$ | $4.802_{-2}$ | $4.017_{-2}$ | $2.392_{-2}$ | $4.398_{-2}$ | $4.410_{-2}$ | $4.412_{-2}$ | $4.412_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $2.344_{-2}$ | $4.985_{-1}$ | $4.683_{-1}$ | $4.112_{-1}$ | $2.100_{-1}$ | $1.211_{-3}$ | $1.187_{-4}$ | $2.893_{-9}$ | $1.643_{-13}$ |
| 6 | $\|F(w_J^l)\|$ | $1.472_{-2}$ | $2.493_{-1}$ | $2.342_{-1}$ | $2.056_{-1}$ | $1.050_{-1}$ | $6.104_{-4}$ | $5.924_{-5}$ | $1.441_{-9}$ | $1.556_{-13}$ |
| | $s_l$ | — | 0 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 11 | 20 | 20 | 20 | 18 | 16 | 12 | 6 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.150_{-2}$ | $4.803_{-2}$ | $4.017_{-2}$ | $2.393_{-2}$ | $4.398_{-2}$ | $4.410_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $1.172_{-2}$ | $2.493_{-1}$ | $2.342_{-1}$ | $2.056_{-1}$ | $1.050_{-1}$ | $6.049_{-4}$ | $5.924_{-5}$ | $1.441_{-9}$ | $1.574_{-13}$ |
| 7 | $\|F(w_J^l)\|$ | $7.359_{-3}$ | $1.246_{-1}$ | $1.171_{-1}$ | $1.028_{-1}$ | $5.249_{-2}$ | $3.052_{-4}$ | $2.961_{-5}$ | $7.198_{-10}$ | $2.755_{-13}$ |
| | $s_l$ | — | 0 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 11 | 20 | 20 | 20 | 18 | 16 | 12 | 6 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.150_{-2}$ | $4.803_{-2}$ | $4.017_{-2}$ | $2.393_{-2}$ | $4.398_{-2}$ | $4.410_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $5.859_{-3}$ | $1.246_{-1}$ | $1.171_{-1}$ | $1.028_{-1}$ | $5.249_{-2}$ | $3.024_{-4}$ | $2.961_{-5}$ | $7.197_{-10}$ | $2.805_{-13}$ |
| 8 | $\|F(w_J^l)\|$ | $1.840_{-3}$ | $3.116_{-2}$ | $2.927_{-2}$ | $2.570_{-2}$ | $1.312_{-2}$ | $7.629_{-5}$ | $7.401_{-6}$ | $1.799_{-10}$ | $1.095_{-12}$ |
| | $s_l$ | — | 0 | 4 | 3 | 1 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 20 | 20 | 20 | 20 | 17 | 15 | 12 | 5 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.150_{-2}$ | $4.803_{-2}$ | $4.017_{-2}$ | $2.393_{-2}$ | $4.398_{-2}$ | $4.410_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $1.465_{-3}$ | $3.116_{-2}$ | $2.927_{-2}$ | $2.570_{-2}$ | $1.312_{-2}$ | $7.559_{-5}$ | $7.401_{-6}$ | $1.799_{-10}$ | $1.112_{-12}$ |

Table 6.2: Results for the test problem with the constraints (6.40). The same data as in Table 6.1 is reported. Here, the inexact Newton strategy has been employed to drive the accuracy of inner system solves.

| $J$ | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | $\|F(w_J^l)\|$ | $1.472_{-2}$ | $2.494_{-1}$ | $2.367_{-1}$ | $1.912_{-1}$ | $1.914_{-2}$ | $7.478_{-5}$ | $7.641_{-7}$ | $4.532_{-12}$ | $1.377_{-13}$ |
| | $s_l$ | — | 0 | 4 | 2 | 0 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 3 | 3 | 2 | 2 | 5 | 11 | 7 | 17 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.153_{-2}$ | $4.777_{-2}$ | $3.254_{-2}$ | $3.490_{-2}$ | $4.409_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $1.172_{-2}$ | $2.494_{-1}$ | $2.367_{-1}$ | $1.912_{-1}$ | $1.914_{-2}$ | $7.424_{-5}$ | $7.641_{-7}$ | $4.530_{-12}$ | $1.400_{-13}$ |
| 6 | $\|F(w_J^l)\|$ | $7.359_{-3}$ | $1.247_{-1}$ | $1.183_{-1}$ | $9.561_{-2}$ | $9.587_{-3}$ | $3.742_{-5}$ | $3.827_{-7}$ | $2.248_{-12}$ | $2.730_{-13}$ |
| | $s_l$ | — | 0 | 4 | 2 | 0 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 3 | 3 | 2 | 2 | 5 | 10 | 7 | 17 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.153_{-2}$ | $4.777_{-2}$ | $3.254_{-2}$ | $3.488_{-2}$ | $4.409_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $5.859_{-3}$ | $1.247_{-1}$ | $1.183_{-1}$ | $9.560_{-2}$ | $9.587_{-3}$ | $3.715_{-5}$ | $3.827_{-7}$ | $2.247_{-12}$ | $2.769_{-13}$ |
| 7 | $\|F(w_J^l)\|$ | $3.680_{-3}$ | $6.234_{-2}$ | $5.917_{-2}$ | $4.781_{-2}$ | $4.796_{-3}$ | $1.871_{-5}$ | $1.914_{-7}$ | $1.239_{-12}$ | $5.474_{-13}$ |
| | $s_l$ | — | 0 | 4 | 2 | 0 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 3 | 3 | 2 | 2 | 5 | 10 | 7 | 17 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.153_{-2}$ | $4.777_{-2}$ | $3.254_{-2}$ | $3.488_{-2}$ | $4.409_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $2.930_{-3}$ | $6.234_{-2}$ | $5.917_{-2}$ | $4.780_{-2}$ | $4.796_{-3}$ | $1.858_{-5}$ | $1.914_{-7}$ | $1.243_{-12}$ | $5.556_{-13}$ |
| 8 | $\|F(w_J^l)\|$ | $1.840_{-3}$ | $3.117_{-2}$ | $2.959_{-2}$ | $2.390_{-2}$ | $2.398_{-3}$ | $9.357_{-6}$ | $9.573_{-8}$ | $1.227_{-12}$ | $1.093_{-12}$ |
| | $s_l$ | — | 0 | 4 | 2 | 0 | 0 | 0 | 0 | 0 |
| | $\bar{m}_l$ | — | 3 | 3 | 2 | 2 | 5 | 10 | 7 | 16 |
| | $\mathcal{J}_J(y_J^l, u_J^l)$ | 2.599 | $5.153_{-2}$ | $4.777_{-2}$ | $3.254_{-2}$ | $3.488_{-2}$ | $4.409_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ | $4.413_{-2}$ |
| | $\|\mathcal{C}_J(y_J, u_J)\|$ | $1.465_{-3}$ | $3.117_{-2}$ | $2.958_{-2}$ | $2.390_{-2}$ | $2.398_{-3}$ | $9.290_{-6}$ | $9.573_{-8}$ | $1.242_{-12}$ | $1.110_{-12}$ |

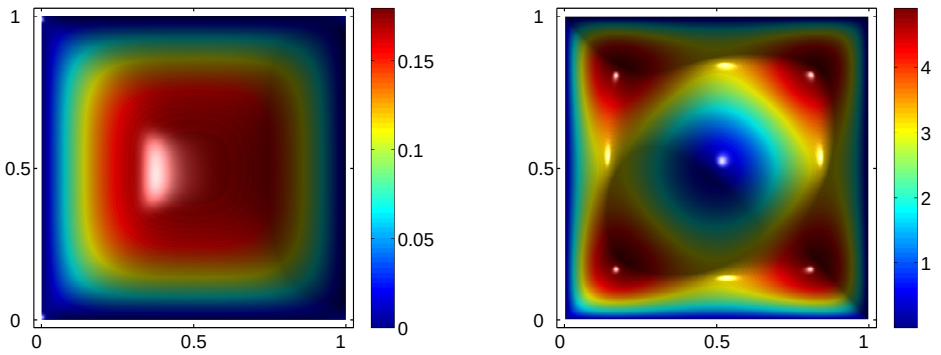



Figure 6.4: Computed optimal state $y_J^*$ (left) and optimal control $u_J^*$ (right) for (6.42) on a mesh with $h_J = 2^{-8}$.

number of multigrid iterations per Newton step depends on $J$. The total number of multigrid cycles, $\sum_l \bar{m}_l$, however, is drastically reduced, e.g. from 139 to 48 for $J = 8$.

### 6.4.3 Optimal Control of a Semilinear Equation Associated with a Scalar Ginzburg-Landau Model

As a final example we consider the optimal control of

$$-\Delta y - y + y^3 = u \text{ in } \Omega = (0,1)^2 \tag{6.42}$$

with homogeneous Dirichlet boundary conditions. This equation appears within a a simplified Ginzburg-Landau model for superconductivity [80], and $y$ is the wave-function in the absence of internal magnetic fields. Note that (6.11) does not hold for (6.42), however the surjectivity of $D_{y,u}\mathcal{C}$ and well-posedness of the optimal control problem have been proved in [80]. This model has also been considered in [100].

**The unconstrained case $\mathcal{U}_{\mathsf{ad}} = L^2(\Omega)$** Figure 6.4 pictures the computed optimal state $y_J^*$ on the left and the computed optimal control $u_J^*$ on the right on a mesh with $h_J = 2^{-8}$. In Figure 6.5, we show iteration histories of the corresponding computation. On the left, the linear residual $r_J^l$ is shown for each multigrid iteration $m$. On the right, the norm of the nonlinear residual, $\|F(w_J^l)\|$, is plotted for all Newton steps $l$. The results in the top row have been computed with the fixed tolerance for the inner system solves, whereas in the bottom row, the inexact Newton strategy has been employed. We observe in both cases that the nonlinear residual converges at a superlinear rate and independent of $J$. However, the inexact strategy requires only a total of 15 multigrid cycles, in contrast to the double computational work for the fixed tolerance method. In both cases, also the reduction factor of the multigrid method does not depend on $J$.

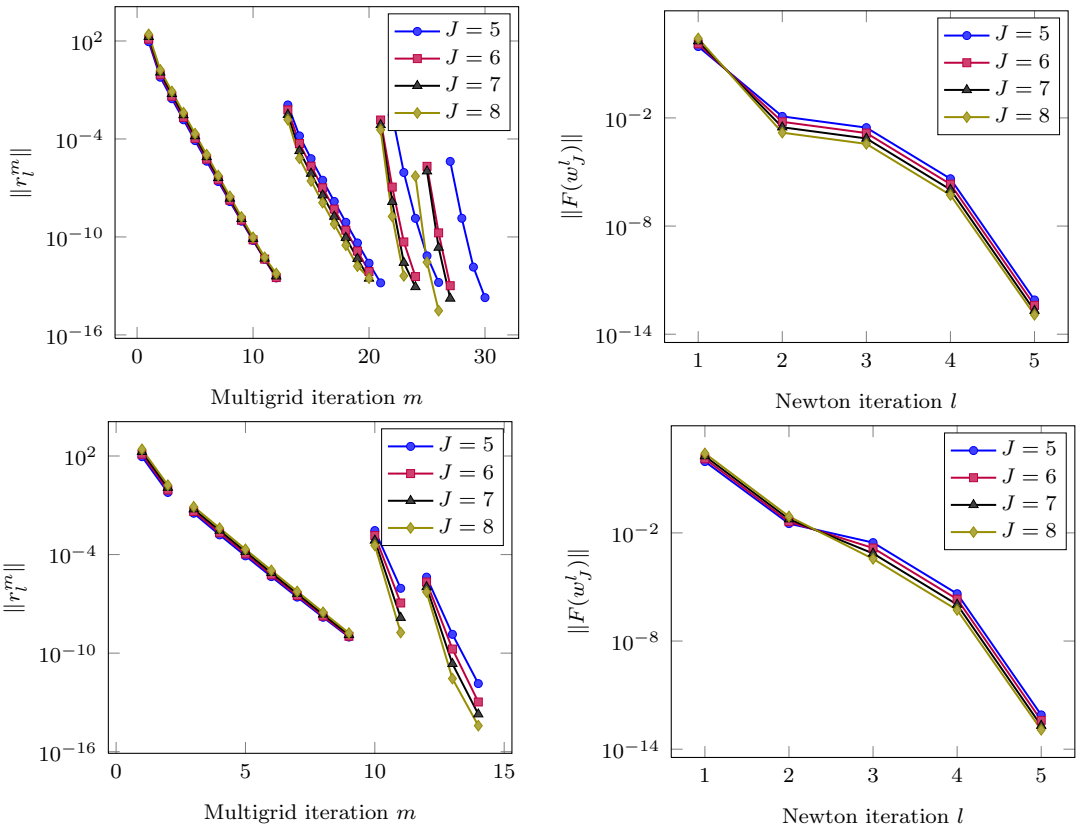



Figure 6.5: Iteration histories for the unconstrained example (6.42). On the left, the norm of the linear residual $r_l^m$ is plotted for all multigrid iterations $m$, on the right the nonlinear residual norm $\|F(w_J^l)\|$ is plotted for all Newton steps $l$. In the top row, the fixed stopping tolerance for the multigrid solution has been enforced, whereas the results in the bottom row are computed with the inexact Newton strategy.

**The constrained case** In this section we consider the control of problem (6.42) with the additional inequality constraints

$$u^\alpha = 1.5 \text{ and } u^\beta = 4.0. \tag{6.43}$$

In Figure 6.6 we depict the computed optimal control $u_J^*$ and the corresponding inactive and active set $\mathcal{I}^*$, $\mathcal{A}^*$ on a mesh with $h_J = 2^{-8}$. Figure 6.7 shows the development of feasibility of the iterates for the computation with $J = 8$. On the left, the errors $e_\alpha^u, e_\beta^u$ are shown for every PDAS step $k$, on the right, the residual of the equality constraint $\mathcal{C}_J(y_J^l, u_J^l)$ is plotted for every SQP step $l$. Compared are the results for the fixed tolerance method and the inexact Newton strategy. The number of SQP steps is the same for both methods and the data matches closely. However, the inexact Newton method achieves these results with far fewer multigrid iterations. The

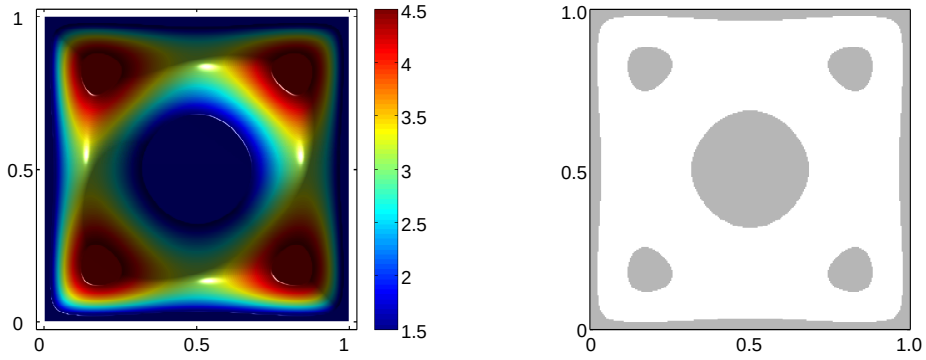



Figure 6.6: Computed optimal control $u_J^*$ (left) and associated inactive set $\mathcal{I}^*$ and active set $\mathcal{A}^*$ (shaded region, right) for (6.42) with bounds (6.43) on a mesh with $h_J = 2^{-8}$.

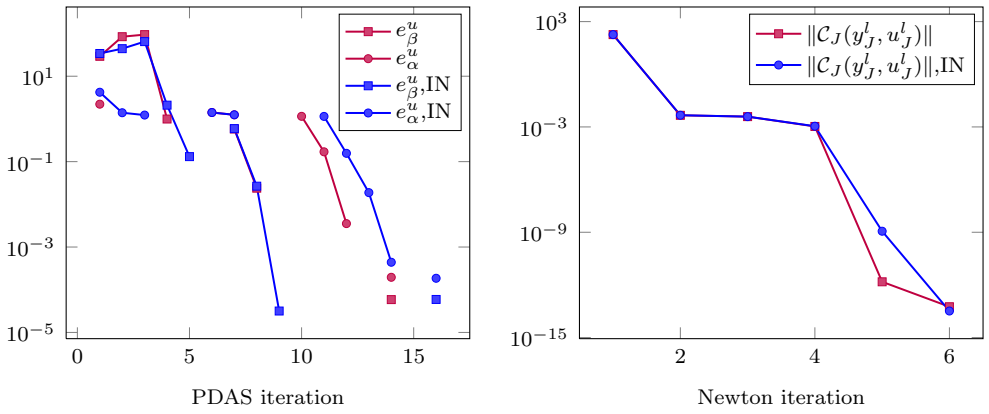



Figure 6.7: Left: error with respect to inequality constraints. Right: Norm of state equation residual, $\|\mathcal{C}_J(y_J^l, u_J^l)\|$.

detailed data is given in Table 6.3. The inexact method for $J = 8$ (marked with $^*$), required one additional SQP with 1 QP step and 11 multigrid cycles in order to reach $\|F_h(w_h^6)\| \sim 6.513_{-14}$.

Finally, Figure 6.8 shows the optimal control $u_J^*$ on the left and the associated active set $\mathcal{A}^*$ (the shaded region, right) which has been obtained with $\sigma = 1_{-4}$. The control is almost of bang-bang-type and correspondingly the inactive set degenerates to the so-called switching region.

## Summary

We presented a full SQP-multigrid method for the solution of semilinear constrained optimization problems. To this end, the multigrid method of Chapter 4 has been

Table 6.3: Results for the simplified Ginzburg-Landau model with control-constraints. For each SQP step $l$, we report the norm of the nonlinear residual, $\|F(w_J^l)\|$, the number of PDAS steps $\bar{k}$ and the number of multigrid iterations $\bar{m}$ per PDAS step.

| $J$ | | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|---|
| 6 | $\|F(w_J^{l,\mathcal{I}})\|$ | $1.282_{2}$ | $9.194_{-3}$ | $7.732_{-3}$ | $2.132_{-3}$ | $3.005_{-12}$ | $2.092_{-14}$ |
| | $\bar{k}$ | — | 5 | 4 | 4 | 2 | 1 |
| | $\bar{m}$ | — | 13,20,20,20,13 | 16,10,11,8 | 12,18,10,6 | 9,4 | 9 |
| 6, IN | $\|F(w_J^{l,\mathcal{I}})\|$ | $1.282_{2}$ | $9.284_{-3}$ | $7.716_{-3}$ | $2.198_{-3}$ | $3.509_{-9}$ | $1.629_{-14}$ |
| | $\bar{k}$ | — | 5 | 5 | 5 | 2 | 1 |
| | $\bar{m}$ | — | 13,20,20,20,15 | 16,10,11,8 | 12,17,10,6 | 9,4 | 8 |
| 7 | $\|F(w_J^{l,\mathcal{I}})\|$ | $1.812_{2}$ | $4.597_{-3}$ | $3.866_{-3}$ | $1.066_{-3}$ | $1.505_{-12}$ | $5.721_{-14}$ |
| | $\bar{k}$ | — | 5 | 4 | 4 | 2 | 1 |
| | $\bar{m}$ | — | 13,20,20,20,14 | 15,10,11,7,3 | 11,17,10,6 | 8,4 | 8 |
| 7, IN | $\|F(w_J^{l,\mathcal{I}})\|$ | $1.812_{2}$ | $4.642_{-3}$ | $3.858_{-3}$ | $1.100_{-3}$ | $1.144_{-9}$ | $3.258_{-14}$ |
| | $\bar{k}$ | — | 5 | 5 | 5 | 2 | 1 |
| | $\bar{m}$ | — | 2,2,2,2,2 | 9,5,4,2,2 | 2,2,2,2,2 | 2,2 | 11 |
| 8 | $\|F(w_J^{l,\mathcal{I}})\|$ | $2.561_{2}$ | $2.299_{-3}$ | $1.933_{-3}$ | $5.330_{-4}$ | $1.012_{-6}$ | $6.654_{-14}$ |
| | $\bar{k}$ | — | 5 | 5 | 4 | 2 | 1 |
| | $\bar{m}$ | — | 2,2,2,2,2 | 9,5,5,2,2 | 2,2,2,2 | 2,2 | 11 |
| 8, IN* | $\|F(w_J^{l,\mathcal{I}})\|$ | $2.561_{2}$ | $2.321_{-3}$ | $1.929_{-3}$ | $5.502_{-4}$ | $1.012_{-6}$ | $2.366_{-11}$ |
| | $\bar{k}$ | — | 5 | 5 | 5 | 3 | 1 (+1) |
| | $\bar{m}$ | — | 2,2,2,2,2 | 10,6,5,3,2 | 2,2,2,2,2 | 2,2,2 | 5 (+11) |

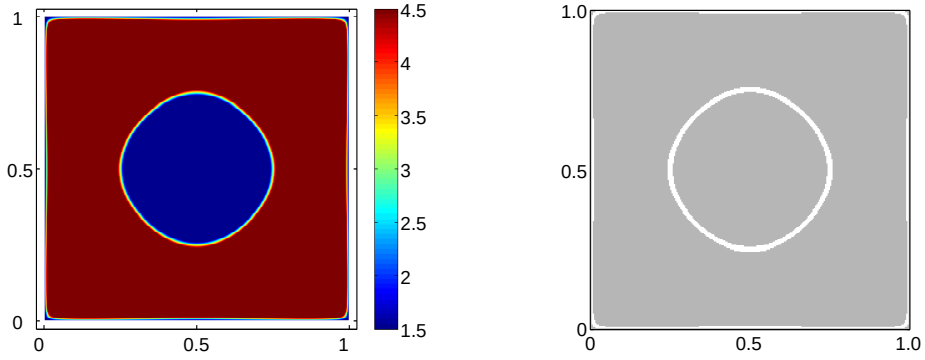



Figure 6.8: Computed optimal control $u_J^*$ (left) and corresponding active and inactive set $\mathcal{A}^*, \mathcal{I}^*$ (right) for (6.42) with bound constraints (6.43) and $\sigma = 1_{-4}$ on a mesh with $h_J = 2^{-8}$.

employed as linear solver for the Newton system arising at each step in the Lagrange-Newton method, and analogously the PDAS-multigrid method of Chapter 5 has been employed as QP solver within a full SQP method. Inexactness of inner system solves and globalization with the augmented Lagrangian merit function has been discussed. Numerical experiments have shown that the resulting SQP-multigrid method converges mesh-independent and at a locally superlinear rate. In particular, the inexact Newton strategy could significantly reduce the number of required multigrid cycles, without compromising on the fast convergence of the SQP iteration.

# 7 Conclusions

We have considered the efficient numerical solution of discretized PDE constrained optimization problems. In particular, we have studied the minimization of a quadratic functional subject to constraints by a linear or semilinear elliptic PDE with distributed control. Further, pointwise inequality constraints on the control unknown have been accounted for.

One main contribution consists in devising a coupled multigrid solver for the solution of linear-quadratic optimization problems. Deviating from the standard constraint elimination techniques, we have defined a smoothing iteration which eventually led to a simultaneous treatment of the optimization and adjoint variables in the multigrid process. A local mode analysis has shown that for discrete optimality systems, smoothing rates close to those of the underlying constraint PDE can be obtained. These findings have been corroborated with detailed numerical results. In particular, employing the full multigrid approach, our experiments clearly demonstrated that discretized PDE constrained optimization problems are solved with optimal cost $\mathcal{O}(n)$, where $n = 3h_J^{-2}$ is the total number of unknowns of the KKT system on the finest grid level $J$. Furthermore, the constant of proportionality with respect to the work which has to be expended for the solution of the constraint PDE, has been found to range between 8 and 10. This underlines the superiority of our approach over common elimination strategies.

We have included numerical experiments of problems with constraints where simple pointwise smoothing is known to fail already for the underlying PDE. Specifically, we have considered the anisotropic diffusion equation and convection-diffusion. The framework of our method allows to include powerful line smoothers or ILU-factorizations, such that these more challenging problems could be solved efficiently. In all cases, it has been shown that convergence rates do not depend on $h_J$ and discrete optimality systems can be solved with a small multiple of the computational cost required to solve the underlying constraint PDE.

Furthermore, the role of the regularization parameter $\sigma$ has been discussed in some detail. It has been shown that the convergence rate is robust with respect to both $J$ and $\sigma$ under a mild restriction on the next to coarsest mesh size $h_1$. A close inspection of the smoothing iteration with respect to the control unknown $u_h$ revealed a way for a possible improvement based on spectral filtering of the reduced Hessian $H_j^\sigma = \sigma M_j + S_j^* M_j S_j$. This allowed us to weaken the mesh size restriction, and with a negligible amount of additional computational work, problems with near-vanishing $\sigma$ could be treated efficiently. Numerical experiments have shown that for fine discretizations $h_J$,

robust convergence is obtained with rates which are independent of the regularization parameter $\sigma$, the coarsest mesh size $h_0$, and the number of levels $J$.

In order to treat linear-quadratic problems with pointwise inequality constraints on the control, the multigrid approach has been modified to solve subproblems generated by a primal-dual active set strategy (PDAS). Numerical experiments demonstrated that such problems can be solved with a high efficiency due to mesh-independent convergence of both the outer PDAS method and the inner multigrid solver.

In a final step, the PDAS-multigrid method has been incorporated in the sequential quadratic programming (SQP) framework. Inexact Newton techniques have been implemented to further enhance computational efficiency, and globalization of the SQP method has been accounted for with a line search based on the augmented Lagrangian merit function. Numerical experiments highlight the efficiency of the resulting full SQP-multigrid approach. In all cases, locally superlinear convergence of the outer SQP iteration has been observed. In combination with the mesh-independent convergence of the inner solver, a solution method with optimal cost has been obtained.

It is indisputable that several issues have to be dealt with in order to tackle large-scale problems stemming from industrial applications. Naturally, extension to three space dimensions, possibly in conjunction with parallelization, is important for real-world applications. Due to the modular structure of our approach, strategies based on the parallelization of the constraint PDE, such as domain decomposition methods, should be readily extensible to our approach. Still, some issues such as agglomeration of coarse grids, which may be required to obtain a satisfactory parallel efficiency, have to be considered, but we expect them to carry over analogously to the scalar case.

Extensions are obvious to related problem classes such as boundary control. A field with numerous applications is flow control [79]. There, the constraint PDE itself, e.g. the Navier-Stokes equations, exhibits saddle point structure (2.54). Consequently, suitable smoothing iterations, such as the Vanka- or the Braess-Sarazin-smoother [37, 151], are required at the constraint level.

A further class of problems with many interesting applications is given by time-dependent problems. Here, the cost functional typically involves integrals over a time interval, and additional difficulties arise, which are mostly due to the backwards-in-time nature of the adjoint equation. Ideally, at every discrete instant in time, the solution of the state and the adjoint equation should be available, however this is computationally infeasible and has given rise to a number of techniques such as reduced order modeling [132] and checkpointing [77]. We see a possible application of our method in the frame of instantaneous control or suboptimal feedback [93]. There, at each time step a problem resembling (2.51) is obtained. In general, the development of a coupled space-time multigrid approach still poses a significant challenge.

In conclusion, the promising results obtained with the presented multigrid method provide a strong incentive to further advance the application of coupled multigrid methods for problems arising in the large and active area of PDE constrained optimization.

# A Appendix

## A.1 Saddle Point Systems in Hilbert Spaces

In this section we briefly review how the model problem (2.18) fits into the standard approximation theory for saddle point problems [43]. To this end, consider the problem of minimizing the quadratic functional

$$J(x) = \frac{1}{2}a(x,x) - \langle f, x\rangle \tag{A.1}$$

subject to the linear constraints

$$b(x,q) = \langle g, q\rangle \quad \text{for all } q \in \Lambda. \tag{A.2}$$

Here, $a(\cdot\,,\cdot)$ is a symmetric bilinear form on $W \times W$, $b(\cdot\,,\cdot)$ is a bilinear form on $W \times \Lambda$ and $f, g$ are continuous linear functionals in $W'$ and $\Lambda'$, respectively. The spaces $W$ and $\Lambda$ are Hilbert spaces and $W'$ and $\Lambda'$ are their dual spaces. We define the subspace

$$Z = \{x \in W : b(x,q) = 0 \ \forall q \in \Lambda\}. \tag{A.3}$$

and furthermore require that $a(\cdot\,,\cdot)$ and $b(\cdot\,,\cdot)$ satisfy the following conditions:

$$a(x,v) \le C_a\|x\|\|v\|, \qquad \forall x, v \in W, \tag{A.4}$$
$$a(x,x) \ge 0, \qquad \forall x \in W, \tag{A.5}$$
$$a(x,x) \ge c_a\|x\|^2, \qquad \forall x \in Z, \tag{A.6}$$
$$b(x,q) \le C_b\|x\|\|q\|, \qquad \forall x \in W, q \in \Lambda, \tag{A.7}$$
$$\sup_{x\in W, x\neq 0} \frac{b(x,q)}{\|x\|} \ge c_b\|q\|, \qquad \forall q \in \Lambda, \tag{A.8}$$

with positive constants $C_a, c_a, C_b, c_b$. Then there exists a unique solution $(x,p) \in W\times\Lambda$ of the saddle point problem

$$\begin{aligned} a(x,v) + b(v,p) &= \langle f, v\rangle, && \forall v \in W, \\ b(x,q) &= \langle g, q\rangle && \forall q \in \Lambda. \end{aligned} \tag{SP}$$

Furthermore, the solution satisfies

$$\|x\| + \|p\| \le C(\|f\| + \|g\|) \tag{A.9}$$

and $x$ is a minimizer of (A.1) satisfying (A.2). The symmetry of $a(\cdot\,,\cdot)$ as well as (A.5) are required only to link the minimization problem (A.1), (A.2) to the saddle point problem (SP), they are not required in order to prove existence and uniqueness of a solution to (SP).

A conforming Galerkin approximation to (SP) is obtained by restricting (SP) to suitable finite dimensional subspaces $W_h \subset W$ and $\Lambda_h \subset \Lambda$. Existence and uniqueness of a solution for the finite dimensional problem follows if assumptions (A.4) – (A.8) hold also on the corresponding subspaces. It is well-known that (A.6) and (A.8) have to be proved for the specific choice of $W_h$ and $\Lambda_h$ since in general $Z_h \not\subset Z$. If all corresponding assumptions hold, the discrete saddle point system has a unique solution $(x_h, p_h) \in W_h \times \Lambda_h$ and

$$\|x - x_h\| + \|p - p_h\| \leq C\big(\inf_{v_h \in W_h} \|x - v_h\| + \inf_{q_h \in \Lambda_h} \|p - q_h\|\big) \tag{A.10}$$

holds. By choosing appropriate bases in $W_h$ and $\Lambda_h$ one obtains a symmetric indefinite system of the form (2.54).

Now we define the quadratic functional

$$J(y,u) = \frac{1}{2} a_1(y - \bar{y}, y - \bar{y}) + \frac{1}{2} a_2(u,u), \quad y \in Y, u \in U, \tag{A.11}$$

with the symmetric bilinear forms $a_1(\cdot\,,\cdot)$ on $Y \times Y$ and $a_2(\cdot\,,\cdot)$ on $U \times U$. The linear constraints in variational form are given in terms of additional bilinear forms $b_1(\cdot\,,\cdot)$ on $Y \times \Lambda$ and $b_2(\cdot\,,\cdot)$ on $U \times \Lambda$ through

$$b_1(y,q) + b_2(u,q) = \langle g, q \rangle, \quad q \in \Lambda. \tag{A.12}$$

The optimization problem is given by the minimization of (A.11) such that (A.12) is satisfied. With the identifications $W = Y \times U$, $w = (y,u)$, $a(x,x) = a_1(y,y) + a_2(u,u)$ and $b(x,q) = b_1(y,q) + b_2(u,q)$ this problem can be written as (A.1), (A.2). Furthermore, if all bilinear forms appearing in (A.11) and (A.12) are continuous and in addition the assumptions

$$a_1(y,y) \geq 0 \qquad \forall y \in Y, \tag{A.13}$$

$$a_2(u,u) \geq k_a \|u\|^2 \qquad \forall u \in U, \tag{A.14}$$

$$\sup_{q \in \Lambda, q \neq 0} \frac{b_1(y,q)}{\|q\|} \geq k_b \|y\|, \qquad \forall y \in Y, \tag{A.15}$$

$$\sup_{u \in U, u \neq 0} \frac{b_1(y,q)}{\|u\|} > 0, \qquad \forall q \in \Lambda, \tag{A.16}$$

hold, then the optimal control problem satisfies the conditions (A.4) – (A.8) with $c_a = \frac{1}{2} k_a \min\{1, \frac{k_b^2}{C_{b_1}^2}\}$ and $c_b = k_b$, where $C_{b_1}$ is the continuity constant of the form

$b_1(\cdot\,,\cdot)$, see [31]. The significance of this assertion is that in particular the stability conditions (A.15),(A.16) of the constraints (A.12) imply the inf-sup condition (A.8) and as such are crucial for the uniform stability of the optimal control problem. On the other hand, the conditions (A.15),(A.16) appear naturally when employing Galerkin approximations for the discretization of (A.12). We conclude that stability of the optimization problem follows from the stability of the constraint equation and coercivity of the functional.

For (2.18), the appropriate choices for $Y, U$ and $\Lambda$ have been given in in Chapter 2, i.e. $Y = \Lambda = H_0^1(\Omega)$ and $U = L^2(\Omega)$. The bilinear forms are defined by

$$a_1(y,z) = \int_\Omega yz \; d\Omega, \tag{A.17}$$

$$a_2(u,q) = \sigma \int_\Omega uq \; d\Omega, \tag{A.18}$$

$$b_1(y,z) = \int_\Omega \nabla y \cdot \nabla z \; d\Omega, \tag{A.19}$$

$$b_2(u,z) = -\int_\Omega uz \; d\Omega, \tag{A.20}$$

and the functionals are given by

$$\langle f, x\rangle = \int_\Omega \bar{y} y d\Omega, \tag{A.21}$$

$$\langle g, q\rangle = 0. \tag{A.22}$$

Note that $k_a = \sigma$, and the regularization parameter enters the stability condition of the optimal control problem. The discrete approximation follows by choosing appropriate subspaces, as has been discussed in Section 2.3.2. The inf-sup condition for the specific choice of $Y_h, U_h$, i.e. the lowest-order Raviart-Thomas spaces $RT_0$, has been proved in [43] for the rectangular case and in [12] for the general quadrilateral case.

# Bibliography


[1] I. Aavatsmark, T. Barkve, Ø. Bøe, and T. Mannseth, *Discretization on non-orthogonal, quadrilateral grids for inhomogeneous, anisotropic media*, J. Comput. Phys., 127 (1996), pp. 2–14.

[2] R. A. Adams, *Sobolev Spaces*, Academic Press, 1975.

[3] E. L. Allgower, K. Böhmer, F. A. Potra, and W. C. Rheinboldt, *A mesh-independence principle for operator equations and their discretizations*, SIAM J. Numer. Anal., 23 (1986), pp. 160–169.

[4] W. Alt, *The Lagrange-Newton method for infinite-dimensional optimization problems*, Numer. Funct. Anal. And Optimiz., 11 (1990), pp. 201–224.

[5] J. Appell and P. P. Zabrejko, *Nonlinear superposition operators*, Cambridge University Press, 1990.

[6] N. Arada, E. Casas, and F. Tröltzsch, *Error estimates for the numerical approximation of a semilinear elliptic control problem*, Comput. Optim. Appl., 23 (2002), pp. 201–229.

[7] T. Arbogast, C. N. Dawson, P. T. Keenan, M. F. Wheeler, and I. Yotov, *Enhanced cell-centered finite differences for elliptic equations on general geometry*, SIAM J. Sci. Comput., 19 (1998), pp. 404–425.

[8] T. Arbogast, M. F. Wheeler, and I. Ivan Yotov, *Mixed finite elements for elliptic problems with tensor coefficient as cell-centered finite differences*, SIAM J. Numer. Anal., 34 (1997), pp. 828–852.

[9] E. Arian and S. Ta'asan, *Multigrid one-shot methods for optimal control problems: infinite dimensional control*, Tech. Rep. 94-52, ICASE, NASA Langley Research Center, 1994.

[10] M. Arioli and L. Baldini, *A backward error analysis of a null space algorithm in sparse quadratic programming*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 425–442.

[11] M. Arioli, I. Duff, and D. Ruiz, *Stopping criteria for iterative solvers*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 138–144.

[12] D. N. Arnold, D. Boffi, and R. S. Falk, *Quadrilateral H(div) finite elements*, SIAM J. Numer. Anal., 42 (2005), pp. 2429–2451.

[13] K. J. Arrow, L. Hurwicz, and H. Uzawa, eds., *Studies in Linear and Nonlinear Programming*, Stanford University Press, Stanford, 4. print. ed., 1972.

[14] U. M. Ascher and E. Haber, *A multigrid method for distributed parameter estimation problems*, Electron. Trans. Numer. Anal., 15 (2002), pp. 1–17.

[15] O. Axelson and M. Neytcheva, *Preconditioning methods for linear systems arising in constrained optimization problems*, Numer. Linear Algebra Appl., 10 (2003), pp. 3–31.

[16] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[17] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc Web page*, 2001. http://www.mcs.anl.gov/petsc.

[18] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser Press, 1997, pp. 163–202.

[19] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition*, SIAM, Philadelphia, PA, 1994.

[20] A. Battermann and M. Heinkenschloss, *Preconditioners for Karush-Kuhn-Tucker matrices arising in the optimal control of distributed systems*, in Optimal Control of Partial Differential Equations, W. Desch, F. Kappel, and K. Kunisch, eds., Birkhäuser Verlag, 1998.

[21] R. Becker, H. Kapp, and R. Rannacher, *Adaptive finite element methods for optimal control of partial differential equations: Basic concept*, SIAM J. Control Optim., 39 (2000), pp. 113–132.

[22] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.

[23] M. Bergounioux, M. Haddou, M. Hintermüller, and K. Kunisch, *A comparison of a Moreau-Yosida-based active set strategy and interior point methods for constrained optimal control problems*, SIAM J. Optim., 11 (2000), pp. 495–521.

[24] M. Bergounioux, K. Ito, and K. Kunisch, *Primal-dual strategy for constrained optimal control problems*, SIAM J. Control Optim., 37 (1999), pp. 1176–1194.

[25] D. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J Control Optim, 20 (1982), pp. 221–246.

[26] L. T. Biegler, T. F. Coleman, A. R. Conn, and F. N. Santosa, eds., *Large-Scale Optimization with Applications. Part I: Optimization in Inverse Problems and Design*, vol. 92 of The IMA Volumes in Mathematics and its Applications, Springer, 1997.

[27] ——, eds., *Large-Scale Optimization with Applications. Part II: Optimal Design and Control*, vol. 93 of The IMA Volumes in Mathematics and its Applications, Springer, 1997.

[28] L. T. Biegler, O. Ghattas, and M. Heinkenschloss, eds., *Large-Scale PDE-Constrained Optimization*, Lecture Notes in Computational Science and Engineering, Springer, 2003.

[29] L. T. Biegler, J. Nocedal, and C. Schmid, *A reduced hessian method for large-scale constrained optimization*, SIAM J. Optim., 5 (1995), pp. 314–347.

[30] H. Blum, D. Braess, and F. T. Suttmeier, *A cascadic multigrid algorithm for variational inequalities*, Comput. Visual. Sci., 7 (2004), pp. 153–157.

[31] P. Bochev and M. D. Gunzburger, *Least-squares finite element methods for optimality systems arising in optimization and optimal control problems*, SIAM J. Numer. Anal., 43 (2006), pp. 2517–2543.

[32] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, Acta Numer., 4 (1995), pp. 1–51.

[33] A. Borzi, *Smoothers for control- and state-constrained optimal control problems*, Comput. Visual. Sci., 11 (2008), pp. 59–66.

[34] A. Borzi and G. Borzi, *An algebraic multigrid method for a class of elliptic differential systems*, SIAM J. Sci. Comput., 25 (2003), pp. 302–323.

[35] A. Borzi and K. Kunisch, *A multigrid scheme for elliptic constrained optimal control problems*, Comput. Optim. Appl., 31 (2005), pp. 309–333.

[36] A. Borzi, K. Kunisch, and D. Y. Kwak, *Accuracy and convergence properties of the finite difference multigrid solution of an optimal control optimality system*, SIAM J. Control Optim., 41 (2003), pp. 1477–1497.

[37] D. Braess and R. Sarazin, *An efficient smoother for the Stokes problem*, Appl. Numer. Math., 23 (1997), pp. 3–19.

[38] J. H. Bramble, R. E. Ewing, J. E. Pasciak, and J. Shen, *The analysis of multigrid algorithms for cell centered finite difference methods*, Adv. Comput. Math., 5 (1996), pp. 15–29.

[39] J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.

[40] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Math. Comp., 31 (1977), pp. 333–390.

[41] ——, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, no. 85 in GMD-Studien, GMD, St. Augustin, Germany, 1984.

[42] A. Brandt and C. W. Cryer, *Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems*, SIAM J. Sci. Stat. Comput., 4 (1983), pp. 655–684.

[43] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, Springer Series in Computational Mathematics, Springer, New York, 1991.

[44] M. Burger and R. Pinnau, *Fast optimal design of semiconductor devices*, SIAM J. Appl. Math., 64 (2003), pp. 108–126.

[45] C. Burstedde, *Fast Optimised Wavelet Methods for Control Problems Constrained by Elliptic PDEs*, PhD thesis, Mathematisch-Naturwissenschaftliche Fakultät, Rheinische Friedrich-Wilhelms-Universität Bonn, 2005.

[46] C. Burstedde and A. Kunoth, *Fast iterative solution of elliptic control problems in wavelet discretizations*, J. Comput. Appl. Math., 196 (2006), pp. 299–319.

[47] B. L. Buzbee, G. H. Golub, and C. W. Nielson, *On direct methods for solving Poisson's equations*, SIAM J. Numer. Anal., 7 (1970), pp. 627–656.

[48] R. H. Byrd, F. E. Curtis, and J. Nocedal, *An inexact SQP method for equality constrained optimization*, SIAM J. Optim., 19 (2008), pp. 351–369.

[49] E. Casas, *Control of an elliptic problem with pointwise state constraints*, SIAM J. Control Optim., 24 (1986), pp. 1309–1818.

[50] E. Casas and F. Tröltzsch, *Error estimates for linear-quadratic elliptic control problems*, in Analysis and Optimization of Differential Systems, V. Barbu, ed., Kluwer Academic Publishers, 2003, pp. 89–100.

[51] P. G. Ciarlet and J. L. Lions, *Handbook of Numerical Analysis, Volume II: Finite Element Methods (Part 1)*, North-Holland, 2003.

[52] T. F. Coleman and A. Verma, *A preconditioned conjugate gradient approach to linear equality constrained minimization*, Comput. Optim. Appl., 20 (2001), pp. 61–72.

[53] S. S. Collis and M. Heinkenschloss, *Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems*, Tech. Rep. 02-01, CAAM, Rice University, 2002.

[54] P. M. de Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver*, J. Comput. Appl. Math., 33 (1990), pp. 1–27.

[55] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[56] J. Dendy, *Black box multigrid*, J. Comput. Phys., 48 (1982), pp. 366–386.

[57] A. Draganescu and T. F. Dupont, *Optimal order multilevel preconditioners for regularized ill-posed problems*, Math. Comp., 77 (2008), pp. 2001–2038.

[58] T. Dreyer, B. Maar, and V. Schulz, *Multigrid optimization in applications*, J. Comput. Appl. Math., 120 (2000), pp. 67–84.

[59] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[60] S. C. Eisenstat and H. F. Walker, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Comput., 17 (1996), pp. 16–32.

[61] H. C. Elman and G. H. Golub, *Inexact and preconditioned uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., 31 (1994), pp. 1645–1661.

[62] M. Embree, *The Tortoise and the Hare: Restart GMRES*, SIAM Review, 45 (2003), pp. 259–266.

[63] R. E. Ewing, M. M. Liu, and J. Wang, *Superconvergence of mixed finite element approximations over quadrilaterals*, SIAM J. Numer. Anal., 36 (1999), pp. 772–787.

[64] R. Falk, *Approximation of a class of optimal control problems with order of convergence estimates*, J. Math. Anal. Appl., 44 (1973), pp. 28–47.

[65] A. V. Fursikov, *Optimal control of distributed systems*, American Math. Soc., Providence, RI, 2000.

[66] O. Ghattas and G. Biros, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: the Krylov-Schur solver*, SIAM J. Sci. Comput., 27 (2005), pp. 687–713.

[67] ——, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange-Newton solver, and its application to optimal control of steady viscous flows*, SIAM J. Sci. Comput., 27 (2005), pp. 714–739.

[68] P. E. Gill, W. Murray, D. B. Ponceleón, and M. W. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 292–311.

[69] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 2. ed., 1982.

[70] L. Giraud, D. Ruiz, and A. Touhami, *A comparative study of iterative solvers exploiting spectral information for SPD systems*, SIAM J. Sci. Comput., 27 (2006), pp. 1760–1786.

[71] H. Goldberg and F. Tröeltzsch, *On a SQP-multigrid technique for nonlinear parabolic boundary control problems*, in Optimal Control: Theory, Algorithms and Applications, Kluwer Academic Publishers, 1997.

[72] K. Goto, *Goto BLAS at Texas Advanced Computing Center*, 2008. http://www.tacc.utexas.edu/resources/software/.

[73] N. Gould, D. Orban, and P. Toint, *Numerical methods for large-scale nonlinear optimization*, Acta Numer., 15 (2005), pp. 299–361.

[74] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.

[75] S. Gratton, A. Sartenaer, and P. L. Toint, *Recursive trust-region methods for multiscale nonlinear optimization*, Tech. Rep. TR/PA/06/32, CERFACS, 2006.

[76] A. Greenbaum and Z. Strakos, *Predicting the behaviour of finite precision Lanczos and conjugate gradient computations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 121–137.

[77] A. Griewank, *Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation*, Optim. Methods Software, 1 (1992), pp. 35–54.

[78] P. Grisvard, *Elliptic problems in nonsmooth domains*, Pitman, Boston, 1985.

[79] M. D. Gunzburger, *Perspectives in Flow Control and Optimization*, Advances in Design and Control, SIAM, Philadelphia, 2003.

[80] M. D. Gunzburger, L. Hou, and T. P. Svobodny, *Finite element approximations of an optimal control problem associated with the scalar Ginzburg-Landau equation*, Computers Math. Applic., 32 (1991), pp. 123–131.

[81] M. D. Gunzburger, L. Hou, and T. P. Svobodny, *Boundary velocity control of incompressible flow with an application to viscous drag reduction*, SIAM J Control Optim, 30 (1992), pp. 167–181.

[82] E. Haber and U. M. Ascher, *Preconditioned all-at-once methods for large, sparse parameter estimation*, Inverse Problems, 17 (2001), pp. 1847–1864.

[83] W. Hackbusch, *Fast solution of elliptic control problems*, J. Optim. Theory Appl., 31 (1980), pp. 565–581.

[84] ——, *Die schnelle Auflösung der Fredholmschen Integralgleichung zweiter Art*, in Beiträge zur numerischen Mathematik, Band 9, Oldenbourg, München, 1981.

[85] ——, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics, Springer, Berlin, Heidelberg, 1985.

[86] ——, *Elliptic Differential Equations*, Springer, New York, 1992.

[87] W. Hackbusch and H. D. Mittelmann, *On multi-grid methods for variational inequalities*, Numer. Math., 42 (1983), pp. 65–76.

[88] J. Haslinger and P. Neittaanmäki, *Finite Element Approximation for Optimal Shape, Material and Topology Design*, John Wiley & Sons, 2nd ed., 1996.

[89] M. Heinkenschloss, *Formulation and analysis of a sequential quadratic programming method for the optimal Dirichlet boundary control of Navier-Stokes flow*, in Optimal Control: Theory, Algorithms and Applications, W. W. Hager and P. M. Pardalos, eds., Kluwer Academic Publishers, 1991, pp. 178–203.

[90] P. W. Hemker, *On the order of prolongations and restrictions in multigrid procedures*, J. Comput. Appl. Math., 32 (1990), pp. 423–429.

[91] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM J. Optim., 13 (2003), pp. 865–888.

[92] M. Hintermüller and M. Ulbrich, *A mesh-independence result for semismooth Newton methods*, Math. Program., 101 (2004), pp. 151–184.

[93] M. Hinze, *Optimal and instantaneous control of the instationary Navier-Stokes equations*, habilitation thesis, Technische Universität Berlin, 2000.

[94] ——, *A variational discretization concept in control constrained optimization: The linear-quadratic case*, Comput. Optim. Appl., 30 (2005), pp. 45–61.

[95] M. Hinze and M. Hintermüller, *A SQP-semi-smooth Newton-type algorithm applied to control of the instationary Navier-Stokes system subject to control constraints*, SIAM J. Optim., 16 (2006), pp. 1177–1200.

[96] M. Hinze and K. Kunisch, *Second order methods for optimal control of time-dependent fluid flow*, SIAM J. Control Optim., 40 (2001), pp. 925–946.

[97] M. Hinze and R. Pinnau, *Optimal control of the drift diffusion model for semiconductor devices*, in Optimal Control of Complex Structures, Birkhäuser, 2001, pp. 95–106.

[98] R. H. W. Hoppe, *Multigrid algorithms for variational inequalities*, SIAM J. Numer. Anal., 24 (1987), pp. 1046–1065.

[99] HSL, *A collection of threadsafe ISO Fortran codes for large scale scientific computation*. `http://www.cse.scitech.ac.uk/nag/hsl/`, 2008.

[100] K. Ito and K. Kunisch, *Augmented Lagrangian-SQP methods for nonlinear optimal control problems of tracking type*, SIAM J. Control Optim., 34 (1996), pp. 874–891.

[101] P. Jiránek and M. Rozložník, *Maximum attainable accuracy of inexact saddle point solvers*, SIAM J. Matrix Anal. Appl., 29 (2008), pp. 1297–1321.

[102] C. Keller, N. I. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.

[103] H. Kim, J. Xu, and L. Zikatanov, *Uniformly convergent multigrid methods for convection-diffusion problems without any constraint on coarse grids*, Adv. Comput. Math., 20 (2004), pp. 385–399.

[104] J. T. King, *Multilevel algorithms for ill-posed problems*, Numer. Math., 61 (1992), pp. 311–334.

[105] R. A. Klausen and R. Winther, *Robust convergence of multi point flux approximation on rough grids*, Numer. Math., 104 (2006), pp. 317–337.

[106] T. Köhler, P. Maass, P. Wust, and M. Seebass, *A fast algorithm to find optimal controls of multiantenna applicators in regional hyperthermia*, Phys. Med. Biol., 46 (2001), pp. 2503–2514.

[107] R. Kornhuber, *Monotone multigrid methods for elliptic variational inequalities I*, Numer. Math., 69 (1994), pp. 167–184.

[108] K. Kunisch and E. W. Sachs, *Reduced SQP methods for parameter identification problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1793–1820.

[109] R. M. Lewis and S. G. Nash, *A multigrid approach to the optimization of systems governed by differential equations*, in 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, 2000, AIAA.

[110] S. Leyffer, *Integrating SQP and branch-and-bound for mixed integer nonlinear programming*, Comput. Optim. Appl., 18 (2001), pp. 295–309.

[111] J.-L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Springer, Berlin, 1971.

[112] ——, *Control of distributed singular systems*, Gauthier-Villars, Paris, 1985.

[113] J. L. Lions and E. Magenes, *Non-homogeneous boundary value problems and applications/1*, Die Grundlehren der mathematischen Wissenschaften, Springer, Berlin, 1972.

[114] K. Lipnikov, M. Shashkov, and D. Svyatskiy, *The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes*, J. Comput. Phys., 211 (2006), pp. 473–491.

[115] D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1969.

[116] L. Lukšan and J. Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained non-linear programming problems*, Numer. Linear Algebra Appl., 5 (1998), pp. 219–247.

[117] J. Maryška, M. Rozložník, and M. Tůma, *Schur complement systems in the mixed-hybrid finite element approximation of the potential fluid flow problem*, SIAM J. Sci. Comput., 22 (2000), pp. 704–723.

[118] H. Maurer and H. D. Mittelmann, *Optimization techniques for solving elliptic control problems with control and state constraints: Part 2. distributed control*, Comput. Optim. Appl., 18 (2001), pp. 141–160.

[119] H. Maurer and J. Zowe, *First-and second-order necessary and sufficient optimality conditions for infinite-dimensional programming problems*, Math. Program., 16 (1979), pp. 98–110.

[120] G. Meurant and Z. Strakos, *The Lanczos and conjugate gradient algorithms in finite precision arithmetic*, Acta Numerica, 15 (2006), pp. 471–542.

[121] C. Meyer and A. Rösch, *Superconvergence properties of optimal control problems*, SIAM J. Control Optim., 43 (2004), pp. 970–985.

[122] B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*, Oxford University Press, 2001.

[123] M. Mohr and R. Wienands, *Cell-centred multigrid revisited*, Comput. Visual. Sci., 7 (2004), pp. 129–140.

[124] S. Nash, *A multigrid approach to discretized optimization problems*, Optimization Methods and Software, 14 (2000), pp. 99–116.

[125] S. G. Nash and A. Sofer, *Preconditioning reduced matrices*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 47–68.

[126] F. Natterer, *Regularisierung schlecht gestellter Probleme durch Projektionsverfahren*, Numer. Math., 28 (1977), pp. 329–341.

[127] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, Berlin, Heidelberg, 1999.

[128] C. W. Oosterlee and T. Washio, *An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems*, SIAM J. Sci. Comput., 19 (1998), pp. 87–110.

[129] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, 1970.

[130] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[131] P. A. Raviart and J. M. Thomas, *A mixed finite element method for second order elliptic problems*, in Mathematical Aspects of the Finite Element Method, I. Galligani, ed., no. 606 in Lecture notes in mathematics, Berlin, 1977, Springer.

[132] S. Ravindran, *A reduced order approach for optimal control of fluids using proper orthogonal decomposition*, Internat. J. Numer. Methods Fluids, 34 (2000), pp. 425–448.

[133] A. Rieder, *A wavelet multilevel method for ill-posed problems stabilized by Tikhonov regularization*, Numer. Math., 75 (1997), pp. 501–522.

[134] H. G. Roos, M. Stynes, and L. Tobiska, *Numerical methods for singularly perturbed differential equations - convection diffusion and flow problems*, Springer, Berlin, 1996.

[135] A. Rösch and D. Wachsmuth, *Numerical verification of optimality conditions*, SIAM J Control Optim, 47 (2008), pp. 2557–2581.

[136] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 368–391.

[137] T. Rusten and R. Winther, *A preconditioned iterative method for saddle-point problems*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 887–904.

[138] Y. Saad, *A flexible inner-outer preconditioned gmres algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 461–469.

[139] Y. Saad and H. M. Schultz, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[140] J. Schöberl and W. Zulehner, *Symmetric indefinite preconditioners for saddle point systems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 752–773.

[141] V. Schulz and G. Wittum, *Multigrid optimization methods for stationary parameter identification problems in groundwater flow*, in Multigrid Methods V, W. Hackbusch and G. Wittum, eds., Lecture Notes in Computational Science and Engineering, Springer, Berlin, 1998, pp. 276–288.

[142] M. Shashkov, *Conservative Finite-Difference Methods on General Grids*, CRC Press, Boca Raton, FL, 1996.

[143] M. Shashkov and S. Steinberg, *Solving diffusion equations with rough coefficients in rough grids*, J. Comput. Phys., 129 (1996), pp. 383–405.

[144] G. L. G. Sleijpen, H. A. V. D. Vorst, and J. Modersitzki, *Differences in the effects of rounding errors in Krylov solvers for symmetric indefinite linear systems*, SIAM J. Matrix Anal. Appl., 22 (2000), pp. 726–751.

[145] F. Tröltzsch, *Optimale Steuerung partieller Differentialgleichungen. Theorie, Verfahren und Anwendungen*, Vieweg, Wiesbaden, 2005.

[146] F. Tröltzsch and S. Volkwein, *The SQP method for control constrained optimal control of the Burgers equation*, ESAIM: Control, Optimisation and Calculus of Variations, 6 (2001), pp. 649–674.

[147] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[148] M. Ulbrich, *Semismooth Newton methods for operator equations in function spaces*, SIAM J. Optim., 13 (2003), pp. 805–841.

[149] S. Ulbrich and J. C. Ziems, *Adaptive multilevel inexact SQP-methods for PDE-constrained optimization*, tech. rep., TU Darmstadt, 2008.

[150] M. Vallejos and A. Borzi, *Multigrid optimization methods for linear and bilinear elliptic optimal control problems*, Computing, 82 (2008), pp. 31–52.

[151] S. Vanka, *Block-implicit multigrid solution of navier-stokes equations in primitive variables*, J. Comput. Phys., 65 (1986), pp. 138–158.

[152] R. Verfürth, *A combined conjugate gradient - multi-grid algorithm for the numerical solution of the Stokes problem*, IMA J. Numer. Anal., 4 (1984), pp. 441–455.

[153] C. R. Vogel, *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics, SIAM, 2002.

[154] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons, Chichester, 1992.

[155] G. Wittum, *Multi-Grid methods for Stokes and Navier-Stokes equations. Transforming smoothers: algorithms and numerical results*, Numer. Math., 54 (1989), pp. 543–563.

[156] I. Yavneh, *Coarse-grid correction for nonelliptic and singular perturbation problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1682–1699.

[157] E. Zeidler, *Variational methods and optimization*, vol. 3 of Nonlinear functional analysis and its applications, Springer, 1985.